Cross-Layer Packetization and Retransmission Strategies for Delay-Sensitive Wireless Multimedia Transmission

Mihaela van der Schaar, *Senior Member, IEEE*, and Deepak S. Turaga, *Member, IEEE*

***Abstract*—Existing wireless networks provide dynamically varying resources with only limited support for the quality of service required by the bandwidth-intense, loss-tolerant, and delay-sensitive multimedia applications. This variability of resources does not significantly impact delay insensitive data transmission (e.g., file transfers), but has considerable consequences for multimedia applications. Recently, the research focus has been to adapt existing algorithms and protocols at the lower layers of the protocol stack to better support multimedia transmission applications, and conversely, to modify application layer solutions to cope with the varying wireless networks resources. In this paper, we show that significant improvements in wireless multimedia performance can be obtained by deploying a joint application-layer adaptive packetization and prioritized scheduling and MAC-layer retransmission strategy. We deploy a state-of-the-art wavelet coder for the compression of the video data that enables on-the-fly adaptation to changing channel conditions and inherent prioritization of the video bitstream. We pose the cross-layer problem as a distortion minimization given delay constraints and derive analytical solutions by modifying existing joint source-channel coding theory aimed at fulfilling rate, rather than delay, constraints. We also propose real-time algorithms that explicitly consider the available information about previously transmitted packets. The obtained results show significant improvements in terms of video quality as opposed to *ad-hoc* optimizations currently deployed, while the complexity associated with performing this optimization in real time, i.e., at transmission time, is limited.**

***Index Terms*—Cross-layer wireless multimedia transmission, packetization strategies, retransmission, resilient transmission.**

I. Introduction

WIRELESS networks provide only limited support for the quality of service (QoS) required by delay-sensitive and high-bandwidth multimedia applications as they provide dynamically varying resources in terms of available bandwidth, due to multipath fading, co-channel interference, and noise disturbances. A variety of application-layer solutions have been proposed to cope with these challenges. These include scalable video coding, rate adaptation, (rate-distortion optimized) scheduling, error-resilience techniques, error-concealment mechanisms, and joint source-channel coding. An excellent review of application-layer research in wireless multimedia streaming is provided in [1]. Cross-layer design for wireless multimedia transmission has also been investigated (e.g., [5], [11], and [15]) and the results indicate that a significant gain in performance can be obtained. However, existing cross-layer solutions often overlook the important issue of packetization and its relationship to other protection strategies at various layers, as well as its impact on the distortion and delay performance at the application layer.

In this paper, we focus on developing content-based flexible and adaptive packetization strategies for three-dimensional (3-D) wavelet-encoded video bitstreams and corresponding medium access control (MAC) retransmission strategies to enable optimal distortion-resilience-delay tradeoffs for wireless multimedia streaming. We develop these joint packetization-retransmission schemes, in a delay-constrained setting, using a cross-layer optimization approach, where the application and MAC layers collaborate to jointly determine the optimal packet sizes and retransmission limits.

A plethora of application-layer packetization strategies have been developed for various video compression schemes. Rogers and Cosman [3] proposed *ad-hoc* strategies of grouping compressed wavelet image codeblocks into packets for improved resilience. Wu, Cheng and Xiong [4] designed optimal strategies to minimize packetization overheads due to bitstream alignment and studied the performance of these schemes against packet erasure at different bit-rates. However, they did not consider any protections offered by the other layers of the OSI stack. Flexible packetization of nonscalable video such as H.264 using a network adaptation layer (NAL) has also been proposed [5]. A similar NAL could also be implemented for scalable video coders, such as the wavelet video coder studied in this paper. However, these application-layer packetization techniques do not consider the protection and adaptation strategies available at the lower layers and do not allow for easy multimedia adaptation based on the channel conditions.

The problem of optimized packetization has also been addressed at the lower layers of the protocol stack. For instance, the error-control parameters such as FEC, ARQ, packet length, and PHY modulation, are optimized based on the network conditions. Qiao and Choi [6] express the effective “goodput” of

Manuscript received March 1, 2005; revised April 12, 2006. 3.This work was supported by NSF Grant CCF 0634988 and UC Micro. The associate editor coordinating the review of this paper and approving it for publication was Dr. Wenjun Zheng.

M. van der Schaar is with the Department of Electrical and Computer Engineering, University of California at Los Angeles (UCLA), Los Angeles, CA 90095-1594 USA (e-mail: mvanderschaar@ece.ucdavis.edu; mihaela@ee.ucla.edu).

D. S. Turaga is with the IBM T. J. Watson Research Center, Hawthorne, NY 10532 USA (e-mail: turaga@us.ibm.com).

Color versions of Figs. 1, 3, and 4, are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2006.886384

TABLE I
DECODED PSNR FOR PACKET SIZE OPTIMIZED AT MAC LAYER

p_e	**Fixed Packet Size** **L= 500 bytes** **Decoded PSNR (dB)**	**Fixed Packet Size** **L= 1000 bytes** **Decoded PSNR (dB)**	**Optimized Scheme** **L* determined by MAC** **Decoded PSNR (dB)**
6×10^{-6}	32.16	30.25	30.08 (L*=2249 bytes)
1×10^{-5}	30.45	28.32	27.90 (L*=1738 bytes)
3×10^{-5}	28.76	25.56	25.86 (L*=997 bytes)
5×10^{-5}	25.01	24.09	24.12 (L*=768 bytes)

an 802.11 system as a closed-form function of the data payload length, the frame retry count, the wireless channel conditions and the data transmission rate, and use this to select the best physical layer (PHY) mode for transmitting data. However, this work does not consider the content characteristics and, as will be shown in this paper, results in suboptimal performance for multimedia.

In [7], some initial work has been presented on cross APP-MAC-PHY layer adaptation for wireless multimedia streaming that explicitly considers adaptive packetization. However, the proposed packetization strategy is *ad-hoc* and uses limited information about the video content, the deployed compression scheme, and the relative importance and dependencies among the various packets.

The problem of joint source-channel coding (JSCC) has been already discussed in a large number of papers, including [18], [19], [21]. An excellent review of various existing JSCC techniques can be found in [20]. The retransmission schemes investigated in our paper may be viewed as a special class of channel codes, and hence, in this paper, we build on prior research results by considering content-based optimal packetization strategies at the application layer in conjunction with adaptive retransmission limits at the MAC layer in a cross-layer manner. However, it should be noted that the MAC layer retransmission exhibits significant benefits as opposed to conventional Adaptive Repeat reQuest (ARQ) schemes deployed at the application-layer, as the delays incurred are significantly lower when implemented at the MAC and the retransmission schemes are already build into existing MAC protocols, such as the 802.11 wireless local-area network (LAN) protocol. Furthermore, we also investigate the use of content-based distortion propagation models for the video to drive our optimization, and reduce the complexity of the proposed scheme. For the video compression, we deploy state-of-the-art wavelet-based video coding techniques. Specifically, we use the scalable interactive video (SIV) codec developed by Secker and Taubman [2] that employs JPEG-2000 like entropy coding for the compression of the spatio-temporal subbands. The SIV codec provides significant benefits for wireless video transmission due to its inherent rate-scalability that enables on-the-fly adaptation to changing channel conditions and easy prioritization of video packets for unequal error-protection purposes. Note, however, that the proposed cross-layer solution can also be deployed for other video coders (e.g., MPEG-4, H.264 and other state-of-the-art wavelet video coders such as 3-D ESCOT [22]) and will also result in performance improvements. However, while the focus of our paper is not on a particular video-coding scheme, and we focus on proposing a content-based optimized joint packetization and retransmission for wireless video transmission under delay constraints, the specifics of the deployed video coder will have a significant impact on the investigated cross-layer optimization as they affect the distortion propagation, delay constraints, resilience to losses, etc.

This paper is organized as follows. Section II formulates the delay constrained joint MAC-Application problem that we address in this paper. We start by motivating the need for cross-layer optimization by highlighting the suboptimality of deploying a packetization scheme optimized solely at the MAC layer, without considering the content distortion or video packet's deadlines for real-time transmission. Next, we provide a brief description of the deployed SIV video coder and introduce the notion of reorganizing the SIV bitstream into deadline layers. We also present a multitrack hinting algorithm that can be used to achieve the desirable property of reorganizing the SIV bitstream on-the-fly, based on the channel conditions or desired application delay, without requiring actual bitstream reorganization. Next, we formulate our cross-layer optimization problem. To enable the cross-layer optimization, we rely on content-based distortion propagation models to determine the distortion impact of the various packets. These distortion models are discussed in Section III. Section IV discusses the proposed analytical solution for the cross-layer optimization and Section V presents a real-time cross-layer algorithm for delay-sensitive wireless video transmission. Section VI presents our obtained results using different real-time cross-layer algorithms. We present our conclusions and directions for future research in Section VII.

II. JOINT MAC-APPLICATION LAYER OPTIMIZATION PROBLEM

A. *Motivation for Cross-Layer Optimization—a Simple Packetization Example*

In this section, we highlight the need for joint application-MAC layer optimization by evaluating existing adaptive packetization algorithms currently deployed at the MAC layer, which do not explicitly consider the video applications delay constraints and distortions [6].

Current 802.11 wireless LANs use orthogonal frequency-division multiplexing (OFDM) to transmit the data symbols, and provide different PHY modes with different modulation schemes and code rates. Assume that the overhead (in terms of bits) that is added to the packet size from the various OSI layers

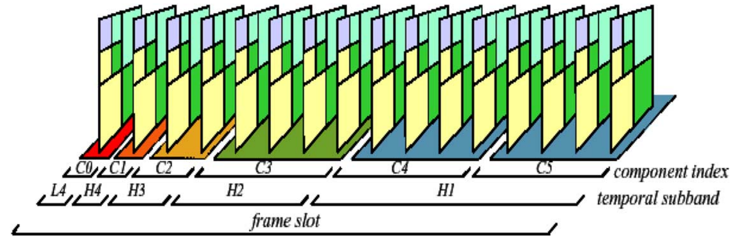

Fig. 1. JPEG2000 code-stream components: four temporal and two spatial decompositions levels.

(PHY, MAC, Network, Transport, Application) is grouped into one overhead that is common to all packets. We label this overhead L^{Header}. (As in [6], [26], L^{Header} reflects the frame header as well as the protocol overhead necessary to send a packet in a practical implementation.) Since the MAC is agnostic to the bitstream distortions or the video application delay constraints, the optimization at this layer is simply aimed at maximizing the throughput. Given the channel signal-to-noise ratio (SNR) and the PHY modes, the optimal packet size L^* that maximizes the throughput is computed in [6] and [26] as

$$L^* = \frac{L^{\text{Header}}}{2} + \frac{1}{2}\sqrt{(L^{\text{Header}})^2 - \frac{4b(L^{\text{Header}})^2}{\ln(1-P_s)}} \quad (1)$$

where b is the number of bits per symbol, and P_s is the probability of symbol error, which will depend on the modulation type and link SNR [6], [26]. However, this packet-size optimization mechanism does not consider either the distortion-impact of the various packets or the video delay constraints. Illustrative results, comparing the decoded PSNR obtained with this optimal packet size versus alternative *ad-hoc* schemes with fixed packet sizes are summarized in Table I. The results were averaged over ten different runs. These results are for the *Coastguard* sequence (at CIF resolution 30 frames/s) that was compressed using the SIV codec [13] and for an application-layer delay constraint of 400 ms. Furthermore, the header overhead L^{Header} was 30 bytes (240 bits). In all scenarios, the retransmission limits have been set to zero.

From Table I, it can be clearly concluded that the optimal packet-size determined at the MAC layer results in a suboptimal performance in terms of the decoded video quality. This motivates the need for cross-layer optimization involving both the channel conditions, but also *explicitly* considering the content characteristics and video encoder features as well as the delay constraints, when determining the packet sizes and the associated retransmissions.

B. Brief Description of the Deployed Video Coder and Multitrack Hinting Approach

As mentioned in the introduction, for the compression of the video data we use the SIV codec [13] that exhibits excellent scalability features needed for robust wireless transmission as well as very good R-D performance. For the temporal filtering, the SIV codec uses a lifting-based implementation of the 5/3 wavelet filters, and for the spatial transform it uses the 9/7 wavelet filters. The resulting spatio-temporal subbands are embedded within the JPEG2000 codestream syntax [14] allowing the codec to leverage the syntax and flexibility existing in the JPEG2000 implementation. In an SIV bitstream, multiple temporal subbands can be grouped into one component (packet). An example grouping of spatio-temporal subbands from [13] is shown in Fig. 1.

Each subband is further divided into codeblocks which are independently decodable units. The codeblocks are encoded into a layered block bitstream for SNR scalability (see [13] for more details). To packetize the SIV bistream, we used the strategy recommended by the JPEG-2000 standard [14], where multiple quality layers that contain incremental contributions from each code-block's layered bitstream, are created (labeled as JPEG-2000 packets). More details on JPEG-2000 packetization may be obtained from [14]. However, in order to enable the delay-sensitive transmission of video packets, we reorganize the bitstream into *deadline-layers*. We illustrate this bitstream reorganization in Fig. 2.

In Fig. 2, we consider a simple case with data from seven codeblocks organized into three quality layers. Each codeblock (labeled B_i) is organized into a layered bitstream, and contributes a certain number of bits to each quality layer, shown in the figure as the height of the bar. However, in a delay-constrained scenario, each codeblock also has an associated decoding deadline with it. In the example depicted in the figure we have five codeblocks with one deadline, and two codeblocks with another deadline (that is after the deadline for the first set of codeblocks). This decoding deadline may be determined based on the encoding parameters (spatio-temporal decomposition structure etc.), the spatio-temporal subband to which the codeblock belongs, and the application layer tolerable delay. We discuss how to compute these deadlines in Section II-C.

We then reorganize the bitstream to collect codeblocks with the same deadline together and label the collection of bits from different codeblocks within one layer (having the same deadline) as one *Deadline-Layer*. This reorganization is performed using information about the codeblocks from the JPEG-2000 packet headers. Hence, in the figure, from three quality layers with two sets of deadlines we generate six deadline-layers. In general, if there are K deadlines, and Q quality layers, we reorganize the bitstream into $K \times Q$ deadline-layers. We then adaptively packetize the bitstream based on these deadline-layers. Hence, data with a common decoding deadline could be jointly packetized, even if it originates from different temporal subbands.

We now briefly discuss a possible solution for hinting the SIV video bitstream that facilitates the real-time adaptive packetization, scheduling and retransmission based on instantaneous

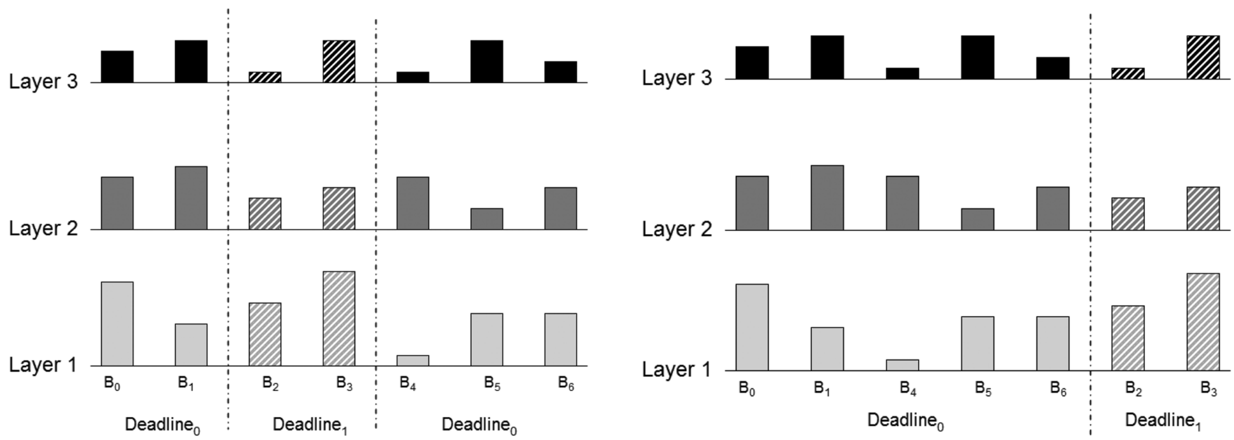

Fig. 2. Reorganization of SIV bitstream Layers (left) into Deadline-Layers (right).

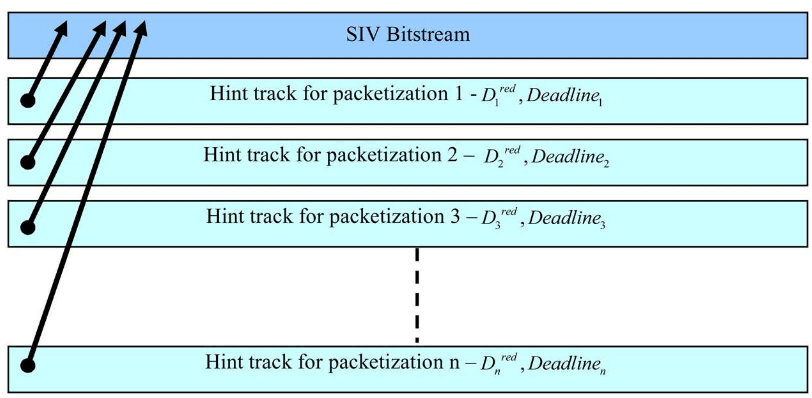

Fig. 3. Deployed multitrack R-D hinting file format.

channel conditions and decoding deadlines (without actually requiring the reorganization of the bitstream). This file storage format is based on our proposal in [16], and introduces an *abstraction* layer referred to as "multitrack hinting", which is an extension of the MP4 file format hinting mechanism [9]. We use the multitrack hinting concept to structure the SIV bitstream into multiple sublayers with different distortion impacts (see Section II-D) and delay constraints (see Section II-C), as illustrated in Fig. 3.

The multitrack concept is useful for wireless multimedia transmission because it enables: i) real-time adaptation of the packet sizes at transmission time, after the encoding has been performed; ii) real-time prioritization of different packets based on distortion impacts and changing delay constraints; and 3) real-time optimized scheduling of video packets based on their deadline and the transmission of the previous packets (see Section IV).

C. Computing Delay Deadlines—an Illustrative Example

Motion-compensated temporal filtering (MCTF) is used to filter frames within a group of pictures (GOP) into temporal low-pass L and high-pass H frames. During this process, different temporal filters such as the Haar or the 5/3 filters (corresponding to bi-directional filtering) may be used. Additionally, the process is applied recursively to the subsequently-produced L frames thereby creating a temporal decomposition pyramid. An example of such a temporal decomposition for a GOP with eight frames is shown in Fig. 4.

In Fig. 4 the frames are decomposed into $T(=3)$ temporal levels. We use the notation $H^{t,k}$ to indicate the kth H frame of temporal level t, where $1 \leq t \leq T$ and $0 \leq k < 2^{T-t}$. Equivalently, the notation $L^{T,0}$ is used to indicate the remaining L frame at the last level after the completion of the temporal decomposition in the GOP. It is important to notice that the example of MCTF illustrated in Fig. 4 represents only one instantiation out of the many possible.

In this illustrative example, the filtered frames $L^{3,0}$, $H^{3,0}$, $H^{2,0}$, and $H^{1,0}$ should be available in the receiving buffer before the decoder can reconstruct and play back the original frame $A^{0,0}$. This implies that these frames in the temporal decomposition have the same deadline (corresponding to the decoding deadline of frame $A^{0,0}$) before which all of their video packets should arrive in the receiving buffer.

We extend the above example to compute the deadlines for all the filtered frames for a 16 frame GOP, with four temporal levels (this is what we use in all our experiments). If the video is played back such that decoded frame i is displayed at time instant t_i, the deadlines for the corresponding temporally filtered 16 frames in the bitstream are shown in Fig. 5.

Fig. 5 shows the decoding deadlines of each of the frames with and without an extra (application tolerable) delay d seconds. When there is an application-tolerable delay, the deadlines

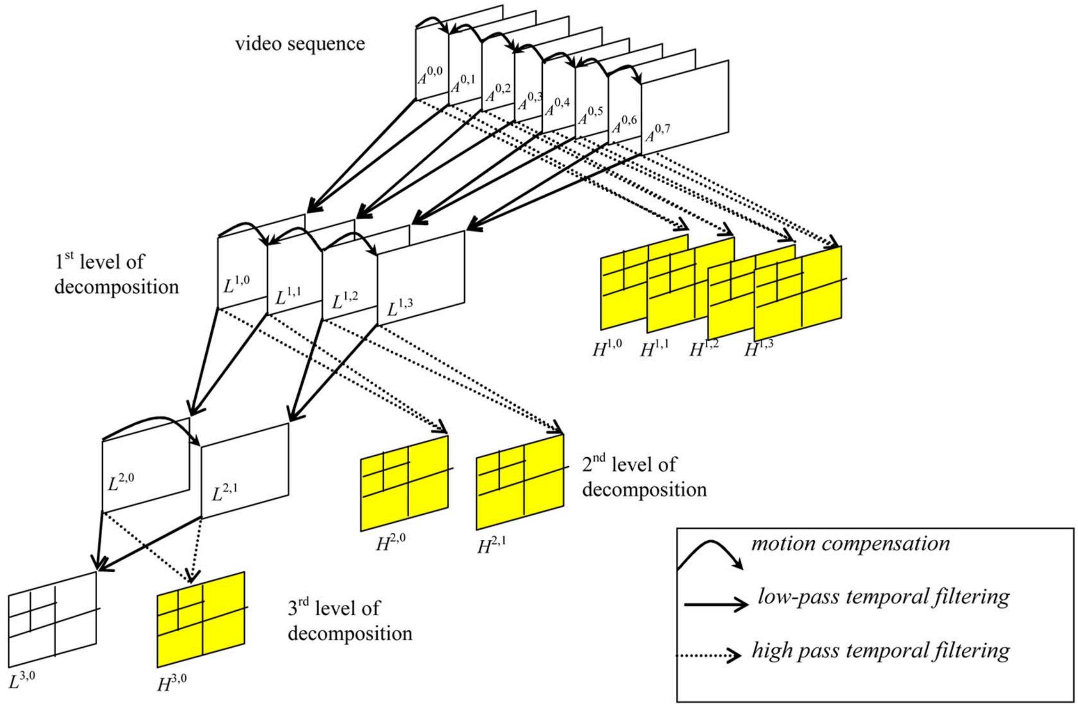

Fig. 4. MCTF decomposition.

for the filtered frames are shifted appropriately. Hence, when the media server schedules and transmits a packet, it should consider the deadline (including the tolerated application delay) of its corresponding video frame.

D. Joint Cross-Layer Optimization Problem

We pose our problem in terms of packetizing (and adaptively retransmitting the obtained packets) the SIV bitstream reorganized into deadline-layers. By first reorganizing the bitstream into these deadline-layers, we ensure that we do not packetize data from different deadline-layers into one packet.

The goal of our cross layer optimization is to determine the optimal packet size L_j and maximum number of times each packet j can be transmitted, $m_j^{\max}$, such that the expected video distortion is minimized, under a given delay constraint. Based on whether packet j is received or lost, the decoded video experiences distortion D_j^{quant} or D_j^{loss}. Hence, when a packet is successfully received, the decoded distortion decreases by an amount D_j^{red}, where $D_j^{\text{red}} = D_j^{\text{loss}} - D_j^{\text{quant}}$. This represents the utility (benefit) of receiving the packet. The goal of our optimization strategy is to maximize the expected utility for a GOP

$$D_{\text{GOP}} = \sum_{j=1}^{N_p} D_j^{\text{red}} P_j^{\text{succ}} \tag{2}$$

with N_p being the number of packets within a GOP and P_j^{succ} being the probability of successfully receiving packet j, given the bit-error probability P_e, subject to a delay constraint

$$\sum_{k=1}^{j} \text{Time}_k \leq \text{Deadline}_j, \quad 1 \leq j \leq N_p \tag{3}$$

where Time_j is the actual time it takes to transmit packet j and Deadline_j is the time deadline for the packet to be received at the application layer of the client in order to be decoded and displayed[1]. The deadlines are determined based on the coding dependencies between frames (and thus, the encoding structure and parameters) and also include the maximum delay tolerated at the application layer $\text{Delay}_{\max}$. We further examine the impact of this tolerable delay $\text{Delay}_{\max}$ on the decoded video quality, in the results section.

In the considered wireless video transmission scenario, there are two reasons for discarding packets: due to packet loss from the BER in the wireless link, and due to exceeded packet transmission deadlines. We define P_j^{succ} as the probability of successfully receiving the packet given bit errors in the network. With packet size L_j (bits) and bit-error probability P_e (controlled by the physical layer, based on the channel SNR, channel coding and modulation strategy used etc.), the packet loss probability is $P_{L_j} = 1-(1-P_e)^{L_j}$. Furthermore, if we assume that the wireless link is a memoryless packet erasure channel [6], such that the packets are dropped independently, we can calculate the probability of success for packet j with an upper limit on the number of transmissions $m_j^{\max}$ (i.e., a retransmission limit $m_j^{\max} - 1$) as

$$P_j^{\text{succ}} = \sum_{k=1}^{m_j^{\max}} (P_{L_j})^{k-1}\left(1 - P_{L_j}\right) = 1 - (P_{L_j})^{m_j^{\max}}. \tag{4}$$

The goal of our packetization and retransmission assignment policy is to solve the delay-constrained optimization problem defined by (2) and (3). Two important differences exist between the conventional JSCC optimization and our optimization. First, in JSCC, the optimal channel codes are determined

[1]Note that we do not transmit any packets of the GOP beyond the deadline of the last packet within the GOP to avoid any impact on future GOPs, since in the SIV codec GOPs are treated as independent entities.

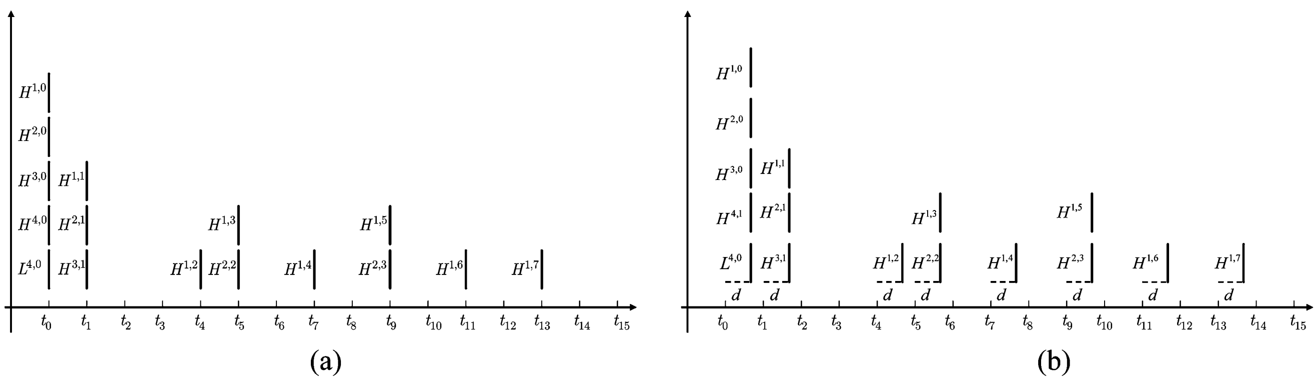

Fig. 5. Decoding deadlines for a 16–frame GOP (a) without additional delay and (b) with additional delay.

given channel rate-constraints, while we are focusing on the delay-constrained transmission scenario for adaptive MAC retransmission and packetization. Second, due to the MAC-layer feedback implemented within the 802.11 wireless protocol, we have access to timely information about the lost packets and the actual time that was needed for transmitting a packet $\mathrm{Time}_{\mathrm{act}}$. Hence, we perform our cross-layer optimization using on-line algorithms that combine real-time information with expected packet loss information, unlike in the conventional JSCC schemes that are deployed at the application layer. Furthermore, the work in this paper also differs from the framework for rate-distortion optimal delivery of streaming scalable media proposed by Chou *et al.* [24], [25] for both FEC and retransmission with deadlines. This is because we rely on the implemented MAC retransmission strategies to consider the actual transmissions for our cross-layer optimization rather than considering the current transmission and hypothetical future retransmission to optimize the expectation of a Lagrangian cost function. Hence, rather than modeling the effect of different transmission policies on the properties of a Markov decision process [20], [24] and finding the strategy that maximizes the expectation of the video quality over all possible paths, we propose to explicitly consider the features of state-of-the-art wireless LAN protocols and develop a cross-layer solution where:

i) the feedback is considered to be immediate, such that coding dependencies are guaranteed to be satisfied, thereby avoiding the polynomial optimization objective encountered in [20] and [24];
ii) the approach is greedy, in the sense that all resources can be consumed in transmitting data which has one deadline, before considering the transmission of data with a later deadline.

III. Content-Based Distortion-Propagation Models

To determine the distortion impact of any packet on the decoded video GOP, we need to consider the propagation of this distortion across the spatio-temporal decomposition tree. Since it is computationally expensive to determine the actual distortion impact exhaustively for each packet, we propose instead to use distortion models that are determined based on content characteristics.

In this paper, we use distortion propagation models that were derived in [12], and have been shown to accurately capture the variations in content as well as motion characteristics of the sequence[2]. In this model, the basic idea is to determine for each video sequence specific low-level features such as the average signal variance, and based on these to predict the distortion impact for different packet losses. At run-time, we use these predicted distortions to determine the optimal cross-layer strategy. Several research papers on developing distortion propagation models for 3-D wavelet schemes already exist, including [8], [14], and [17]. The work in [12], extended prior work to develop a unified mathematical model that describes the operational behavior of motion-compensated wavelet video coders for different encoder settings. There are two parts involved in the modeling: 1) develop a distortion propagation model within one frame and 2) develop a distortion propagation model across frames, by tracking the propagation of quantization noise along the 3-D wavelet decomposition trees. Furthermore, the model parameters are tuned depending on the sequence content and motion characteristics.

The spatial distortion propagation within each frame is derived as in [14]. For a J-scale 2-D spatial domain wavelet transform, with $3J+1$ subbands, let the kth $(k=1,2,3)$ orientation in scale j be denoted as subband (j,k), and the coarsest representation subband be (J). The average distortion in the frame caused by quantization of its subband coefficients can be determined as

$$\mathbf{d} = 4^{-J} G_J \varepsilon_J + \sum_{j=1}^{J} \sum_{k=1}^{3} 4^{-j} G_{j,k} \varepsilon_{j,k} \quad (5)$$

where $G_{j,k}$ is the synthesis gain and $\varepsilon_{j,k}$ is the quantization noise associated with subband (j,k). Hence, by determining the quantization error $\varepsilon_{j,k}$ associated with a subband or within the wavelet coefficients encapsulated in a video packet, the distortion impact on the frame can be determined. We can also use this equation in the loss case, where we replace $\varepsilon_{j,k}$ with the energy of the subband being discarded. Similarly, to determine the distortion impact of a single packet loss, we only need to determine

[2]Note that the SIV codec also uses distortion models to perform rate allocation across these different quality layers. While the SIV spatial distortion propagation model is very accurate, the temporal propagation model does not explicitly consider irregular motion fields (with several multiple-connected and unconnected pixels). Hence, we use instead, the models proposed in [12].

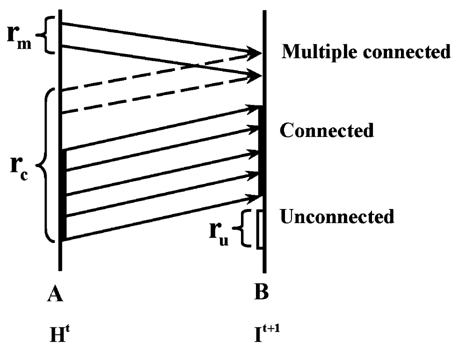

Fig. 6. Pixels classification into three types: multiple connected, connected, and unconnected [8].

which subband it belongs to and replace $\varepsilon_{j,k}$ with the contribution of this packet to the energy of the coefficients within it[3].

We now describe the modeling of temporal distortion propagation across different temporal levels for the $t+2\mathrm{D}$ structure of 3-D wavelet video coders. State-of-the-art wavelet video coders typically use the Haar filter and 5/3 filter for MCTF. First, we discuss the simplest filter (the Haar filter), because it was used in prior work [8], and then present the generalized results for longer filters, such as the ones used in the SIV coder. If the even and odd frames in the temporal lifting structure are labeled A and B, the corresponding high-pass filtered frame H has the same time location as frame A and the low-pass frame I has the same time location as frame B. During motion estimation, pixels can be classified into three types: connected, unconnected, and multiple connected. We show an example of this in Fig. 6 (which is taken from [8]).

Let r_c be the fraction of connected pixels, r_u be the fraction of unconnected pixels, and r_m be the fraction of multiple connected pixels. It is well known, and can be seen from Fig. 6, that $r_c + r_m = 1$ and $r_u = r_m$. For connected pixels, a motion vector can be determined which maps the motion trajectory from frame A to frame B with the inverse motion vector mapping the motion trajectory back from frame B to frame A. Multiple connected pixels may be treated identically to the connected pixels [8].

The work in [12] extends the derivation proposed in [8]. Using that, we can compute a recursive relationship that captures the average distortion of I frames in the $(k-1)$th temporal level given the distortion of I and H frames at the kth temporal level as

$$\overline{\mathbf{d}}_I^{(k-1)} = \left(\frac{3}{4} - \frac{\overline{r}_c(k)}{4}\right)\mathbf{d}_H^{(k)} + \left(\frac{1}{2}\right)\overline{\mathbf{d}}_I^{(k)} \tag{6}$$

where the superscript (k) denotes the kth temporal level. In the above equation, $\mathbf{d}_H^{(k)}$ is the observed distortion in the high-pass H frame, and $\overline{\mathbf{d}}_I^{(k)}$ is the distortion in the low-pass frame (that may be derived from the inversion of a previous temporal level). Iterating through all the temporal levels (T), we can obtain the average distortion in the decoded video frames i.e., at level $k=0$ (after inverse temporal filtering) as

$$\overline{\mathbf{d}}_I^{(0)} = \sum_{k=1}^{T}\left(\frac{3}{4} - \frac{\overline{r}_c(k)}{4}\right)\left(\frac{1}{2}\right)^{k-1}\mathbf{d}_H^{(k)} + \left(\frac{1}{2}\right)^{T}\overline{\mathbf{d}}_I^{(T)} \tag{7}$$

where $\mathbf{d}_H^{(k)}$ and $\overline{\mathbf{d}}_I^{(T)}$ are calculated using (5). Note that this derivation can be considered as an approximation for the cases where subpixel interpolation is used for motion estimation.

The lifting structures for longer filters such as the 5/3 and 9/7 filters are much more complicated than that for the Haar filter, which makes it almost impossible to track the distortion along the temporal wavelet decomposition tree. In order to simplify the analysis, we model the resultant average distortion of I frames in the (k)th temporal level as a linear combination of the distortions of the I and H frames at the $(k+1)$th temporal level. This is a reasonable assumption if the distortions from the two frames are uncorrelated. More specifically, we model

$$\overline{\mathbf{d}}_I^{(k)} = W_I^{k+1}\overline{\mathbf{d}}_I^{(k+1)} + W_H^{k+1}\mathbf{d}_H^{(k+1)} \tag{8}$$

where the weights W_H^{k+1} and W_I^{k+1} are determined empirically and depend heavily on the lifting structures, deployed motion estimation method, and sequence motion characteristics. Hence, the average distortion for the original video frames may be computed as

$$\overline{\mathbf{d}}_I^{(0)} = \sum_{k=1}^{T} W_H^k \prod_{n=1}^{k-1} W_I^n \mathbf{d}_H^{(k)} + \prod_{n=1}^{k-1} W_I^n \overline{\mathbf{d}}_I^{(T)}. \tag{9}$$

Equation (7) is a special form of (9). In particular, for the Haar example, these weights are $W_H^{k+1} = ((3/4) - (\overline{r}_c(k)/4))$ and $W_H^{k+1} = (1/2)$.

In general, for an arbitrary temporal decomposition structure, the weights W_H^{k+1} and W_I^{k+1} are first trained using experimental data, and once this is done the model is then used to make predictions of operational distortion propagation in real-time, for various coding parameters (see [12] for more details). In this paper, the model was trained separately for sequences with different levels of motion, in particular for one GOP each from the *Coastguard* and *Foreman* sequences. These trained models were used in the results section to compute the content-based distortion propagation for the various deadline-layers and as a result, for the packets themselves.

IV. Packetizing and Retransmitting Data With Common Deadlines

In order to derive a solution to our optimization problem, we rely on prior JSCC research. There have been several different approaches, including [18]–[21], to solve the related problem of optimizing distortion for a (scalable) source, under a constrained total transmission length, by adjusting the channel code redundancy appropriately. The authors in [21] present a general Lagrangian optimization formulation to derive the optimal code-length and redundancy for a family of convex and nonconvex sources. Our solution will be derived based on their general formulation.

[3]This information about the contribution of a packet to the energy of the subband coefficients may be extracted from the SIV bitstream that computes it for each codeblock during encoding R-D optimization.

Consider the problem of solving the cross-layer optimization for all deadline-layers with one common deadline. First, we show that we can map this problem of delay-constrained transmission into a rate-constrained transmission. Based on our discussion in Section II-B there are Q deadline-layers with a common decoding deadline, which we label *Deadline*. We want to partition each deadline-layer into packets (one or more) and determine the optimal retransmission strategy for this set of packets. Let one such partitioning lead to a total of $\hat{N}$ packets (for this set of deadline layers). Furthermore, consider that packet j, with packet size L_j [4], is transmitted m_j times. Then, given the physical layer transmission rate $\mathrm{Rate}_{\mathrm{PHY}}$, the time to transmit this packet may be computed as

$$\mathrm{Time}_j = m_j \left(\frac{L_j}{\mathrm{Rate}_{\mathrm{PHY}}} + \mathrm{Time}_O \right) \tag{10}$$

where Time_O is the timing overhead for the 802.11 MAC protocol (necessary to send a packet in a practical implementation), which can be approximated based on [22] and includes the time of waiting for acknowledgements, duration of empty slots, expected backoff delays for transmitting a frame etc. Given that all packets have the same deadline, we may rewrite the delay constraint (from Section II-D) on the packet transmission as

$$\sum_{j=1}^{\hat{N}} m_j \left(\frac{L_j}{\mathrm{Rate}_{\mathrm{PHY}}} + \mathrm{Time}_O \right) \leq \mathrm{Deadline} \tag{11}$$

or, equivalently

$$\sum_{j=1}^{\hat{N}} m_j \left(\frac{\hat{L}_j}{\mathrm{Rate}_{\mathrm{PHY}}} \right) \leq \mathrm{Deadline} \tag{12}$$

where we include the time overhead as an equivalent packet length overhead. We can further rewrite this as

$$\sum_{j=1}^{\hat{N}} m_j \hat{L}_j \leq \mathrm{Rate}_{\mathrm{PHY}} \times \mathrm{Deadline} = L_{\max}. \tag{13}$$

Hence, our delay-constrained optimization becomes

$$\max \left[\sum_{j=1}^{\hat{N}} D_j^{\mathrm{red}} P_j^{\mathrm{succ}} \right] \text{ subject to } \sum_{j=1}^{\hat{N}} m_j \hat{L}_j \leq L_{\max}. \tag{14}$$

This is similar to the formulation of Thie and Taubman [21], where m_j corresponds to the redundancy rate associated with packet k. However, since the actual value of m_j cannot be determined analytically without actually transmitting the packet (it is a particular instance of an underlying random process), we consider the *expected* redundancy rate, in terms of the expected number of transmissions of the packet. For packet j with a transmission limit $m_j^{\max}$ the expected number of times the packet is transmitted is

$$\begin{aligned} \overline{m}_j &= \sum_{i=1}^{m_j^{\max}} i \left(1 - P_{L_j}\right) \left(P_{L_j}\right)^{i-1} + m_j^{\max} (P_{L_j})^{m_j^{\max}} \\ &= \frac{1 - (P_{L_j})^{m_j^{\max}}}{1 - P_{L_j}} = \frac{P_j^{\mathrm{succ}}}{1 - P_{L_j}}. \end{aligned} \tag{15}$$

Using a similar Lagrangian formulation as in [20] and [21], we can rewrite our optimization functional as

$$F = \sum_{j=1}^{\hat{N}} \left(D_j^{\mathrm{red}} P_j^{\mathrm{succ}} - \lambda \hat{L}_j \overline{m}_j \right) \tag{16}$$

where $\lambda\ (\geq 0)$ is the Lagrange multiplier. The goal of the optimization is thus to maximize F. This problem may be further decomposed into a set of $\hat{N}$ independent optimizations for the packets, where the goal is to optimize the individual cost functional

$$F_j = \left(P_j^{\mathrm{succ}} - \lambda_j \overline{m}_j \right), \text{ with } \lambda_j = \lambda \frac{\hat{L}_j}{D_j^{\mathrm{red}}}. \tag{17}$$

A direct consequence of the result reported in [20] and [21] to our problem is that the optimal solution may be obtained based on the convex hull of the probability of success P_j^{succ} versus the expected redundancy rate $\overline{m}_j$ curve. Note that the actual packet length L_j parameterizes this curve. In particular, the optimal solution ($m_j^{\max,\mathrm{opt}}$, L_j^{opt}) is obtained on the curve at the point with the maximum redundancy, where the slope of the curve is larger (or equal) than the parameter λ_j. More details on determining the optimal λ using e.g., a bisection search, and the corresponding optimal retransmission limit and packet size may be obtained from [20].

Comparing the derived expressions for P_j^{succ} and $\overline{m}_j$ for the considered cross-layer optimization, we can clearly see that they have a linear relationship (with a slope $1-\dot{P}_{L_j}$). Hence, for such a linear curve, the optimal limit on the number of transmissions $m_j^{\max,\mathrm{opt}}$ for packet j will be ∞, when $1 - P_{L_j} \geq \lambda_j$ and 0 otherwise, i.e., either transmit a packet until it is received, or do not transmit the packet at all. This is an important result, which indicates that the optimum retransmission policy is to retransmit as often as needed $\left(m_j^{\max,\mathrm{opt}} = \infty\right)$ the most important packets corresponding to high distortion impacts and to not transmit the less important packets at all.

Using the above analysis, we can develop a real-time algorithm to tune the retransmission limit based on the actual number of packet transmissions (instead of the expected redundancy rate). After analytically determining the optimal packet size[5] and the maximum number of packet transmissions (as ∞ or 0), we sort the set of packets in decreasing order of the fraction $\left(\left(1 - P_{L_j^{\mathrm{opt}}}\right)/\lambda_j\right)$. The packets are then transmitted in this order, where no packet is transmitted before all preceding packet transmissions are either completed or terminated. This ensures that coding dependencies between the layers are maintained and the additive distortion model is not violated. Since, in our delay-constrained wireless video transmission, the maximum number of times a packet j can be transmitted

[4]Given that all bits within a deadline-layer have roughly the same importance, we partition each deadline-layer into equal parts for packetization. Hence, the packet sizes L_j are the same for all packets within one deadline-layer. Furthermore, we assign the same retry limit to packets from the same deadline-layer.

[5]Note that the packet size is upper-bounded by the size of the deadline layers.

TABLE II
PROPOSED REAL-TIME GREEDY ALGORITHM FOR ADAPTIVE PACKETIZATION AND MAC RETRANSMISSION

Set $Time_{cur} = 0$.
Compute the decoding deadlines for each subband, and hence each codeblock, based on the encoding parameters, and tolerable application delay. Let there be K separate deadlines (with values $Deadline^k$*)*
Reorganize (hint) the SIV bitstream into deadline-layers.
Sort the deadlines in ascending order.
for k=1:K
Gather all deadline-layers with deadline ($Deadline^k$ *).*
Determine instantaneous channel conditions P_e *and PHY modulation strategy* $Rate_{PHY}$.
Solve the equivalent rate-constrained optimization using the probability of success versus expected redundancy rate curve to determine optimal λ*, and determine optimal packet sizes and initial retransmission limits (as* ∞ *or 0);*
Packetize data using these obtained packet sizes $L_j^{opt,k}$.
Sort the packets in descending order of $\dfrac{1-P^k_{L_j^{opt,k}}}{\lambda_j^k}$
for j=1: $\hat{N}^k$
Tune the actual retransmit limit $m_j^{\max,opt,k} = \left\lfloor \dfrac{Deadline - Time_{cur}}{\left(\dfrac{L_j^{opt,k}}{Rate_{PHY}} + Time_o\right)} \right\rfloor$.
Transmit the packet to determine the actual number of transmissions m_j^k ($m_j^k \le m_j^{\max,opt,k}$).
Set $Time_{cur} \leftarrow Time_{cur} + m_j^k \left(\dfrac{L_j^{opt,k}}{Rate_{PHY}} + Time_o\right)$
If $Time_{cur} + \left(\dfrac{L_{j+1}^{opt,k}}{Rate_{PHY}} + Time_o\right) > Deadline^k$
break
end
end

cannot actually be ∞, and is bounded by the delay deadline *Deadline* (which in this section is assumed to be the same for all packets), we tune this limit for each packet based on the actual number of observed transmissions that occurred before it, as

$$m_j^{\max,\text{opt}} = \left\lfloor \frac{\text{Deadline} - \sum_{k=1}^{j-1} m_k \left(\frac{L_k^{\text{opt}}}{\text{Rate}_{\text{PHY}}} + \text{Time}_o\right)}{\left(\frac{L_j^{\text{opt}}}{\text{Rate}_{\text{PHY}}} + \text{Time}_o\right)} \right\rfloor \tag{18}$$

where $\lfloor \bullet \rfloor$ is the floor operation.One additional advantage of computing this limit in real-time is that we can recompute the retransmission limits (and also packet sizes if necessary) when the channel condition P_e or used PHY modulation strategy (that determines Rate_{PHY}) change. Hence, for data belonging to different deadline-layers with a common deadline, we can determine the optimal packet size as well as the retransmission limit using the above analysis. In Section V, we extend this approach to the case when we have deadline-layers with different decoding deadlines for the real-time transmission scenario.

V. PROPOSED REAL-TIME CROSS-LAYER ALGORITHM FOR WIRELESS VIDEO STREAMING

In this section, we extend the analysis in Section IV to include sets of packets with different deadlines, as is the case in typical video streaming scenarios. One approach to solve this cross-layer optimization is to formulate it as a joint optimization problem (optimization across different deadlines, quality layers etc.) like performed in [20], [24]. However, the complexity of such an algorithm increases rapidly with the number of different deadlines that need to be considered, especially as each transmission impacts all future transmissions. Additionally, such a joint optimization would require that assumptions are made about future channel conditions, modulation strategies employed, etc.

Instead, we use a real-time greedy approach, which is based on the analysis in the previous section, but which has the benefits of simplicity, as well as of enabling real-time instantaneous adaptation to varying channel conditions or PHY modulation strategies. In this greedy approach, we solve the optimization problem (to determine the optimal packet size and the retransmission limit) independently for each set of deadline-layers with a common deadline. Note that our approach does not consider the benefits of transmitting packets (deadline-layers) with a late deadline before packets with an earlier deadline (which might be advantageous in some cases). However, a major advantage of 3-D wavelet video coders such as SIV is that the packets with the largest distortion impact are mostly transmitted with an early decoding deadline due to the hierarchical temporal structure deployed, and hence, the greedy algorithm is likely to perform close to the optimal solution. Our real-time greedy algorithm is outlined in more detail in Table II.

TABLE III
OPTIMIZATION STRATEGIES CONSIDERED

<table>
<tr><th>Strategy</th><th>Packetization</th><th>Retransmission Limit</th><th>Transmission</th></tr>
<tr><td>Heuristic</td><td>Packetize SIV bitstream into fixed size packets, determined based on the MAC throughput optimization (Section 2.1).</td><td>Assign retransmission limits[7] 10, 5, 1 and 0 to all packets belonging to temporal level 3, 2, 1, and 0 correspondingly (increasing temporal level corresponds to coarser temporal resolution).</td><td>Transmit in order in which packets are encoded</td></tr>
<tr><td>InfZero</td><td>Reorganize the SIV bitstream into deadline-layers. Packetize each deadline-layer into fixed size packets, with size determined based on the MAC throughput optimization (Section 2.1).</td><td colspan="2">Joint Strategy
Given packet size, solve the optimization problem (for each set of packets with the same deadline, using the probability of success versus the expected redundancy curve) to determine which packets get retransmission limit ∞ and which packets get 0. Transmit packets in increasing order of deadline, and in decreasing order of $\frac{1-P_{L_j}^k}{\lambda_j^k}$. Packets are transmitted only if their deadline is not violated by previous transmissions. Furthermore, any packet that can be transmitted is retransmitted until it is successfully received (even past its deadline).</td></tr>
<tr><td>Online Retrans (OR)</td><td>Reorganize the SIV bitstream into deadline-layers. Packetize each deadline-layer into fixed size packets, with size determined based on the MAC throughput optimization (Section 2.1).</td><td colspan="2">Joint Strategy
Given packet size, solve the optimization problem (for each set of packets with the same deadline, using the probability of success versus the expected redundancy curve) to determine which packets get retransmission limit ∞ and which packets get 0. Transmit packets in increasing order of deadline, and in decreasing order of $\frac{1-P_{L_j}^k}{\lambda_j^k}$. Packets are transmitted only if their deadline is not violated by previous transmissions. The actual number of transmissions is used to tune in real-time the retransmission limits of the packets such that they are not transmitted past their deadline.</td></tr>
<tr><td>Online Packet Size + Retrans (OPR)</td><td colspan="3">Joint Strategy
Follow the greedy algorithm described in Table 2. Determine optimal packet size and retransmission limit jointly by solving the optimization problem (for each set of packets with the same deadline). Transmit packets in increasing order of deadline, and in decreasing order of $\frac{1-P_{L_j}^k}{\lambda_j^k}$. Packets are transmitted only if their deadline is not violated by previous transmissions, and not transmitted past their deadline. Use the actual number of transmissions to tune the retransmission limits in real-time.</td></tr>
</table>

In Table II, the superscript k corresponds to a group of packets having the same deadline $\mathrm{Deadline}^k$. In Section VI, we examine the performance of such a scheme under different loss and delay scenarios and compare it against alternative algorithms.

VI. SIMULATION RESULTS AND DISCUSSION

We present the performance of the proposed real-time cross-layer algorithms for the video sequences *Coastguard*, *Foreman*, and *Mobile* at CIF resolution and 30 frames/s. We consider the application layer delay $\mathrm{Delay}_{\max}$ to take various values from very low delays[6] (250 ms) to larger delays (up to 2 s) in order to highlight the performance of our algorithms under different transmission scenarios. We use a GOP structure with 16 frames with four levels of temporal decomposition, and five levels of spatial decomposition. We encode the SIV bitstream into ten quality (rate) layers. The overheads and adaptive modulation strategies are determined as in [6] and [7].

[6]Note that the encoding was performed offline, and the abovementioned delays are only transmission delays.

We compare our proposed cross-layer optimization scheme against a heuristic algorithm that uses the MAC packetization strategy and assigns retransmission limits solely based on the temporal level that the packet belongs to. Additionally, we present different sets of results to highlight the incremental gains achievable with the real-time tuning of the retransmission limit, and the use of the optimal packet sizes. Specifically, we consider four algorithms whose features we describe in Table III.

In Table III, we collapse columns corresponding to packetization, retransmission limit and transmission whenever the algorithm determines these jointly. The OPR algorithm corresponds to the greedy algorithm described in Section V. By comparing the Heuristic algorithm with the InfZero algorithm, we can determine the incremental benefit of determining the retransmission limit using a JSCC optimization framework versus assigning it heuristically, and transmitting packets in the order of their deadlines. Next, by comparing the InfZero algorithm with the OR algorithm, we can measure the benefits of tuning the retransmission limit online, based on the specific delay deadline, by using information about the actual transmission. Finally, we

TABLE IV
FOREMAN SEQUENCE: PSNR (16 FRAME GOP) IN dB

Bit-Rate	Bit Error Rate	$Delay_{\max}$ (sec)	Heuristic	InfZero	OR	OPR
768 Kbps	10^{-4}	2.0	33.05	33.41	33.41	33.92
		1.0	32.21	32.98	33.02	33.76
		0.5	31.16	32.52	32.64	33.35
		0.25	29.27	31.28	31.55	32.84
	10^{-5}	2.0	33.47	33.64	33.63	34.09
		1.0	32.73	33.37	33.42	33.76
		0.5	32.02	32.91	33.02	33.34
		0.25	31.09	32.15	32.27	32.91
1024 Kbps	10^{-4}	2.0	33.62	33.72	33.84	34.47
		1.0	32.58	33.17	33.19	33.92
		0.5	31.35	32.60	32.72	33.53
		0.25	29.28	31.31	31.63	32.84
	10^{-5}	2.0	33.95	34.09	34.27	34.70
		1.0	33.01	33.54	33.68	33.91
		0.5	32.21	32.93	33.12	33.55
		0.25	31.15	32.20	32.25	32.89

TABLE V
COASTGUARD SEQUENCE: PSNR (16 FRAME GOP) IN dB

Bit-Rate	Bit Error Rate	$Delay_{\max}$ (sec)	Heuristic	InfZero	OR	OPR
768 Kbps	10^{-4}	2.0	32.03	32.63	32.64	33.44
		1.0	31.35	32.20	32.24	33.28
		0.5	29.66	31.65	31.76	32.89
		0.25	28.64	30.93	31.15	32.12
	10^{-5}	2.0	32.69	32.91	32.91	33.78
		1.0	31.98	32.47	32.52	33.52
		0.5	31.30	32.08	32.13	33.16
		0.25	30.22	31.26	31.47	32.59
1024 Kbps	10^{-4}	2.0	32.30	32.92	33.13	33.85
		1.0	31.47	32.44	32.51	33.66
		0.5	29.83	31.80	31.85	32.94
		0.25	28.64	30.99	31.20	32.15
	10^{-5}	2.0	33.06	33.36	33.58	34.46
		1.0	32.09	32.69	32.86	33.55
		0.5	31.43	32.13	32.28	33.08
		0.25	30.31	31.31	31.54	32.59

can determine the benefit of jointly determining the packet size and the retransmission limit by comparing the OPR algorithm against the OR algorithm.

We present decoded PSNR results for different bit-error rates, and different tolerable application layer delays $\mathrm{Delay}_{\max}$ in Tables IV –VI. The results are averaged across 100 runs of an independent error-generating process.

As expected, while the decoded quality decreases, the gains for the optimized algorithms (as compared against the Heuristic algorithm) increase with tighter delay constraints. An important observation we make is that the gains of increasing the decoding bit-rate are limited by the decoding delay constraints. Hence, with a tight (250 ms) delay constraint, there is virtually no gain in the decoded PSNR whether we transmit data encoded at 1024 or 768 kbps. This is because the original bit-rate matters little when the delay tolerance is small, and the proposed algorithm effectively truncates the original stream by assigning 0 transmissions to some packets. As we relax the delay constraint, the gains from encoding at higher bit-rate become more apparent, however they are still not as high as the actual coding gain (decoding with no loss at these bit-rates). As can be seen from the tables, the InfZero algorithm outperforms the Heuristic algorithm by up to 1.5 dB (at 250 ms). The real-time tuning further improves the performance by around 0.2–0.3 dB, and finally, the joint determination of the packet size and retransmission limit leads to an additional improvement of up to 1.3 dB. Overall, the OPR algorithm outperforms the Heuristic algorithm by over 2 dB, highlighting the need for such an optimization strategy.

VII. CONCLUSIONS

In this paper, we propose a cross-layer (joint application and MAC) optimized packetization and retransmission strategy for delay-sensitive multimedia transmission over wireless networks. We first motivate the need for cross-layer optimization and conclude that both the packetization and retransmission strategies need to be optimized jointly based on the distortion impact and delay constraints of the various packets. We formulate this joint optimization problem in terms of maximizing the expected utility per GOP given delay constraints. Our proposed solution is derived using the analysis presented in the more general formulation for JSCC proposed in [20] and [21]. We extend that analysis specific to the delay-constrained problem

TABLE VI
MOBILE SEQUENCE: PSNR (16 FRAME GOP) IN dB

Bit-Rate	Bit Error Rate	$Delay_{max}$ (sec)	Heuristic	InfZero	OR	OPR
768 Kbps	10^{-4}	2.0	28.68	29.29	29.42	30.11
		1.0	27.11	28.34	28.53	29.46
		0.5	25.54	27.41	27.68	28.73
		0.25	24.34	26.63	26.91	27.88
	10^{-5}	2.0	29.47	29.78	29.82	30.37
		1.0	28.20	28.66	28.77	29.58
		0.5	27.23	27.98	28.15	29.00
		0.25	26.54	27.67	27.86	28.62
1024 Kbps	10^{-4}	2.0	29.69	29.91	30.03	30.41
		1.0	28.34	28.73	28.92	29.67
		0.5	27.24	28.00	28.30	29.18
		0.25	26.56	27.70	27.89	28.66
	10^{-5}	2.0	29.77	29.87	30.05	30.52
		1.0	28.28	28.72	28.83	29.75
		0.5	27.30	28.12	28.35	29.18
		0.25	26.59	27.70	27.89	28.62

(as that provides a solution for a rate-constrained problem) and tune the analytical solution in real-time based on the available MAC-feedback and changing channel conditions.

Summarizing, the main conclusions of this paper are threefold. First, an analytical solution can be determined based on prior joint source-channel coding research, for a special case of the considered cross-layer optimization problem, i.e., when all packets have the same decoding deadline. Under such a scenario, the optimal cross-layer strategy results in retransmitting the most important packets (subbands) as often as needed and discarding the lesser important packets. Second, we use the above analysis to develop a real-time greedy algorithm to perform cross-layer optimization for the case when different sets of data have different decoding deadlines. This proposed greedy algorithm can successfully take advantage of the available feedback at the MAC about the actual number of times previous packets have been transmitted to correctly determine the number of times the current packet can be retransmitted. Moreover, this algorithm can also successfully adapt on-the-fly to the changing channel conditions or physical layer modulation strategies. Finally, the proposed algorithm achieves significant improvements of 2 dB or more for a variety of video sequences, transmission bit-rates and delay constraints.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers that pointed out that it is possible to derive an analytical solution to the proposed cross-layer optimization based on prior joint source-channel coding research and provided numerous other essential comments. The paper has been improved significantly as a result of their comments.

REFERENCES

[1] B. Girod and N. Farber, "Wireless video," in *Compressed Video Over Networks*, M. T. Sun and A. R. Reibman, Eds. New York: Marcel Dekker, 2001.

[2] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform framework for highly scalable video compression," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1530–42, Dec. 2003.

[3] J. Rogers and P. Cosman, "Wavelet zerotree image compression with packetization," *IEEE Signal Process. Lett.*, vol. 11, no. 5, pp. 105–107, May 1998.

[4] X. Wu, S. Cheng, and Z. Xiong, "On packetization of embedded multimedia bitstreams," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 132–40, Mar. 2001.

[5] S. Wenger, "H.264/AVC over IP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 645–56, Jul. 2003.

[6] D. Qiao, S. Choi, and K. G. Shin, "Goodput analysis and link adaptation for IEEE 802.11a wireless LAN," *IEEE Trans. Mobile Comput.*, vol. 1, no. 4, pp. 580–589, Jun. 2002.

[7] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE J. Select. Areas Commun.*, vol. 21, no. 10, pp. 1752–1763, Dec. 2003.

[8] T. Rusert, K. Hanke, and J. Ohm, "Transition filtering and optimization quantization in interframe wavelet video coding," in *VCIP, Proc. SPIE*, 2003, vol. 5150, pp. 682–93.

[9] D. Singer, W. Belknap, and G. Franceschini, ISO Media File Format Specification—MP4 Technology Under Consideration for ISO/IEC 14496-1:2002 Amd 3 2001, Committee Draft, ISO/IEC JTC1/SC29/WG11 MPEG01/N4270-1.

[10] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. ACM SIGCOMM'96*, 1996, pp. 117–130.

[11] A. Majumdar, D. G. Sachs, I. V. Kozintsev, and K. Ramchandran, "Multicast and unicast real-time video streaming over wireless LANs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 524–534, Jun. 2002.

[12] M. Wang and M. van der Schaar, "Rate-distortion modeling for wavelet video coders," in *Proc. ICASSP*, Philadelphia, PA, Mar. 2005.

[13] D. Taubman, M. Reji, D. Maestroni, and S. Tubaro, SVC Core Experiment 1—Description of UNSW Contribution 2004, MPEG document m11441.

[14] D. S. Taubman and M. W. Marcellin, *JPEG 2000-Image Compression Fundamentals, Standards and Practice*. Norwell, MA: Kluwer, 2002.

[15] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 278–290, Apr. 2004.

[16] ——, A Flexible Streaming Architecture for Efficient Scalable Coded Video Transmission Over IP Networks 2002, ISO/IEC JTC 1/SC 29/WG 11/M8944.

[17] G. Feideropoulou and B. Pesquet-Popescu, "Stochastic modeling of the spatio-temporal wavelet coefficients—Application to quality enhancement and error concealment," *EURASIP J. Appl. Signal Process.*, vol. 2, pp. 1931–1942, 2004.

[18] R. Puri, K.-W. Lee, K. Ramchandran, and V. Bharghavan, "Forward error correction (FEC) codes based multiple description coding for internet video streaming and multicast," *Signal Process.: Image Commun.*, vol. 16, no. 8, pp. 745–762, May 2001.

[19] A. E. Mohr *et al.*, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 819–828, Jun. 2000.

[20] J. Thie, Optimal Erasure Protection Assignment for Scalably Compressed Data Over Packet-Based Networks 2003, Ph.D. dissertation, Univ. New South Wales. Sydney, Australia.

[21] J. Thie and D. Taubman, "Optimal erasure protection assignment for scalable compressed data with small packets and short channel codewords," *EURASIP J. Appl. Signal Process.*, no. 2, pp. 207–219, Feb. 2004.
[22] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
[23] J. Xu, Z. Xiong, S. Li, and Y. Zhang, "Three-dimensional embedded subband coding with optimized truncation (3-D ESCOT)," *Appl. Commun. Harmonic Anal.*, vol. 10, pp. 290–315, 2001.
[24] P. A. Chou and Z. Miao, Rate-Distortion Optimized Streaming of Packetized Media 2001, Microsoft Research Tech. Rep. MSR-TR-2001-35.
[25] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra, "FEC and pseudo-ARQ for receiver-driven layered multicast of audio and video," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 440–449.
[26] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, "Cross-layer design of *ad-hoc* networks for real-time video streaming," *IEEE Wireless Commun. Mag.*, vol. 12, no. 4, pp. 59–65, Aug. 2005.

Mihaela van der Schaar (SM'04) received the Ph.D. degree from Eindhoven University of Technology, The Netherlands, in 2001.

She is currently an Assistant Professor in the Electrical Engineering Department at the University of California, Los Angeles (UCLA). Prior to this, she was between 1996 and June 2003 a Senior Researcher at Philips Research in the Netherlands and USA, where she led a team of researchers working on multimedia coding, processing, networking, and streaming algorithms and architectures. She has published extensively on multimedia compression, processing, communications, networking and architectures and holds 28 granted U.S. patents and several more pending.

Dr. van der Schaar was an active participant to the ISO Motion Picture Expert Group (MPEG) standard since 1999, to which she made more than 50 contributions and for which she received three ISO recognition awards. She was also chairing for three years the ad-hoc group on MPEG21 Scalable Video Coding, and also co-chairing the MPEG ad-hoc group on Multimedia Test-bed. She was also elected as a Member of the Technical Committee on Multimedia Signal Processing and of the Technical Committee on Image and Multiple Dimensional Signal Processing, both of the IEEE Signal Processing Society. She was an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA and *SPIE Electronic Imaging Journal*. Currently, she is an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEM FOR VIDEO TECHNOLOGY (T-CSVT) and of IEEE Signal Processing Letters. She received the NSF CAREER Award in 2004, IBM Faculty Award in 2005, Okawa Foundation Award in 2006 and the Best Paper Award for her paper published in 2005 in the IEEE T-CSVT.

Deepak S. Turaga (M'01) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, in 1997 and the M.S. and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1999 and 2001, respectively.

He is currently a Research Staff Member in the Exploratory Stream Processing Department, IBM T. J. Watson Research Center, Hawthorne, NY. He was with Philips Research during 2001–2003 and with Sony Electronics during 2003–2004. His research interests lie primarily in statistical signal processing, multimedia coding and streaming, machine learning, and computer vision applications. In these areas, he has published over 30 journal and conference papers and one book chapter. He has also filed over 15 invention disclosures and has participated actively in MPEG standardization activities.

Dr. Turaga is an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA.