Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 13969, 12 pages
doi:10.1155/2007/13969

Research Article

Cross-Layer Perceptual ARQ for Video Communications over 802.11e Wireless Networks

P. Bucciol,[1] E. Masala,[1] E. Filippi,[2] and J. C. De Martin[1]

[1] Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Torino, Italy
[2] Advanced System Technologies, STMicroelectronics S.r.l., Cornaredo, 20010 Milano, Italy

Received 29 December 2006; Revised 13 April 2007; Accepted 2 July 2007

Recommended by Zhu Han

This work presents an application-level perceptual ARQ algorithm for video streaming over 802.11e wireless networks. A simple and effective formula is proposed to combine the perceptual and temporal importance of each packet into a single priority value, which is then used to drive the packet-selection process at each retransmission opportunity. Compared to the standard 802.11 MAC-layer ARQ scheme, the proposed technique delivers higher perceptual quality because it can retransmit only the most perceptually important packets reducing retransmission bandwidth waste. Video streaming of H.264 test sequences has been simulated with *ns* in a realistic 802.11e home scenario, in which the various kinds of traffic flows have been assigned to different 802.11e access categories according to the Wi-Fi alliance WMM specification. Extensive simulations show that the proposed method consistently outperforms the standard link-layer 802.11 retransmission scheme, delivering PSNR gains up to 12 dB while achieving low transmission delay and limited impact on concurrent traffic. Moreover, comparisons with a MAC-level ARQ scheme which adapts the retry limit to the type of frame contained in packets and with an application-level deadline-based priority retransmission scheme show that the PSNR gain offered by the proposed algorithm is significant, up to 5 dB. Additional results obtained in a scenario in which the transmission relies on an intermediate node (i.e., the access point) further confirms the consistency of the perceptual ARQ performance. Finally, results obtained by varying network conditions such as congestion and channel noise levels show the consistency of the improvements achieved by the proposed algorithm.

1. INTRODUCTION AND MOTIVATIONS

The IEEE 802.11 wireless local area networking standard [1] provides network access capabilities to an ever expanding array of devices, including multimedia-enabled devices. The 802.11 standard addresses channel noise and MAC-level collision issues by means of a link-layer automatic repeat request (ARQ) scheme. This mechanism is well suited for generic data transmission, because it is fast and simple to implement. However, for the specific—and increasingly important—case of multimedia traffic, more advanced ARQ techniques could deliver higher perceptual quality as well as use network resources more efficiently.

Multimedia communications exhibit peculiar features if compared with conventional data transmissions. Two of the most important characteristics of multimedia streams, in fact, are the highly nonuniform perceptual importance of data and the strong time sensitivity. Most ARQ techniques designed for multimedia communications often consider one or both characteristics. For instance, the *Soft ARQ* proposal [2] avoids retransmitting late data that would not be useful at the decoder, thus saving bandwidth. This technique has also been adapted to deal with layered encoded streams.

Other works focused on optimizing prioritization mechanisms in order to take advantage of the different perceptual importance of the syntax elements contained in a compressed multimedia bitstream. For instance, video packets can be protected by different error correcting codes depending on the type of frame to which the packets belong, as in [3], in which an additional ARQ scheme that privileges the most important classes of data is also implemented by means of different retry limit values. The work in [4] proposes to schedule video frames according to the priority given by their position inside the group of pictures (GOP), and at the same time, it assigns different priorities to motion and texture information contained in each packet. A retry limit adaptation scheme for layered video has been proposed in [5]. That work presents an algorithm which can dynamically determine the best retry limit value for each layer, depending on channel error and MAC-level buffer overflow probabilities, in a priority queueing transmission system.

Further improvements can be achieved by optimizing the transmission policy for each single packet rather than relying on a priori determination of the average importance of the elements contained in the compressed bitstream [6, 7]. In the low-delay wireless video transmission system presented in [8], for instance, 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 [9–11].

This work addresses the specific case of video streaming over a congested 802.11 network. In order to overcome the limitations of the standard 802.11 MAC-level ARQ which retransmits all packets regardless of their importance, we propose a cross-layer perceptual ARQ scheme which exploits information about the *perceptual* and the *temporal* importance of each packet. The proposed ARQ scheme is composed of three parts: an algorithm which determines retransmission opportunities, a retransmission scheduling algorithm, and a formula to compute a priority value for each packet. More specifically, first a set of retransmission opportunities is determined at the beginning of each GOP, then the scheduling algorithm retransmits packets according to their priority and on the basis of the receiver feedback. The priority of each packet 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 [10], which is explained in Section 3.

This paper presents a detailed analysis of the cross-layer perceptual ARQ scheme first presented in [12]. Extensive simulation results quantify the impact of the main algorithm parameters and illustrate how varying levels of congestion or channel noise affect the performance of the proposed ARQ scheme. This work focuses on a congested 802.11e home network scenario in which the access point represents the home access gateway (HAG). Test H.264 video transmissions in presence of several concurrent interfering flows are simulated. Two scenarios have been considered: direct transmission, from the access point to a PC, and indirect transmission from the PC to a TV set, relaying on the access point. Both perceptual video quality (as measured by PSNR) and network performance metrics are obtained in different conditions and for different values of the main algorithm parameters, such as the maximum retransmission bandwidth. Moreover, the sensitivity of the proposed technique to variations in the scenario (i.e., the amount of concurrent traffic and channel noise) has also been evaluated. Besides the standard MAC-level ARQ scheme, two reference techniques have been implemented and studied for comparison purposes. The first technique is a deadline-driven application-level ARQ in which the highest retransmission priority is given to the packet whose playout deadline is the nearest, similarly to [2]. The second one is a MAC-level ARQ technique which imposes different retry limits value for each type of packet, as proposed in [3], to give unequal protection to the various elements of the video sequence.

The remainder of the paper is organized as follows. Section 2 briefly reviews the H.264 standard focusing on communication issues. Then, Section 3 provides details on the analysis-by-synthesis distortion estimation technique, which is used compute perceptual importance values. Section 4 describes the cross-layer ARQ technique studied in this work. Simulation setup and an extensive discussion of results are presented in Sections 5 and 6, respectively. Finally, conclusions are drawn in Section 7.

2. H.264 VIDEO COMMUNICATIONS

In this work, we consider video communications based on the state-of-the-art H.264 video codec [13, 14]. This codec is particularly suitable for transmission over packet networks. In fact, one of the most interesting characteristics of the H.264 standard is the attempt to decouple the coding aspects from the bitstream adaptation needed to transmit it over a particular channel. The part of the standard that deals with the coding aspects is called Video Coding Layer (VCL), while the other is the network adaptation layer (NAL).

As in previous video coding standards, the H.264 VCL groups consecutive macroblocks into *slices*, that are the smallest independently decodable units. Slices are important because they allow 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 bitstream of others.

Differently from other video coding standards, the H.264 provides a NAL which aims to efficiently support transmission over IP networks [15]. In particular, it relies on the use of the real-time transport protocol (RTP), which is well suited for real-time wired and wireless multimedia transmissions. The implementation of our proposed algorithm is compliant with this NAL specification.

However, some dependencies exist between the VCL and the NAL. For instance, the packetization process 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. The packetization strategy, as the frame subdivision into slices, is not standardized and the encoder has the possibility to vary both of them for each frame. Usually, however, the maximum packet size (hence slice size) is limited and slices cannot be too short due to the resulting overhead that would reduce coding efficiency.

3. DISTORTION ESTIMATION

The quality of multimedia communications over packet networks may be impaired in case of packet loss. The amount of quality degradation strongly vary depending on the importance of the lost data. In order to design efficient loss protection mechanisms, a reliable importance estimation method for multimedia data is needed. Such importance is often defined a priori, based on the average importance of the elements of the compressed bitstream, as with the data partitioning approach.

In order to provide a quantitative importance estimation method at a finer level of granularity, we define the importance of a video coding element, such as a macroblock or a packet, as a value proportional to the distortion that would be introduced at the decoder by the loss of that specific element. The potential distortion of each element, could, therefore, be computed using the analysis-by-synthesis technique [10]. The conceptual scheme is depicted in Figure 1. In this work, we apply the analysis-by-synthesis technique on a packet basis. Hence, the video sequence has to be coded and packetized before the activation of the algorithm. The analysis-by-synthesis distortion estimation algorithm performs, for each packet, the following steps:

(1) decoding, including concealment, of the bitstream simulating the loss of the packet being analyzed (synthesis stage);

(2) quality evaluation, that is, computation of the distortion caused by the loss of the packet. The original and the reconstructed picture after concealment are compared using, for example, MSE;

(3) storage of the obtained value as an indication of the perceptual importance of the analyzed video packet.

The previous operations can be implemented with small modifications of the standard encoding process. The encoder, in fact, usually reconstructs the coded pictures simulating the decoder operations, since this is needed for motion-compensated prediction. If step (1) of the analysis-by-synthesis algorithm exploits the operations of the encoding software, complexity is only due to the simulation of the concealment algorithm. In case of simple temporal concealment techniques, this is trivial and the task is reduced to provide the data to the quality evaluation algorithm.

The analysis-by-synthesis technique, as a principle, can be applied to any video coding standard. In fact, it is based on repeating the same steps that a standard decoder would perform, including error concealment. Obviously, the importance values computed with the analysis-by-synthesis algorithm are dependent on a particular encoding, that is, if the video sequence is compressed with a different encoder or using a different packetization, values will be different. Note, however, that in principle the analysis-by-synthesis scheme does not impose any particular restriction on encoding parameters or packetization.

Due to the interdependencies usually present between data units, the simulation of the loss of an isolated data unit is not completely realistic, particularly for high packet loss rates. Every possible combination of events should ideally be considered, weighted by its probability, and its distortion computed by the analysis-by-synthesis technique, obtaining the expected distortion value. For simplicity, however, we assume that all preceding data units have been correctly received and decoded. Nevertheless, this leads to a useful approximation as demonstrated by some applications of the analysis-by-synthesis approach to MPEG-coded video [6, 7, 10]. The results section will show the effectiveness of the proposed video transmission algorithm which relies on these distortion values.

The application of the analysis-by-synthesis method is straightforward when considering elements of the video stream which do not contribute to later referenced frames, since the mismatch due to concealment does not propagate. If propagation is possible, the distortion caused in subsequent frames should be evaluated until it becomes negligible, for instance, at the beginning of the next group of pictures (GOP) for MPEG video, or until its value falls below a given threshold. In this case, the complexity of the proposed approach could be high, but it is still suitable for stored-video scenarios that allow precomputation. In order to reduce complexity, statistical studies on many different video sequences have been conducted and a model-based approach [16] has been developed. According to that model, the encoder computes the distortion that would be caused by the loss of the packet into the current frame and then, using a simple formula, it computes an estimation of the total distortion which includes future frames. The reader is referred to the work in [16] for further details.

4. IMPORTANCE-BASED CROSS-LAYER ARQ

4.1. *Overview*

This work proposes an application-level end-to-end ARQ technique which relies on the perceptual and temporal importance of each multimedia packet in order to optimize the usage of retransmission bandwidth. The technique has been designed to work with the IP/UDP/RTP protocol stack [17]. RTCP packets are used to provide feedback information.

According to the proposed technique, every packet is transmitted once, then it is stored in a retransmission buffer RTX_{buf} waiting for its acknowledgment. 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. When a retransmission opportunity is available, packets in the retransmission buffer are sent in the order given by their combined temporal-perceptual priority, as defined in Section 4.4.

A few key parameters can be used to tune the performance of the proposed technique, for instance, the peak transmission bandwidth B_{peak} granted to retransmissions and the relative weight of temporal with respect to perceptual importance.

4.2. *Retransmission opportunities*

The first step of the scheduling algorithm consists in determining the transmission time of each packet. The task is carried out, at the beginning of each GOP, 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 let B_{peak} be the peak bandwidth granted to the transmission, including retransmissions. Retransmission opportunities are identified using the following algorithm.

(1) Determine the number of retransmission opportunities N_{rtx}, for the current GOP, as $N_{rtx} = \lfloor B_{\text{peak}} - B_{\text{GOP}} \rfloor / \overline{S}_{pck}$, where $\overline{S}_{pck}$ is the average size of all the packets belonging to the original video sequence.

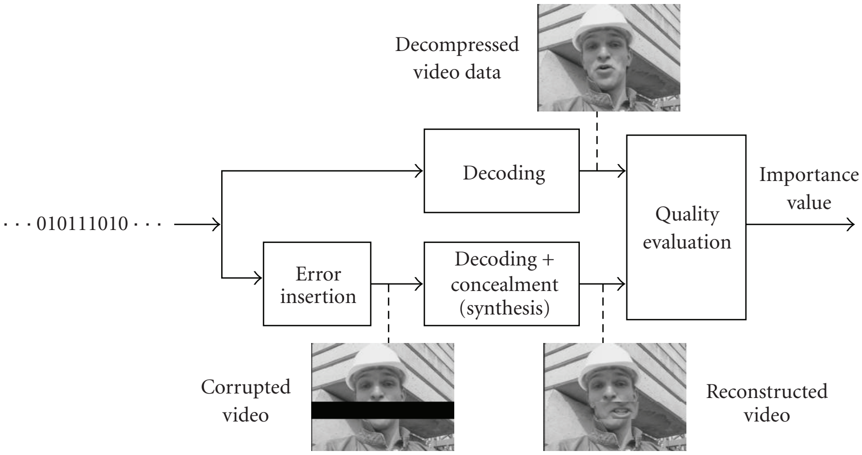

Figure 1: Conceptual scheme of the analysis-by-synthesis technique.

(2) Determine the time instants corresponding to the retransmission opportunities.

(2.1) Compute the total size, including retransmissions, of each frame. Identify the smallest one.

(2.2) Set the time of a retransmission opportunity as the midway between the time instant of the first packet of the smallest frame (including retransmissions) and the last packet of the previous frame.

(2.3) Repeat steps (2.1) and (2.2) until N_{rtx} opportunities have been determined, considering at each step the opportunities filled by packets of size $\overline{S}_{pck}$ and including them in the total frame size.

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 can be designed. Note also that the opportunities will not be necessarily used completely.

4.3. Scheduling algorithm

The retransmission policy, illustrated in Figure 2, is implemented by means of a retransmission buffer $\mathrm{RTX}_{\mathrm{buf}}$. After that a packet is sent for the first time, it is placed in the $\mathrm{RTX}_{\mathrm{buf}}$, waiting for its acknowledgment, and marked as *unavailable* for retransmission. When an ACK is received, the corresponding packet in the $\mathrm{RTX}_{\mathrm{buf}}$ is discarded because it has been successfully transmitted. If a NACK is received, the corresponding packet is marked as *available* for retransmission. Packets belonging to the $\mathrm{RTX}_{\mathrm{buf}}$ that will never arrive at the decoder in time for playback (considering the estimated FTT) are discarded. To limit the impact of receiver report losses, the sender piggybacks the highest sequence number for which it received an ACK or a 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.

A priority function (see Section 4.4) is computed for each packet marked as *available* in the $\mathrm{RTX}_{\mathrm{buf}}$ each time a new retransmission opportunity approaches. The packet with the highest priority is then transmitted.

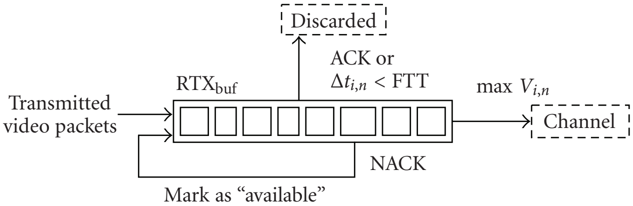

Figure 2: Diagram of the scheduling algorithm for packet retransmission.

It is important to stress that the retransmission opportunities computed according to B_{peak} *not necessarily* will be actually used by the algorithm, leading to an actual bandwidth usage which can be considerably lower than B_{peak}.

4.4. Packet priority function

In a generic multimedia 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 nth frame is played back. All packets containing data needed to synthesize the nth frame must be available at the decoder at time $t_n - T_P$, where T_P is the decoder processing time. Note that data dependencies in the coded video (e.g., due to reference to future frames) must also be considered.

We define *deadline* of a packet as the time instant at which that packet must be available at the decoder to be played back correctly. Let $t_{i,n}$ be the deadline of a packet i belonging to the nth frame. From the definition, it is clear that $t_{i,n} = t_n - T_P$. If a packet never arrives or it arrives after $t_{i,n}$, it will cause a distortion increase $D_{i,n}$ that can be evaluated using the analysis-by-synthesis technique.

Obviously, the sender should always select a packet for transmission only among the ones that can arrive before their deadline, that is, $t_{i,n} > t_s + \mathrm{FTT}$, 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 depending on 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 $\Delta t_{i,n} > \text{FTT}$.

A policy is needed to choose which packet must be retransmitted and in which order, because at any given time several packets satisfy the condition $\Delta t_{i,n} > \text{FTT}$. Consider, for instance, 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].

A criterion is needed to select, at each retransmission opportunity, the video packet which maximizes the expected quality of the transmission. We propose to compute, for each packet, a priority function of both its potential distortion and its distance from the deadline:

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

Packets will then be sorted by $V_{i,n}$ and the one with the highest priority value is sent. 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:

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

The normalization factor C is computed as

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

where T_B is the receiver buffer length, in seconds, and $\overline{D_{i,n}}$ is the average packet distortion. The normalization factor, C, 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 (2) is introduced to control the relative importance of the perceptual and temporal terms of the formula.

5. SIMULATION SCENARIO

The network simulator *ns* [18] has been used to assess the performance of the proposed technique. An 802.11e MAC layer [19] has been configured to operate 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.

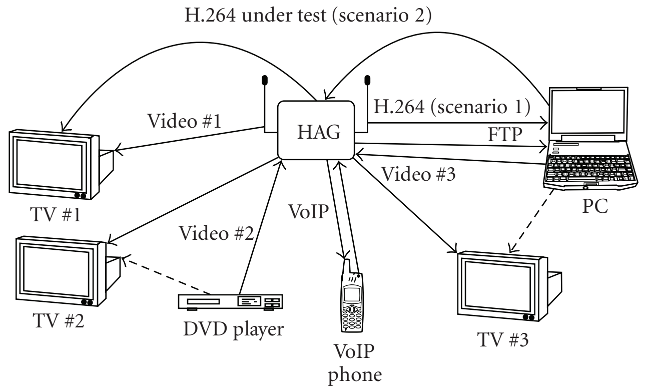

Figure 3: Home network scenario used in the experiments. The tested H.264 video stream is transmitted from the home access gateway (HAG), that is, the access point, to the PC (scenario 1) or from the PC to the TV set (scenario 2). The solid lines show the actual path of the transmitted packets, while the dashed lines indicate logical connections.

We simulated an H.264 video streaming transmission in a realistic home network scenario, in which many wireless devices, that is, three TV sets, a DVD player, a PC, and a VoIP terminal, all share the same physical bandwidth. The home access gateway is represented by the access point. Three concurrent video transmissions, a VoIP call, an FTP transfer as well as the H.264 video transmission under test, are active at the same time. Two scenarios have been analyzed. In scenario 1, the H.264 packets originate from the access point and they are directly sent to the PC, while in scenario 2, the H.264 packets are sent from the PC to the TV #1 by means of the access point which acts as a relay node. Both scenarios are depicted in Figure 3.

Tests have been performed using three different standard sequences (*paris*, *tempete*, *bus*) at CIF (352 × 288) resolution. They were encoded using version 6.1e of the H.264 test model software [13] with a fixed quantization parameter. The GOP encoding scheme is IBBPBBPBBPBB. The characteristics of the tested video sequences are shown in Table 1. Each sequence is concatenated with itself to reach a length of approximately 500 seconds. The video encoder is instructed to make RTP packets whose size is approximately constant. Unless otherwise noted, the playout buffer size is 1-second 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 assignment of the various kinds of traffic to the 802.11e access categories has been based on the Wi-Fi alliance WMM specification [20]. The FTP stream is assigned to the lowest priority class, that is, access category (AC) 0. The tested H.264 stream is assigned to AC1, while all the remaining video flows are sent as AC2. The VoIP and the receiver reports flows are assigned to AC3, which provides the highest available QoS. This assignment provides

Table 1: Characteristics of the sequences used as H.264 streams.

Sequence	Avg. bitrate (kbit/s)	Encoding distortion (dB)	Resolution	Frame rate (fps)	Maximum packet size (bytes)	Avg. number of packets/s
Paris	765	35.68	352 × 288	30	750	146
Tempete	1205	34.23	352 × 288	30	1200	141
Bus	1304	34.25	352 × 288	30	750	235

Table 2: Access category assignment for all traffic.

Access category	Stream	Bandwidth
AC0	FTP	Variable
AC1	Tested H.264	765–1304 kbit/s
AC2	Video #1	1.5 Mbit/s
	Video #2	3 Mbit/s
	Video #3	6 Mbit/s
AC3	VoIP	72 kbit/s
	RTP receiver reports	3–6 kbit/s

the maximum protection against receiver report losses. The maximum number of MAC retransmissions is seven for all the classes except AC1, for which no MAC-level retransmissions are used unless MAC-level ARQ techniques are simulated. 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 2 summarizes the assignments and the bandwidth of each flow. Note that 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 milliseconds receiver report interval, and, if needed, could be further improved by packing ACK and NACK information more efficiently than the current implementation.

6. RESULTS

6.1. MAC-level ARQ techniques

First, the performance of the MAC-level ARQ techniques is presented. Two techniques have been studied. The first one is the current 802.11 standard ARQ implementation without any modification. The second technique employs MAC-level retransmissions with a different retry limit depending on the characteristics of the data contained in the packet, which allows MAC-level unequal protection as done in [3]. We divided the video flow into two classes, the first one containing packets belonging to I and P frames, and the second one for the rest of the packets (B frames), and we tested several retry limit values ($RL_{I,P}$, RL_B) for these two classes. We refer to this technique as the class-based retry-limit (CBRL) ARQ technique.

As a reference for comparisons, the performance of both techniques as a function of the retry limit has been studied for both the *paris* and the *tempete* sequences. For the MAC-level ARQ scheme, we varied the retry limit of the AC1 class, while for the CBRL ARQ technique, various combinations of retry limit values ($RL_{I,P}, RL_B$) have been used, in particular, for the case $RL_{I,P} = RL_B + 1$ and $RL_{I,P} = RL_B + 2$. The PSNR performance of the MAC-level and the CBRL ARQ techniques is nearly equivalent and the maximum is achieved when $RL_{I,P}$ is equal to four. For higher retry limit values, the performance slightly decreases due to the higher packet delay caused by the increased network congestion. The higher packet delay, in fact, causes the expiration of the MAC-level timeout of many packets. The used bandwidth, expressed as a percentage of the average bitrate of the original sequence, shows a saturation effect when the retry limit is increased over a certain threshold, that is, about four in our simulations. However, note that the number of packets in queues is limited, although infinite size queues are assumed due to the MAC-level timeout. The corresponding used bandwidth is about 120%, that is, 20% of the bandwidth is used for retransmissions. The best results obtained with the MAC-level and the CBRL ARQ techniques will be used as reference values in the rest of the paper.

6.2. Application-level ARQ techniques: direct transmission (scenario 1)

This section is aimed at investigating the performance of the proposed perceptual ARQ algorithm in scenario 1 by means of performance indicators such as the PSNR value and the used bandwidth. Moreover, the impact of the two main parameters of the algorithm, namely, the peak bandwidth (B_{peak}) and the weighting parameter (w), will also be examined. In addition to the MAC-level and CBRL ARQ techniques, another retransmission scheme, that is, the *Soft ARQ*, will be used as reference to evaluate the performance of the proposed perceptual ARQ algorithm. The *Soft ARQ* proposal [2] assigns the highest retransmission priority to the packet whose playout deadline is the nearest. Packets whose deadline is the same (belonging to the same frame) are retransmitted sequentially. The algorithm implements the same strategy of our proposed application-level ARQ technique to compute retransmission opportunities, but the distortion term in (2) is not considered.

Figure 4 shows the performance in terms of PSNR of the proposed ARQ technique as a function of the peak bandwidth parameter, which is expressed as a percentage of the sequence average bitrate, for different values of the w parameter when the *tempete* sequence is transmitted. The graph also includes the three reference ARQ techniques. The dashed curve shows the performance of the *Soft ARQ* scheme, while the best performance achieved by the

MAC-level and the CBRL ARQ techniques, as determined in Section 6.1 by varying the MAC-level retry limit, is represented by two horizontal lines. In fact, for these two reference techniques, it is not possible to impose a peak bandwidth parameter B_{peak} constraint and the average used bandwidth is only indirectly controlled by means of the retry limit.

The proposed perceptual ARQ technique achieves a consistent performance gain, up to 0.8 dB, with respect to the best performance achieved by the standard MAC-level ARQ technique, while the gain with respect to the best performance of the CBRL ARQ technique is up to 0.5 dB, provided that the B_{peak} parameter is set higher than about 127%. Since this percentage value is computed with reference to the average bitrate of the video sequence, a value less than 127% may impose a bandwidth constraint which is sometimes lower than the instantaneous bitrate of the video sequence. This is the main reason of the performance decrease under the B_{peak} value. Hence, it is recommended that the B_{peak} parameter is set to about 130% (or higher values), as done for a large part of the results presented in this paper. With this B_{peak} setting, the bandwidth usage of the proposed ARQ technique is equal or slightly higher than the one used by the two reference techniques and the impact on concurrent traffic is very limited, as it will be shown later in this section. Hence, the 130% value represents the best tradeoff for video streaming applications in scenario 1.

The *Soft ARQ* performance in Figure 4 shows that for the same B_{peak} parameter the distortion term in (2) plays an important role. In fact, using the perceptual importance of each packet when selecting the packet to retransmit allows to achieve the error-free performance for a smaller peak bandwidth value. Note that the saturation value in the graph is equal to the encoding distortion. Different values of the w parameter have a significant impact on the performance, especially when the B_{peak} parameter is low. In this situation, the best value for the w parameter is zero, thus the packets should be retransmitted based on the perceptual importance only, that is, the distortion term in (2). If the B_{peak} parameter is increased, the best PSNR performance of the proposed technique is achieved for progressively increasing values of the w parameter. This effect can be observed in Figure 5, which presents the PSNR values as a function of the w parameter for different B_{peak} values. Each curve presents a maximum for progressively increasing values of w until flatness when the values reach the error-free PSNR performance. For the recommended B_{peak} value of approximately 130%, performance maximization is achieved using a w value equal to about one.

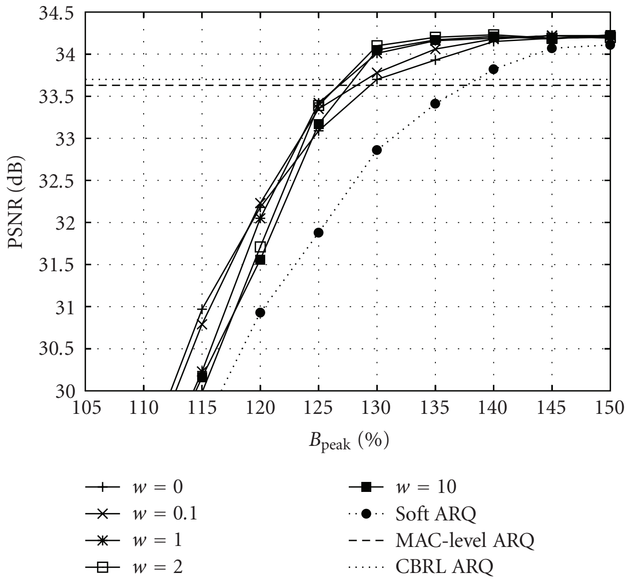

Figure 4: PSNR as a function of the peak bandwidth for the proposed ARQ scheme for different values of the w parameter, compared to the Soft ARQ technique and to the best performance of the MAC-level and CBRL ARQ schemes (*tempete* sequence).

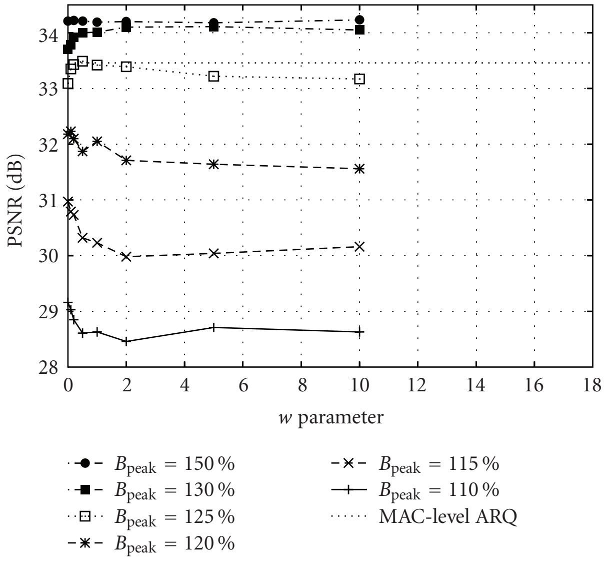

Figure 5: PSNR as a function of w parameter for the proposed perceptual ARQ technique (*tempete* sequence).

The results of the same set of experiments for the *paris* sequence are shown in Figure 6. The behavior is similar to the case of the *tempete* sequence. The performance gain is up to 0.5 dB when considering the best performance of the standard MAC-level ARQ technique, and up to 0.8 dB for the case of the CBRL ARQ technique.

Figure 7 shows the results achieved with the *bus* sequence. For this sequence, the performance of the standard MAC-level and the CBRL ARQ techniques are not shown because they are very low compared to the other results. The best performance of the standard MAC-level ARQ technique is 21.21 dB. The CBRL ARQ technique also provides similar performance (23.72 dB), which is well below the acceptable quality threshold. Hence, the gain of the proposed algorithm with respect to the MAC-level ARQ technique is up to 12 dB.

Note that the performance of the MAC-level ARQ schemes for the case of the *bus* and *tempete* sequences is strongly different despite the similar sequence average bitrate. The fact can be explained noting that the *bus* sequence is packetized into about 66% more packets per second compared to the *tempete* sequence (please refer to the last column of Table 1). The higher number of packets of the bus

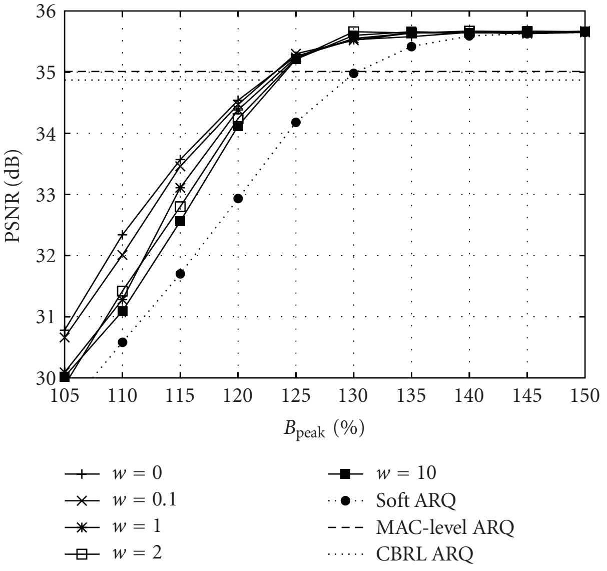

Figure 6: PSNR as a function of the peak bandwidth for the proposed ARQ scheme for different values of the *w* parameter, compared to the Soft ARQ technique and to the best performance of the MAC-level and CBRL ARQ schemes (*paris* sequence).

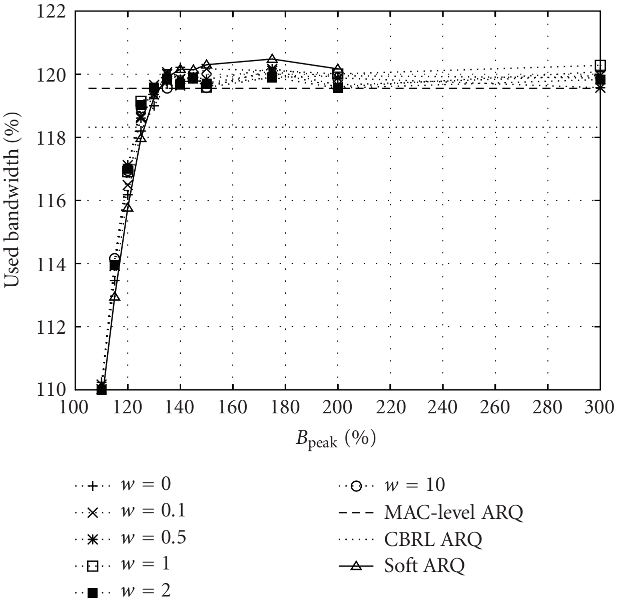

Figure 8: Used bandwidth as a function of the peak bandwidth for the proposed ARQ scheme, compared to the Soft ARQ technique and to the value corresponding to the best PSNR performance of the MAC-level and CBRL ARQ schemes (*paris* sequence).

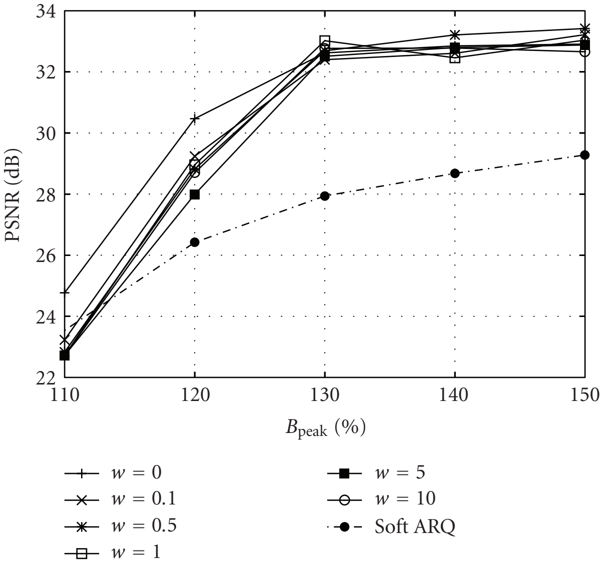

Figure 7: PSNR as a function of the peak bandwidth for the proposed ARQ scheme for different values of the *w* parameter, compared to the MAC-level ARQ performance (*bus* sequence). The maximum PSNR values achieved by the MAC-level ARQ and the CBRL ARQ techniques are not reported because they are very low.

sequence causes a saturation effect in the 802.11 network. Consequently, the performance drastically decreases. The application-level ARQ algorithms, instead, cause a lower number of network access attempts because packets are never retransmitted at the MAC level, hence good performance can be achieved in this congested scenarios as shown by the results. In this situation, it is very important to use retransmission opportunities for more perceptually important packets, as shown by the curve with *w* equal to zero. Moreover, when packets have the same playout deadline, the proposed perceptual ARQ technique selects them for retransmission in decreasing order of perceptual importance because in this case the total importance value is influenced only by the left term in (2). The *Soft ARQ* technique, instead, does not exploit the different perceptual importance of the packets and retransmits packets with the same deadline sequentially. This behavior leads to lower performance compared to the perceptual ARQ algorithm.

Figure 8 shows the average bandwidth used by the various algorithms, expressed as a percentage of the sequence average bitrate. Note that the value is much lower than the corresponding peak bandwidth parameter (B_{peak}). The peak transmission bandwidth, in fact, is fully used only when a GOP has much higher bandwidth than the average. Therefore, if B_{peak} is increased, the PSNR gain comes from allowing the algorithm to timely retransmit a higher number of packets when it is more needed. The two horizontal lines in the graph represent the bandwidth usage of the MAC-level and the CBRL ARQ techniques corresponding to their best PSNR performance. On average, the perceptual ARQ algorithm presents bandwidth usage similar to the MAC-level ARQ and *Soft ARQ* techniques and slightly higher than CBRL ARQ technique, up to 2% for the *paris* and *tempete* (not shown) sequences. However, the PSNR performance of the perceptual ARQ algorithm is consistently better than the other algorithms.

Figure 9 shows the average used bandwidth for the *bus* sequence. In this case, the difference is higher in comparison with the MAC-level ARQ (up to 5%) and the CBRL ARQ (up to 9%), but these algorithms are unable to provide an

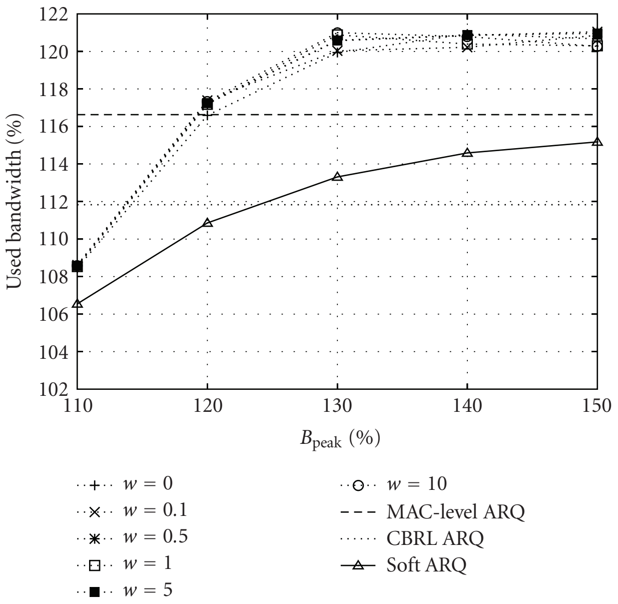

Figure 9: Used bandwidth as a function of the peak bandwidth for the proposed ARQ scheme, compared to the Soft ARQ technique and to the value corresponding to the best PSNR performance of the MAC-level ARQ scheme (*bus* sequence).

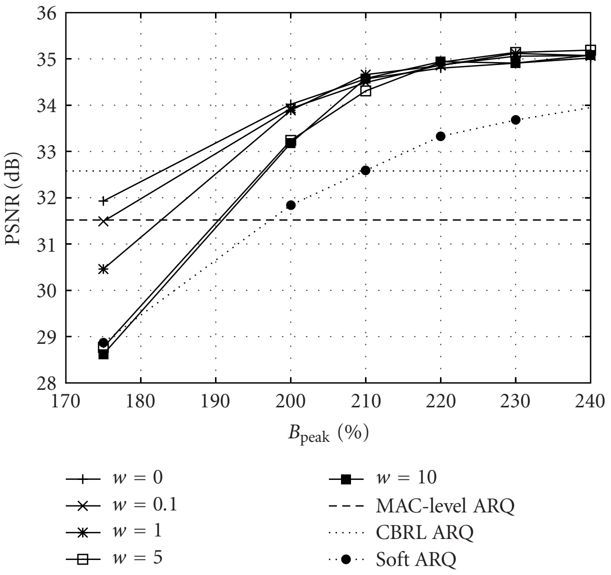

Figure 10: PSNR as a function of the peak bandwidth for the proposed ARQ scheme for different values of the *w* parameter, compared to the Soft ARQ technique and to the best performance of the MAC-level and CBRL ARQ schemes (*paris* sequence.) Transmission through the relay node.

acceptable video quality. The increase with respect to the *Soft ARQ* technique is up to 8%, but the corresponding increase of the proposed perceptual ARQ algorithm in terms of PSNR gain is significant, up to 5 dB.

6.3. Application-level ARQ techniques: transmission through a relay node (scenario 2)

The performance of the proposed perceptual ARQ algorithm has also been evaluated when a node transmits the H.264 video sequence to another node by means of the access point (scenario 2). In this case, the transmission originates from the PC and is directed to a TV set, as shown in Figure 3. Note that, in case of packet losses, the application-level ARQ techniques (i.e., the proposed perceptual ARQ and the *Soft ARQ*) employ an end-to-end retransmission approach, hence packets are sent again from the PC even if they are lost during transmission from the AP to the TV set. In this scenario, we also simulated the MAC-level and the CBRL ARQ techniques for comparison purposes. However, note that to implement the CBRL ARQ technique in this scenario, some mechanism has to be designed to let the access point know the classification of each video packet so that the correct retry limit for each packet can be used, otherwise the retry limit information would not be available at the access point.

Figures 10 and 11 show the PSNR performance of the proposed technique as a function of the B_{peak} parameter and for different *w* values. The performance is compared with the other three algorithms. The PSNR gain with respect to the MAC-level ARQ is much more pronounced than in scenario 1, up to 6 dB. With respect to the CBRL ARQ technique, the gain is still significant, up to 4 dB in the *tempete* case.

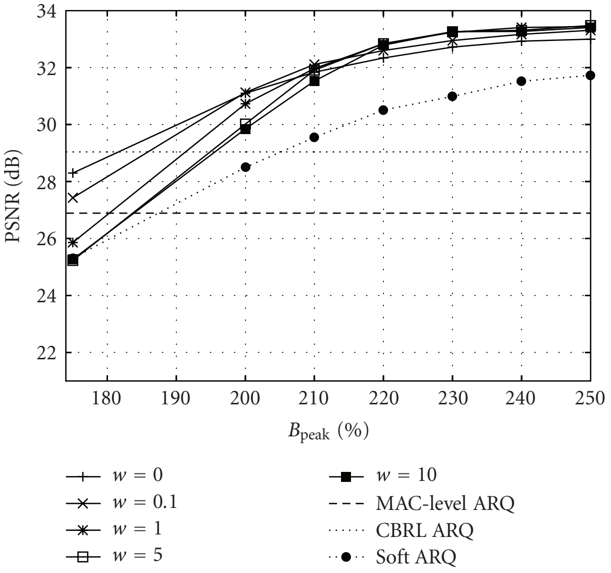

Figure 11: PSNR as a function of the peak bandwidth for the proposed ARQ scheme for different values of the *w* parameter, compared to the Soft ARQ technique and to the best performance of the MAC-level and CBRL ARQ schemes (*tempete* sequence.) Transmission through the relay node.

The comparison with the *Soft ARQ* algorithm shows that the PSNR gain is up to 3 dB if the value of the *w* which leads to the best performance is chosen. Note also that the performance of the proposed perceptual ARQ technique is more sensible to the value of the *w* parameter than in scenario 1. When the B_{peak} parameter is low, the *w* parameter should

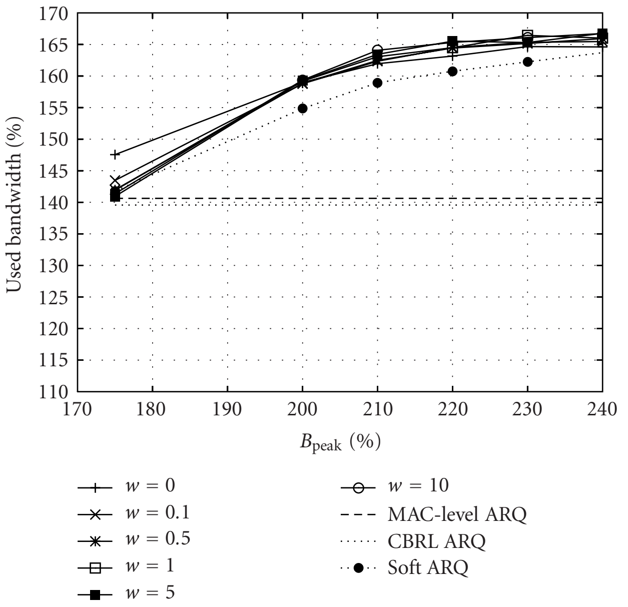

FIGURE 12: Used bandwidth as a function of the peak bandwidth for the proposed ARQ scheme, compared to the Soft ARQ technique and to the value corresponding to the best PSNR performance of the MAC-level and CBRL ARQ schemes (*paris* sequence). Transmission through the relay node.

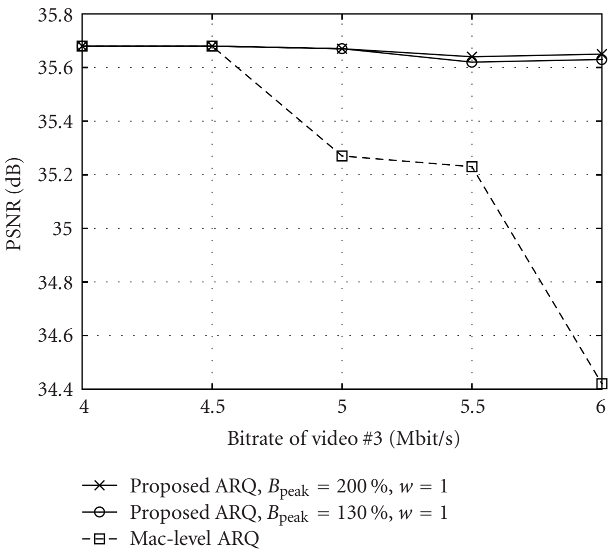

FIGURE 13: PSNR as a function of the Video #3 bandwidth (varied from 4 to 6 Mbps) for both the MAC-level ARQ and the proposed ARQ scheme; *paris* sequence.

TABLE 3: Average delay (millisecond) for the standard MAC-level ARQ and the proposed perceptual ARQ technique.

Sequence	MAC-level ARQ	Perceptual ARQ	
	retry limit = 4	B_{peak}=130%	B_{peak} = 200%
Paris	152	81	61
Tempete	870	76	49
Bus	1315	258	235

be equal to zero, so that the most perceptually important packets are privileged when a retransmission opportunity is available. The case of the *bus* sequence is not shown because none of the considered techniques is able to achieve an acceptable performance. In this case, all packets in the network experience long access delays due to the large number of packets offered to the network, and this causes a generalized performance decrease.

In Figure 12, the average used bandwidth for the *paris* sequence in scenario 2 is reported. The perceptual ARQ technique shows a used bandwidth increase with respect to the MAC-level and the CBRL ARQ techniques up to about 25% of the sequence average bitrate, however a considerable PSNR gain is provided by the perceptual ARQ technique. Moreover, the difference in used bandwidth is limited, about 5%, while the perceptual ARQ technique can achieve a PSNR gain up to 2 dB (as seen in the previous graph) with respect to the *Soft ARQ* technique. Simulations results thus indicate that the proposed perceptual ARQ technique can be effectively used in an infrastructured scenario to perform video communication between nodes.

6.4. Delays

We now present further results obtained in scenario 1 in which the video sequence under test is transmitted from the access point to the destination node without using any intermediate hop. The perceptual ARQ technique has also been evaluated in terms of the average delay, which is reported in Table 3. The results show that the MAC-level ARQ scheme causes a relatively high delay for the *tempete* and *bus* sequences, about one second or more, which might be annoying in real-time applications. Note also that, for the *bus* sequence in the MAC-level ARQ case, the playout buffer had to be increased to 2 seconds. On the contrary, the perceptual ARQ technique achieves a very low transmission delay, especially in the *paris* and *tempete* cases (50–80 milliseconds). The average delay for the *bus* sequence is slightly higher, about 250 milliseconds, which however greatly improves with respect to the 1.3 seconds average delay of the MAC-level ARQ technique and it allows to use a 1 second playout buffer as in the other cases. Moreover, increasing the peak bandwidth parameter further reduces the transmission delay. Hence, the proposed perceptual ARQ algorithm can be very interesting in scenarios with very strict delay requirements.

6.5. Influence of the scenario

This section assesses the impact of variations in scenario 1 on the performance of the proposed perceptual ARQ technique. First, the effect of different network congestion levels is evaluated. Figures 13 and 14 show the PSNR performance as a function of the bandwidth of the heaviest video flow (Video #3), varied from 4 Mbit/s to 6 Mbit/s. As illustrated in the graphs, the proposed ARQ technique is only

Table 4: Packet loss rate (%) of concurrent traffic for different retransmission schemes. B_{peak} is equal to 130% for the application-level ARQ schemes. The table shows the highest observed value.

Sequence	Traffic flow	MAC-level ARQ	Soft ARQ	Perceptual ARQ
Paris	Video #1	17.31	17.40	15.64
	Video #2	18.51	18.64	17.40
	Video #3	17.52	17.72	15.69
	VoIP	0.002	0.00	0.253
Tempete	Video #1	19.98	19.93	20.24
	Video #2	21.39	21.27	21.69
	Video #3	20.46	20.59	20.80
	VoIP	0.002	0.00	0.004
Bus	Video #1	19.45	22.16	23.06
	Video #2	21.05	23.74	24.71
	Video #3	20.39	23.60	24.55
	VoIP	0.003	0.00	0.00

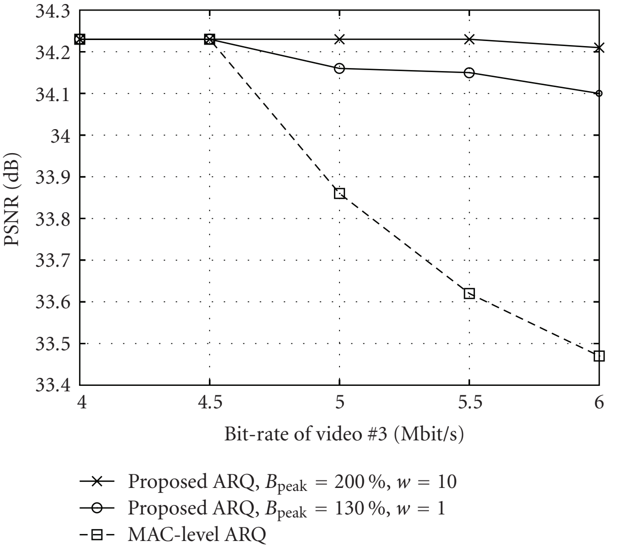

Figure 14: PSNR as a function of the Video #3 bandwidth (varied from 4 to 6 Mbps) for both the MAC-level ARQ and the proposed ARQ scheme; *tempete* sequence.

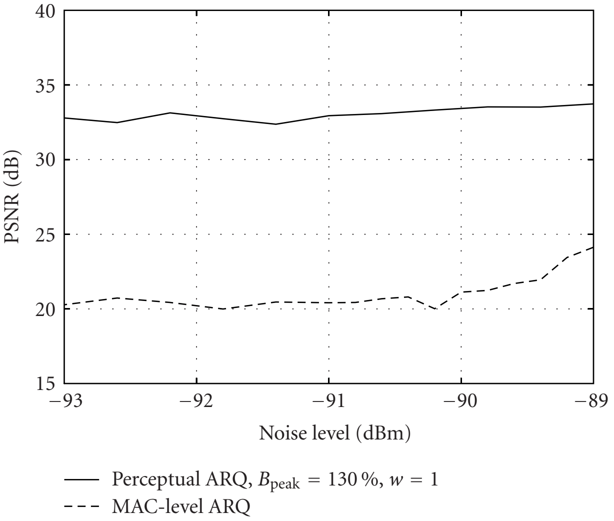

Figure 15: PSNR as a function of channel noise for both the MAC-level ARQ and the proposed ARQ scheme; *bus* sequence.

minimally affected by the augmented network load, while the MAC-level ARQ suffers a sharp decrease in terms of PSNR performance. Note also that the Video #3 transmission is relayed by the access point, hence the actual traffic offered to the network is doubled compared to the nominal bitrate.

Figure 15 compares the PSNR performance of the proposed perceptual ARQ and the MAC-level ARQ techniques as a function of the average channel noise level, for the *bus* sequence. Besides the performance gap between the perceptual and MAC-level ARQ, it is interesting to note that the perceptual ARQ performance is almost constant over the full range of the considered noise level.

The last set of results investigates the impact of the various ARQ techniques on the concurrent traffic. Results are shown in Table 4, in terms of the packet loss rate experienced by the various traffic flows, except the FTP flow which is not included because the throughput it can deliver is very limited and not significant due to the high network congestion. The results compare three techniques, namely, the MAC-level ARQ, Soft ARQ, and the proposed perceptual ARQ, for the three considered video sequences. The shown values represent the highest packet loss rate measured in the simulations. The B_{peak} parameter is set equal to 130%, which is the saturation point for the PSNR performance of the proposed perceptual ARQ technique in scenario 1. For all the three concurrent high-bandwidth video flows, the packet loss rate increase is less than 0.34% when using the perceptual ARQ instead of the MAC-level ARQ technique for the *tempete* sequence. For the *paris* video sequence, the perceptual ARQ technique causes a lower packet loss rate compared to the MAC-level ARQ. The packet loss rate slightly increases in the *bus* case (up to 4.16%); however, the proposed perceptual ARQ technique is able to deliver an acceptable quality while in the same conditions, the degradation using the

MAC-level ARQ is intolerable. Similar considerations hold when the perceptual ARQ technique is compared with the *Soft ARQ* technique, but in this case the packet loss difference is smaller. Finally, the results show that the impact on the VoIP transmission, which is assigned to AC3, that is, the highest-QoS access category, is negligible in all cases.

7. CONCLUSIONS

In this paper, we presented and investigated the performance of a perceptual ARQ algorithm for video streaming over 802.11e wireless networks. The algorithm uses a simple and effective formula in order to combine the perceptual and temporal importance of each packet into a single priority value, which is then used to drive the packet selection process at each retransmission opportunity. Extensive simulations of H.264 video streaming in a heavily congested 802.11e home scenario have been carried out by means of *ns*. The results show that the proposed method consistently outperforms the standard link-layer 802.11 retransmission scheme, with PSNR gains up to 12 dB. Comparisons with a MAC-level ARQ scheme which adjusts the retry limit of each packet based on the frame type and with an application-level deadline-based priority retransmission scheme show that the PSNR gain offered by the proposed perceptual ARQ algorithm is significant, up to 5 dB. Further results indicate that the proposed algorithm presents a very low transmission delay and a limited impact on concurrent traffic. Finally, consistent performance is achieved with various network congestion and channel noise levels.

ACKNOWLEDGMENTS

This work was supported in part by STMicroelectronics. The authors would like to thank the anonymous reviewers for their insightful and constructive comments.

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," Tech. Rep. UCB/CSD-98-1024, Computer Science Division, University of California, Berkeley, Calif, USA, November 1998.

[3] Y. Shan and A. Zakhor, "Cross layer techniques for adaptive video streaming over wireless networks," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '02)*, vol. 1, pp. 277–280, Lausanne, Switzerland, August 2002.

[4] S. H. Kang and A. Zakhor, "Packet scheduling algorithm for wireless video streaming," in *Proceedings of International Packetvideo Workshop (PV '02)*, Pittsburgh, Pa, USA, April 2002.

[5] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through realtime retry limit adaptation," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 278–290, 2004.

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

[7] F. De Vito, L. Farinetti, and J. C. De Martin, "Perceptual classification of MPEG video for Differentiated-Services communications," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '02)*, vol. 1, pp. 141–144, Lausanne, Switzerland, August 2002.

[8] S. Aramvith, 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, 2002.

[9] J. Chakareski, P. A. Chou, and B. Aazhang, "Computing rate-distortion optimized policies for streaming media to wireless clients," in *Proceedings of the Data Compression Conference (DCC '02)*, pp. 53–62, Snowbird, Utah, USA, April 2002.

[10] E. Masala and J. C. De Martin, "Analysis-by-synthesis distortion computation for rate-distortion optimized multimedia streaming," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '03)*, vol. 3, pp. 345–348, Baltimore, Md, USA, July 2003.

[11] J. Chakareski and B. Girod, "Rate-distortion optimized packet scheduling and routing for media streaming with path diversity," in *Proceedings of Data Compression Conference (DCC '03)*, pp. 203–212, Snowbird, Utah, USA, March 2003.

[12] P. Bucciol, G. Davini, E. Masala, E. Filippi, and J. C. De Martin, "Cross-layer perceptual ARQ for H.264 video streaming over 802.11 wireless networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 5, pp. 3027–3031, Dallas, Tex, USA, November-December 2004.

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

[14] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

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

[16] F. De Vito, D. Quaglia, and J. C. De Martin, "Model-based distortion estimation for perceptual classification of video packets," in *Proceedings of the 6th IEEE Workshop on Multimedia Signal Processing (MMSP '04)*, pp. 79–82, Siena, Italy, September-October 2004.

[17] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," *RFC 3550*, July 2003.

[18] UCB/LBNL/VINT, "Network Simulator—ns-version 2," http://www.isi.edu/nsnam/ns/, 1997.

[19] IEEE 802 Committee, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications - Amendment 8: Medium access control (MAC) quality of service enhancements," *IEEE Std 802.11e*, September 2005.

[20] Wi-Fi alliance, "WMM specification, v.1.1," http://www.wi-fi.org/, 2006.