Content-Based Cross-layer Packetization and Retransmission Strategies for Wireless Multimedia Transmission

M. van der Schaar[1] and D. S. Turaga[2],
[1]Electrical and Computer Engineering, University of California, Davis, CA 95616
[2]IBM T. J. Watson Research Center, Hawthorne, NY 10532
mvanderschaar@ece.ucdavis.edu, turaga@us.ibm.com

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 data applications (e.g., file transfers), but has considerable consequences for multimedia applications and often leads to unsatisfactory user experience. Recently, the research focus has been to adapt existing algorithms and protocols at the lower layers of the network 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 packetization and MAC-layer retransmission strategy. First, we show that packet-size optimizations solely determined at the MAC-layer result in a sub-optimal performance in terms of the multimedia quality. Subsequently, we propose cross-layer strategies that optimize the packetization, prioritization and retransmission strategies based on content characteristics, channel conditions, and the specific features of the deployed video coder. Finally, we investigate the use of content based distortion models for the video, to reduce the complexity of our proposed optimization.

1. 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 multi-path fading, co-channel interference, and noise disturbances. A variety of application-layer solutions have been proposed to cope with these challenges. These include 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. [7][11][15]) and the results indicate that a significant gain in performance can be obtained. It is however important to note that 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 rate-distortion (R-D) performance at the application-layer.

In this paper, we focus on developing content-based flexible and adaptive packetization strategies for scalable multimedia streams and corresponding Medium Access Control (MAC) retransmission strategies to enable optimal rate-distortion-resilience tradeoffs for wireless multimedia streaming. We develop these joint packetization-retransmission schemes using a cross-layer optimization approach, where the application layer collaborates with the MAC layer 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. Similar techniques of grouping codeblocks into packets will be deployed in our approach, with the key difference being that our solution will explicitly consider the resulting R-D performance due to joint packetization-MAC

retransmission under different channel conditions. 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. In this paper, we also investigate the overheads associated with different packetization strategies and the impact on performance at the application-layer. Flexible packetization of non-scalable video such as H.264 using a network adaptation layer (NAL) has also been proposed [5]. A similar NAL could also be implemented for the studied wavelet video coder. 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 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 PHY mode for transmitting data. However, this work does not consider the content characteristics and, as will be shown in this paper, results in sub-optimal 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 very limited information about the video content, the deployed compression scheme, and the relative importance and dependencies among the various packets.

Alternatively, in this paper, we build on prior research results and improve them 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. Furthermore, we also investigate the use of content based R-D 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 SIV codec developed by Secker and Taubman [2] that employs JPEG-2000 like entropy coding for the compression of the spatio-temporal subbands. Note, however, that the proposed cross-layer solution can also be deployed for other coders (e.g. MPEG-4 or H.264) and will also result in distortion performance improvements. Hence, the focus of our paper is not on a particular coding scheme, but rather on proposing a content-based optimized joint packetization and retransmission for wireless video transmission.

This paper is organized as follows. Section 2 motivates the need for cross-layer optimization by presenting results obtained by deploying an optimized packetization scheme determined solely by the MAC layer, without considering the content distortion. Subsequently, we describe the SIV codec bitstream in Section 3 and describe how simple, adaptive packetization strategies can be designed based on the specific features of the scalable SIV codec. We present the proposed content-aware R-D optimized application-layer packetization and MAC retransmission limit adaptation in Section 4. Section 5 uses content-based operational R-D models for determining the impact of packet-losses and guiding the packetization-retransmission optimization. We present our conclusions and directions for future research in Section 6.

2. Performance analysis of optimal packet sizes (content independent) determined at the MAC

In this section, we highlight the sub-optimal performance that is obtained when the MAC solely decides the packet-size for video applications. As in [6], we assume that the noise over the wireless medium is white Gaussian with spectral density $N_0/2$.

Current 802.11 based systems use orthogonal frequency division multiplexing (OFDM) to transmit the data symbols, and provide eight different PHY modes (number from 1 through 8) with different modulation schemes and code rates. The frame error probability for a frame of size L bits using the PHY mode m is a function of bit error rate (BER) p_e [1]. The probability of error p_L in a packet of length L bits, assuming random bit errors is:

$$p_L = 1-(1-p_e)^L$$

Let the overhead (in terms of bits) that is added to the packet size from the various OSI layers (PHY, MAC, Network, Transport, Application) be grouped into one overhead that is common to all packets and that we label L^{Header}. Since the MAC is agnostic to the content characteristics, one way for it to improve the video quality is to increase the throughput. The expression for throughput as a function of packet error rate is given by:

$$Throughput = \frac{L(1-p_L)}{L+L^{Header}}$$

Differentiating the above expression with respect to L and equating it to 0 we obtain the optimal packet size that maximizes the throughput, as:

$$L^* = \frac{-L^{Header} + \sqrt{\left(L^{Header}\right)^2 - \frac{L^{Heder}}{2\log(1-p_e)}}}{2}$$

The second derivative of the equation is negative suggesting that the above expression for the optimal frame length maximizes the throughput of the IEEE 802.11a system for a fixed PHY mode. While the throughput achieved by this scheme is maximized, the decoded video PSNR is fairly poor. Illustrative results, comparing the decoded PSNR obtained with this optimal packet size versus schemes with ad-hoc packet size selection, are summarized in Table 1. These results are for the Coastguard sequence (at CIF resolution 30Hz) that was compressed using the SIV codec [13]. In all scenarios, the retransmission limits have been set to 0.

Table 1.Decoded PSNR for packet size optimized at MAC layer

p_e	**Ad-Hoc Scheme 1: L= 500 bytes Decoded PSNR (dB)**	**Ad-Hoc Scheme 2 L= 1000 bytes Decoded PSNR (dB)**	**Optimized Scheme L^* determined by MAC Decoded PSNR (dB)**
0.000006	32.86	30.65	27.90
0.000010	30.93	28.10	31.20
0.000030	28.76	25.43	26.86
0.000050	24.01	23.09	25.12

From Table 1, it can be clearly concluded that the optimal packet-size determined at the MAC layer, without considering the content characteristics and compression strategies can result in sub-

[1] This is a function of the PHY mode m.

optimal performances depending on the bit-error rates. This is not surprising, since the optimization was only performed with respect to the throughput (rate) and not the distortion, as this information is not available at the MAC. This motivates the need for cross-layer optimization involving both the channel conditions, but also the content and application layer characteristics, when determining the packet sizes.

3. Proposed adaptive packetization strategies for the SIV coder

In this paper, we use the SIV codec [13] that is a t+2D wavelet video coder, which performs the motion-compensated temporal filtering (MCTF) first, followed by 2D Discrete Wavelet Transform (DWT). 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 codestream syntax and flexibility existing in the JPEG2000 implementation. In a SIV codec, multiple temporal subbands can be included in one component (packet). An example grouping of spatio-temporal subbands from [13] is shown in Figure 1.

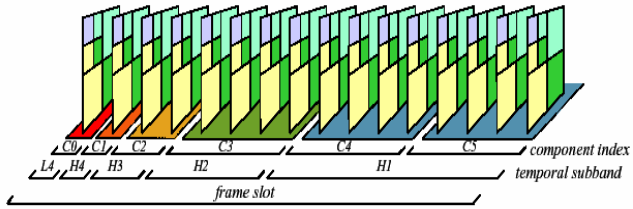

Figure 1. JPEG2000 code-stream components: 4 temporal and 2 spatial decompositions levels.

Each spatio-temporal subband is further divided into codeblocks which are independently decodable units. The codeblocks are coded into a collection of layered block bitstreams for SNR scalability (see [13] for more details). In order to eliminate dependencies between packets, the codeblocks cannot be fragmented across packets. For wireless transmission, the various packets created by the SIV coder are further encapsulated in Application (RTP), Transport (UDP), Network (IP) and MAC packets, which will add additional packetization overheads (see subsequent sections). Note, however, that the flexibility provided by the JPEG2000 codeblocks is mainly aimed at enabling accessibility to region of interests or provide easy packetization for Internet transmission. Hence, it does not necessarily allow on-the-fly adjustment of packet-sizes as required for wireless transmission. As shown in e.g. [6][7], for optimized transmission the packet sizes would need to be adjusted in real-time. The optimal packet sizes vary between 100 bytes to 2000 bytes based on the channel conditions. Figure 1 illustrates for 2 sequences – Foreman and Coastguard, the maximum bytes required by each codeblock and PSNR values at different bit-rates for various codeblock sizes starting with 64×64 codeblocks going down to 8×8 codeblocks.

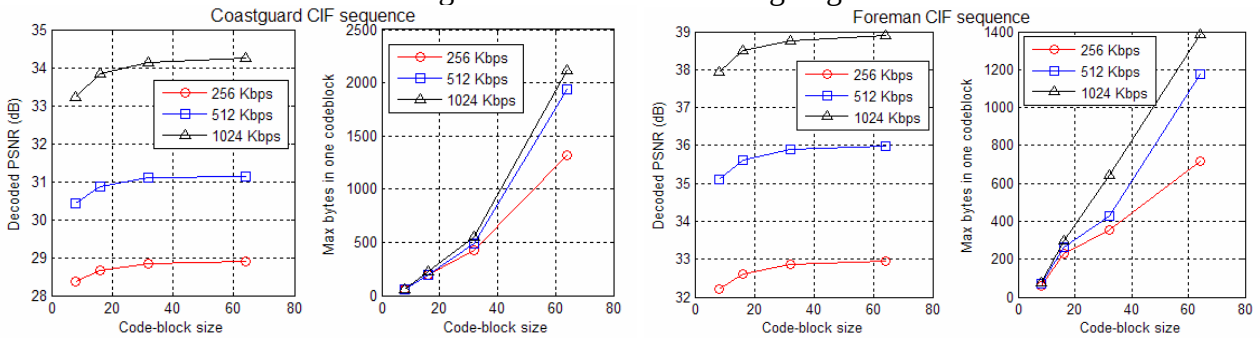

Figure 2. Variation of decoded PSNR and maximum bytes per codeblock with codeblock size

Using smaller sizes for codeblocks reduces the coding efficiency at the application layer and leads to larger packet overheads, but has the beneficial effect of improving the R-D performance under moderate to heavy packet-losses. As can be seen from these tables, using 8×8 sample blocks leads to a reduction in PSNR over using 64×64 sample blocks, which is less than 1dB for a majority of decoding bit-rates and for both these sequences. The goal of our packetization strategy is to determine the optimal size packets for each subband based on the cross-layer optimization strategy proposed in Sections 3-5 and correspondingly adjust the packet-sizes for the SIV coder in real-time, by grouping together multiple codeblocks from the same subband to form packets. To enable such adaptation, we propose to compress the data using 8×8 blocks, and at transmission time, based on the instantaneous channel conditions and available R-D information, encapsulate multiple 8×8 codeblocks into a single packet of the desired size.

One possible solution for implementing the real-time packetization in a wireless streaming scenario is to consider the file format of the media. In [16], we introduced an *abstraction* layer referred to as "multi-track hinting", which is an extension of the hinting mechanism that is part of the MP4 file format specification [9]. We use the multi-track hinting concept to structure compressed video into multiple sub-streams (e.g. spatio-temporal subbands) that can be independently transmitted through multiple (RTP) channels, as illustrated in Figure 3. The multi-track concept is useful for wireless multimedia transmission because it enables (i) adapting the packet size on-the-fly at transmission time, after the encoding has been performed, (ii) prioritization of different video layers (subbands) [15][16] that can assist the cross-layer optimization (e.g. the MAC retransmissions and physical layer modulation strategies can be adapted for each priority layer [7]) and (iii) optimized scheduling and rate adaptation by changing the number of transmitted RTP channels.

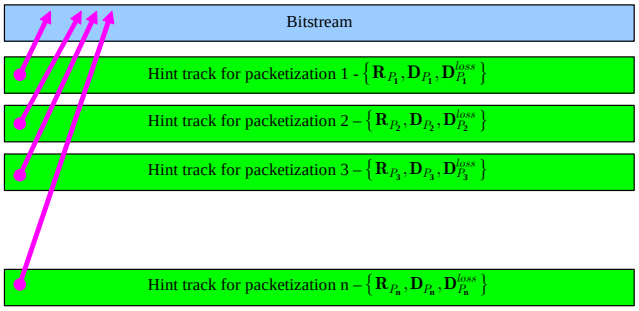

Figure 3. Proposed multi-track R-D hinting file format.

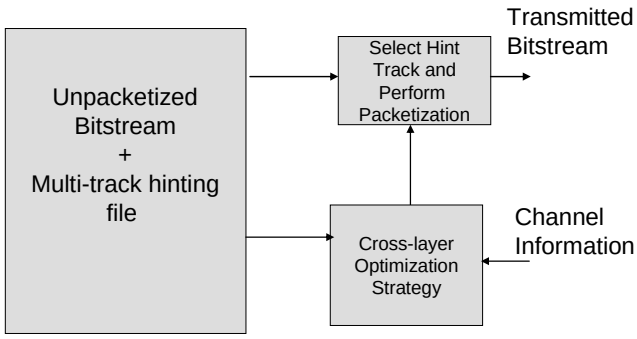

Figure 4. Real-time packetization using the multi-track hinting concept.

This extension to conventional hinting mechanisms provides the flexibility necessary for enabling

real-time adaptive packetization for wireless multimedia streaming (see Figure 4) by storing multiple tracks, containing packet sizes formed by adjusting the number of codeblocks aggregated into one packet (see Figure 3). Using our multi-track hinting method, each bitstream (using 8×8 codeblocks) remains unchanged and it is stored once, but can be virtually divided into multiple streams having different corresponding packetization schemes P_n resulting in different R-D truncation points $\{\mathbf{R}_{P_n}, \mathbf{D}_{P_n}\}$ and distortion impacts under loss for the bitstream $\mathbf{D}_{P_n}^{loss}$ (see next section for more details).

4. Content-aware R-D Optimized Cross-layer Packetization, Prioritization and Retransmission strategies

4.1. Cross-layer optimization problem

For improved multimedia transmission over wireless networks, cross-layer optimization over the various layers of the OSI stack is required. We need to account for both the *content characteristics*, as well as the *channel conditions* in order to make an optimal decision. The cross-layer optimization problem can be formulated as follows:

Determine the optimal joint strategy that minimizes the video distortion, given the channel conditions, i.e. $S^{opt}(\mathbf{x}) = \arg\min_{S} D(S(\mathbf{x}))$ given channel conditions $\mathbf{x} = (p_e, R_c)$, with p_e being the bit error rate, and R_c being the channel bit-rate[2].

In this paper, we investigate content-based packetization and prioritization strategies at the application layer in conjunction with adaptive retransmission limits for each packet at the MAC layer to optimize the multimedia transmission. We show these two layers in the OSI stack and list our employed optimization strategies in Figure 5. Next, we describe the set of adaptation strategies available at the MAC and application layers, and subsequently, we present our proposed cross-layer solution.

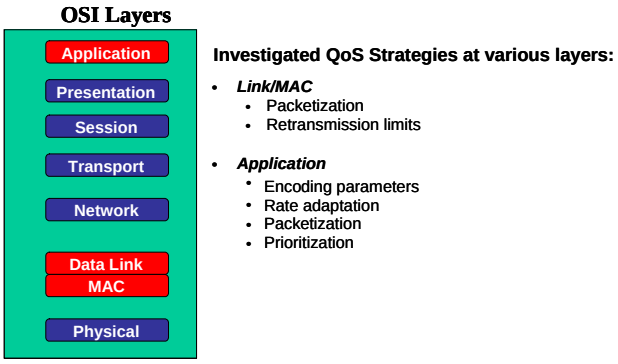

Figure 5. Cross-layer strategies involved in the proposed wireless multimedia optimization framework.

4.2. Adaptive MAC Retransmissions

The MAC layer provides error robustness by adapting the number of retransmission of lost packets (see e.g. previous work in [7][11][15]). Packets that are not received are retransmitted up to a certain maximum number of times $T_{\max}$. The probability that the packet is received in t transmissions, i.e. with t-1 retransmissions is $p_L^{t-1}(1-p_L)$, where p_L is the packet loss probability as defined in Section 2. Hence, with a retry limit $T_{\max}$, the probability that a packet is received is

[2]We can also add constraints on the maximum available computation resources and delay, but these issues are not explored in this paper.

$P(succ) = \sum_{t=1}^{T_{\max}+1} p_L^{t-1}(1-p_L)$, and the probability that the packet is dropped is $P(fail) = 1 - P(succ)$. Hence, the expected number of transmissions for any packet is $\bar{T} = \sum_{t=1}^{T_{\max}+1} t p_L^{t-1}(1-p_L) + P(fail)(T_{\max}+1)$.

As we have shown in Section 2, while the MAC is aware of the channel conditions, packetization overheads and even packet priorities, it cannot independently determine the distortion-optimal retransmission limit. Specifically, the MAC is not aware of the video characteristics, the relative importance and dependencies between packets, the impact of losing specific packets on the quality etc. Hence it can either maximize throughput (as in Section 2) or it can reformulate the problem in terms of minimizing the MAC packet loss rate for all priority layers, which also leads to sub-optimal decoded video quality (see [15]).

In prior work [15], a joint application-MAC algorithm for determining the retransmission strategy for various priority layers has been designed. In that approach, partial information about the content characteristics is passed from the application layer to the MAC in terms of priorities for different packets. To maximize the video quality, the following cross-layer strategy was adopted [15]: higher priority packets were transmitted first and with a higher retransmission limit since they have the highest impact on multimedia quality, while the lower priority packets were discarded or transmitted with a lower number of retransmissions. The number of retransmissions for the various layers was adapted based on the channel conditions, multimedia traffic rates and delay constraints. To enable this cross-layer strategy, multiple priority queues were maintained at the interface between MAC and application layer and different retransmission limits were adapted based on the video layer priority. All the queues were managed by a common absolute Priority-Queuing (PQ) discipline. However, the proposed strategy was MAC-centric, considered a very simple prioritization algorithm of the packets at the application layer, and did not consider R-D impact of the packetization employed. In this paper, we will specifically investigate the additional improvements obtained by deploying a content-based (R-D) optimized solution for retransmission limit adaptation at the MAC.

4.3. Content Based Packetization and Prioritization

The application layer first performs rate adaptation of the multimedia based on the available source bit-rate *R*. Specifically the source rate *R* should be chosen such that the total bit-rate, including any overheads incurred due to retransmission and packetization, does not exceed the channel rate, i.e. $R + overhead \le R_c$ [3]. Then, in order to prioritize the different codeblocks, and packetize them, the application layer has to determine their relative importance in terms of the number of bits required by each codeblock, and the corresponding impact of the codeblock on the decoded video quality.

The SIV codec performs an R-D optimization to determine the bit allocation for each codeblock for the target decoding source bit-rate *R*, such that the decoded distortion is minimized. Based on this R-D optimization, R_{sb} bits are assigned to codeblock *b* in subband *s* with an associated decoded quantization distortion $D_{sb}^{Quant,R}$. This quantization distortion is computed exhaustively as follows: We first quantize only the current codeblock (*s*,*b*) corresponding to the rate of interest, while leaving the other codeblocks unquantized. We then decode all the codeblocks, and the resulting squared error in the decoded video frames is $D_{sb}^{Quant,R}$. Hence, the computed error corresponds to the total distortion due to the quantization of this one codeblock on

[3] In this paper we do not describe how to derive the appropriate *R* from R_c, but, given an *R* we present the obtained overheads and decoded quality under different loss scenarios.

all the decoded frames (i.e. it includes the error propagation across the temporal decomposition tree[4]). The total number of bits assigned to subband *s* is $R_s = \sum_b R_{sb}$ and the distortion in the decoded video due to this subband being quantized is $D_s^{Quant,R} = \sum_b D_{sb}^{Quant,R}$. Similarly, the total distortion in the decoded video, when all the codeblocks in all subbands are quantized to meet this particular source bit-rate *R*, is $D_{total}^{Quant,R} = \sum_s D_s^{Quant,R} = \sum_s \sum_b D_{sb}^{Quant,R}$.

If codeblock *b* in subband *s* is lost during transmission, and is not received by the decoder, there is a different distortion associated with it, and we label this distortion D_{sb}^{loss}. This D_{sb}^{loss} is independent of the decoding bit-rate. As before, we can compute this loss distortion exhaustively, by discarding the current codeblock (e.g. by setting all its coefficients to zero), decoding all other codeblocks with no quantization error, and observing the resulting squared error in the decoded video frames. Thus, the computed error corresponds to the total distortion due to the loss of this one codeblock on all the decoded frames. As before, if an entire subband is lost, the resulting loss distortion is $D_s^{loss} = \sum_b D_{sb}^{loss}$ [5], and if all codeblocks in all subbands are lost, the resulting distortion is $D^{loss} = \sum_b D_s^{loss}$, which is the same as the energy of video frames.

In a real transmission scenario, the resulting total distortion is likely to be a sum of $D_{sb}^{Quant,R}$ for quantized codeblocks and D_{sb}^{loss} for lost codeblocks. Consider that, for subband *s*, we construct packets of size $L_{p,s} = L_{p,s}^{Data} + L^{Header}$ ($L_{p,s} \approx L$) bits without fragmenting codeblocks across packets[6]. Let packet *p* contain a set of codeblocks C_p, i.e. $L_{p,s}^{Data} = \sum_{b \in C_p} R_{sb}$. Then, the quantization distortion associated with this packet is $D_{p,s}^{Quant,R} = \sum_{b \in C_p} D_{sb}^{Quant,R}$, and the loss distortion associated with this packet is $D_{p,s}^{loss} = \sum_{b \in C_p} D_{sb}^{loss}$.

The application layer has knowledge of all the content characteristics, and hence can determine all the rates and distortions for each codeblock. However, it cannot independently perform the packetization, because it lacks information about the channel conditions.

4.4. Cross-Layer Optimization: Solution for Adaptive Packetization and Retransmission Limit Selection

We combine information about the content characteristics as well as the channel conditions to jointly determine the optimal packetization as well as the retransmission limit. Since various subbands have different priorities, we perform the optimization independently for each subband.

Based on the MAC retransmission strategy, a packet is received with probability $P(succ)$ and lost with probability $P(fail)$. Hence, the expected distortion associated with each packet, under lossy conditions, is $\overline{D}_{p,s} = P(succ) \times D_{p,s}^{Quant,R} + P(fail) D_{p,s}^{loss}$. Similarly, the expected number of bits

[4] JPEG-2000 Tier-2 [14] uses distortion models to estimate the quantization error for each codeblock, however it does not consider any temporal propagation of the error, and its spatial models need to be extended before we can use them. We describe preliminary work on building spatio-temporal distortion models for wavelet based video codecs in Section 5.

[5] This distortion is likely to be much higher for low-pass subbands than for the finer resolution high-pass subbands.

[6] In such a case it is not guaranteed that all packets have the same size, however if the codeblocks are small enough, we can generate packets with roughly the same size.

transmitted for this packet is $\overline{T}L_{p,s}$, and the expected number of additional bits that we need to send due to the packetization and retransmissions in this lossy scenario becomes: $\overline{R}_{p,s} = \left(\overline{T} - 1\right)L_{p,s} + L^{Header}$.

We have to determine the optimal packet-size $L_{p,s} \approx L$ and the retransmission limit $T_{\max}$, for this subband so that the total expected distortion is minimized and the rate overhead is also minimized. We can formulate this optimization problem as:

$$\left(T_{\max,s}^{opt}, L_s^{opt}\right) = \arg\min_{(T_{\max},L)} \left[\sum_{p=1}^{P_s} \left(\overline{D}_{p,s} + \lambda \overline{R}_{p,s}\right)\right], \quad (1)$$

where P_s is the total number of packets in subband *s*, and λ is an optimization parameter that determines the desired R-D tradeoff.

4.5. Experimental Results

In our experiments, we use a 4 frame Group Of Pictures (GOP) with two levels of temporal decomposition, and four levels of spatial decomposition, producing 13 spatial subbands per temporally filtered frame, and a total of 52 spatio-temporal subbands. For ease of notation we number these subbands in increasing spatio-temporal resolution order, as shown in Figure 6.

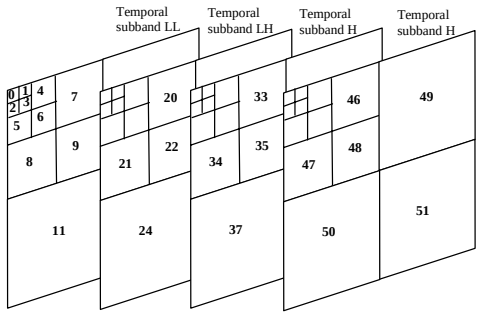

Figure 6. Spatio-temporal subband numbering

We further constrain the SIV codeblocks for each subband to contain data from only one frame, i.e. we do not group frames from different temporal resolutions in one codeblock. The number of bits assigned to every codeblock R_{sb} is computed by the SIV codec using a JPEG-2000 like R-D optimization for every target decoding source bit-rate. Finally, we set the packetization overhead to $L^{Header} = 30$ bytes, which is typical for wireless IP networks**.**

4.5.1. Implementation Issues

Since the number of codeblocks in each frame is large (~1600), determining the quantization and loss distortions for each codeblock exhaustively can be very computationally demanding. Hence, instead, we compute the distortions $D_s^{Quant,R}$ and D_s^{loss}, subband by subband[7], i.e. by quantizing or discarding all codeblocks in the subband and observing the distortion in the decoded video frames. From $D_s^{Quant,R}$ and D_s^{loss} we determine the distortion for each codeblock $D_{sb}^{Quant,R}$ and D_{sb}^{loss} by simply assuming that each codeblock within the subband contributes equally to the observed distortion. This assumption is not necessarily true in all cases, but it is reasonable, and helps us reduce the computations associated with determining these distortions.

[7]Measuring these distortions once per subband, requires only 13 decodings per frame, as opposed to ~1600 if we measure these for each codeblock.

We solve the cross-layer optimization problem of determining $\left(T_{\max,s}^{opt}, L_s^{opt}\right)$ for each subband independently. We use numerical approaches to solve the optimization problem and allow the packet sizes for subband *s* to vary in the interval $\left[\frac{R_s}{10}, R_s\right]$ and the retransmission limits to lie in the interval $[0,10]$.

4.5.2. Optimal packet sizes and retransmission limits

We show the determined optimal retry limits and packet sizes for 4 frame GOPs from Foreman, Football, Mobile and Coastguard CIF sequences, decoded at 1024 Kbps, with a bit-error rate $p_e = 10^{-4}$ in Figure 7-Figure 10.

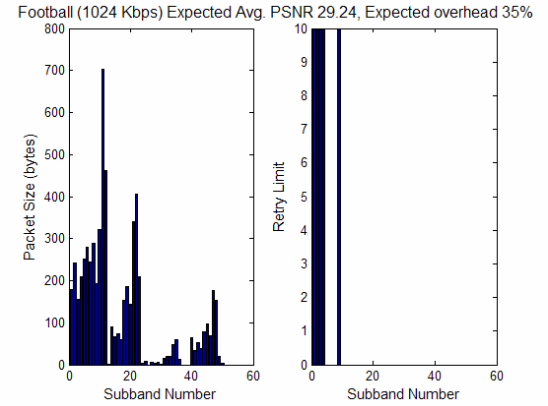

Figure 7. Football: Determined retry limits and packet sizes for four frame GOPs $p_e = 10^{-4}$

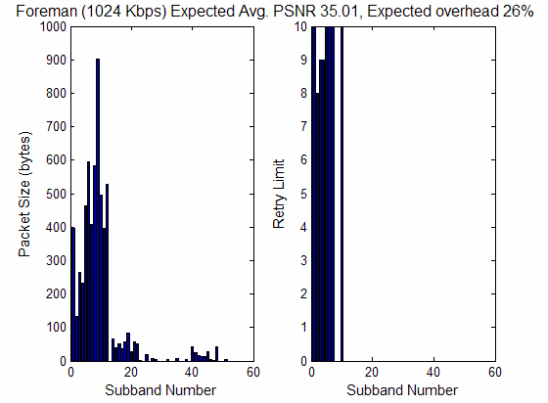

Figure 8. Foreman: Determined retry limits and packet sizes for four frame GOPs $p_e = 10^{-4}$

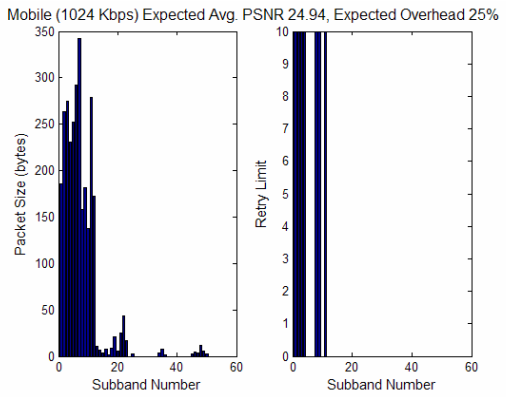

Figure 9. Mobile: Determined retry limits and packet sizes for four frame GOPs $p_e = 10^{-4}$

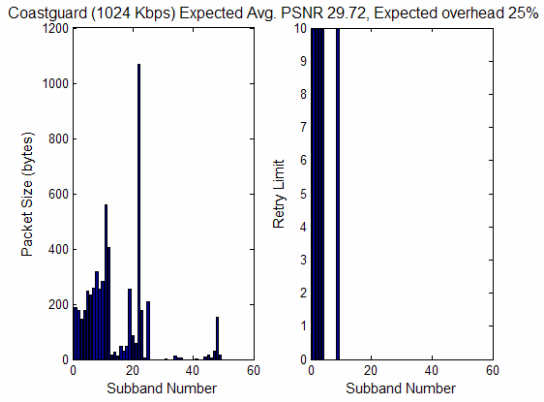

Figure 10. Coastguard: Determined retry limits and packet sizes for four frame GOPs $p_e = 10^{-4}$

The expected PSNR is in dB, and the expected overhead corresponds to the total additional bytes transmitted due to packetization headers and retransmissions, and is computed as a percentage of the original bitstream size. Instead of observing different values for the derived retry limits, we see that they converge to either a large value (>8) for some subbands, or to 0 for the rest. This indicates, *especially for high bit error rates*, subbands worth retransmitting should be transmitted as many times as possible, at the expense of not retransmitting other subbands. These optimal retry limits and packet sizes $\left(T_{\max,s}^{opt}, L_s^{opt}\right)$ are determined for each different decoding source bit-rate R and bit error rate p_e.

After determining the optimal packet sizes and retransmission limits, we test the performance of this scheme under lossy conditions. In order to compare against state-of-the-art packetization schemes, we design a reference scheme that we describe in the following sub-section.

4.5.3. Reference Fixed-Packetization Scheme

Our reference scheme is based on commonly used state-of-the-art wireless multimedia retransmission scheme [7][11][15] that uses layered priority queuing and a common fixed packet size for all the multimedia data in the bitstream. Furthermore, as in [15], we partition the multimedia data into multiple priorities and set higher retry limits for data with higher priority. In particular, we prioritize our 52 spatio-temporal subbands as shown in Table 2. The priorities are set based on the coding dependencies, i.e. to low-pass spatio-temporal subbands a higher priority is assigned than to the higher spatio-temporal subbands.

Table 2. Prioritization of spatio-temporal subbands

Priority	Subband Number
5 (High)	0, 13
4	1-3, 14-16
3	4-6, 17-19, 26, 39
2	7-9, 20-22, 27-29, 40-42
1 (Low)	10-12, 23-25, 30-38, 43-51

In order to achieve a fair comparison against our optimized scheme, we restrict the retry limit to take values either 10 or 0 and we ensure that the number of subbands with retry limit 10 is the same as in the case of the optimized scheme. By doing the above, we implicitly provide our reference scheme with the cross-layer optimized information on the number of subbands to be selected with non-zero retry limit. For instance, as in Figure 7 for the Football sequence, if the result of the optimization indicates 7 subbands with non-zero retry limit, we set the retry limit to 10 for 7 subbands even for the reference scheme. Furthermore, to determine which subbands are assigned a retry limit 10 we select them in order of decreasing priority. Hence, in our example with 7 subbands, we will select subbands 0, 13, 1, 2, 3, 14 and 15 to have a retry limit 10. After we assign retry limits to the different subbands, we determine the fixed packet size (common to all subbands) that leads to roughly the same amount of overhead as expected for the optimized scheme.

Summarizing, the main differences between the reference scheme and the proposed optimization scheme are twofold. First, in the proposed scheme the retransmission limits are determined based on a content-based R-D optimization framework and secondly, adaptive packetization strategies are deployed in conjunction with the retransmission limit adaptation.

4.5.4. Performance Comparison under Lossy Scenarios

We study the performance obtained using the proposed content-based cross-layer optimization solution for 10 GOPs of the following sequences Football, Foreman, Mobile and Coastguard. The performance is determined for $p_e = 10^{-4}$ and $p_e = 10^{-5}$. We present the average decoded PSNR across 100 different error patterns. The results are summarized in Table 3. (The overhead is computed as explained in Section 4.4 based on $\bar{R}_{p,s}$ **.)**

Table 3. Loss Performance: Football

Bit-rate (kbps)	p_e	Scheme	PSNR (dB)	Overhead
1024	10^{-4}	Reference	28.01	35%
		Optimized	29.14	36%
	10^{-5}	Reference	28.67	10%
		Optimized	29.11	9%
1536	10^{-4}	Reference	30.17	33%
		Optimized	31.66	31%
	10^{-5}	Reference	29.89	11%
		Optimized	31.64	11%

Table 4. Loss Performance: Foreman

Bit-rate (kbps)	p_e	Scheme	PSNR (dB)	Overhead
512	10^{-4}	Reference	30.11	26%
		Optimized	30.78	26%
	10^{-5}	Reference	31.56	12%
		Optimized	31.75	13%
1024	10^{-4}	Reference	34.90	27%
		Optimized	35.66	25%
	10^{-5}	Reference	36.89	12%
		Optimized	37.08	12%

Table 5. Loss Performance: Mobile

Bit-rate (kbps)	p_e	Scheme	PSNR (dB)	Overhead
1024	10^{-4}	Reference	24.29	29%
		Optimized	25.10	27%
	10^{-5}	Reference	26.23	17%
		Optimized	26.67	15%
1536	10^{-4}	Reference	26.99	28%
		Optimized	27.87	26%
	10^{-5}	Reference	29.43	17%
		Optimized	29.74	16%

Table 6. Loss Performance: Coastguard

Bit-rate (kbps)	p_e	Scheme	PSNR (dB)	Overhead
512	10^{-4}	Reference	27.54	26%
		Optimized	28.55	26%
	10^{-5}	Reference	29.02	14%
		Optimized	29.51	12%
1024	10^{-4}	Reference	28.91	25%
		Optimized	30.17	23%
	10^{-5}	Reference	31.56	15%
		Optimized	32.11	12%

The sequences Football and Mobile are more complex than Foreman and Coastguard, and hence to achieve ~ 30 dB decoded PSNR, we need to use higher bit-rates for them. Clearly, the optimized scheme with adaptive packet size with retry limit selection outperforms the reference scheme consistently across all the sequences. The largest observed PSNR gain was ~1.5 dB for the same amount of overhead. The advantage of the proposed optimization solution is especially exhibited at higher bit-rates, where the possible packetization and protection strategies are numerous and can impact the distortion to a larger extent.

The gains for the optimized scheme are lower for the Foreman and Mobile sequence because, in both these sequences the high frequency subbands are allotted fewer bits (as may be observed in

Figure 7), and therefore adaptive packet sizes and retry limits do not affect performance significantly.

Finally, to assess the PSNR gains obtained due to use of adaptive packetization, as opposed to optimized retry limit selection, we perform an additional experiment. We use a fixed common packet-size for all subbands, as used in the Reference scheme, but we determine $\left(T_{\max,s}^{opt}\right)$ by solving the optimization problem. We call this the *Partial Optimization* scheme. The results for this scheme are included in Table 7.

Table 7. Partial Optimization versus Optimized and Reference Schemes

(1024 Kbps $p_e = 10^{-4}$)

Seq.	Scheme	PSNR (dB)	Overhead
Foreman	Reference	34.90	27%
	Partial	35.03	27%
	Optimized	35.66	25%
Football	Reference	28.01	35%
	Partial	28.19	36%
	Optimized	29.14	36%

We observe that the results for Partial optimization are very close to the Reference scheme. We can conclude that a significant portion of the gains of the Optimized scheme are derived from the cross-layer optimized packetization strategy. Hence, at run-time, the proposed retransmission limit optimization can be enabled depending on the complexity constraints.

5. Content Model-based cross-layer optimization

While the previously presented distortion optimized cross-layer solution leads to a very good performance, it requires exhaustively determining the distortions (D_{sb}^{loss} and $D_{sb}^{Quant,R}$**)**. Hence, the complexity associated with performing such optimizations in real-time is very high, even if it is done only once per subband. Alternatively, we propose to reduce this computational complexity associated with determining optimal cross-layer strategies for wireless multimedia transmission by using models based on content characteristics. 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 deploy these predicted distortions to determine the optimal cross-layer strategy. Next, we describe the used content-aware distortion model and compare the obtained results with that of the exhaustive R-D optimized cross-layer strategy.

5.1. Content-based distortion models

There has been only very preliminary work on developing models to capture the content characteristics and distortion propagation for 3D wavelet schemes [8][17]. In [12], this work was extended to develop a unified mathematical model that describes the operational R-D behavior of motion-compensated wavelet video coders for different encoding settings. There are two parts involved in the R-D modeling: a) develop an R-D model for one frame; b) develop an R-D model across frames, by tracking the propagation of quantization noise along the 3D wavelet decomposition trees. Importantly, note that these model parameters depend on the sequence characteristics.

To capture the distortion propagation within each frame we base our derivation on [12][14]. For a *J*-scale 2D spatial domain wavelet transform, there are 3*J* +1 subbands. Let the subband of

the *k*-th (*k* = 1, 2, 3) orientation in scale *j* be denoted as (*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:

$$\mathbf{d} = 4^{-J} w_J G_J \varepsilon_J + \sum_{j=1}^{J} \sum_{k=1}^{3} 4^{-j} w_{j,k} G_{j,k} \varepsilon_{j,k} \tag{2}$$

where $G_{j,k}$ is the synthesis gain, $w_{j,k}$ are defined based on the bi-orthogonal wavelet filter [14] 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 we can determine the distortion impact on the frame. We can also use this equation in the loss case, where we replace $\varepsilon_{j,k}$ with the energy of the subband being discarded.

We now consider the propagation of this distortion $\mathbf{d}$ across the temporal decomposition tree. Let us consider the simple case of a Haar motion-compensated temporal filter [8]. In a temporal filtering structure with *T* levels, there are 2^T frames in one group of frames for the Haar filtering case. We assume approximately constant content (signal) variance σ_0^2 within one group of frames and label the even and odd frames in the temporal lifting structure as *A* and *B*. The high pass frame *H* has the same time location as frame *A* and the low pass frame *I* has the same time location as frame *B*. In motion estimation, the pixels can be classified into three types: connected, unconnected and multiple connected. We show an example of this in Figure 11 (which is taken from [8]).

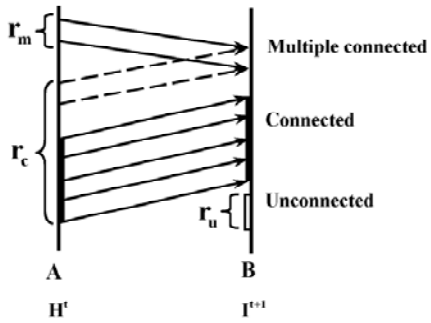

Figure 11. Pixels classification into three types: multiple connected, connected and unconnected.

Let r_c be the ratio of connected pixels, r_u be the ratio of unconnected pixels and r_m be the multiple connected pixels. In the following notations, the superscript (*k*) denotes the *k*-th temporal level. By following the same method as in [8], the average distortion of *I* frames in the *k*-th temporal level is given by:

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

where $\mathbf{d}_H^{(k)}$ 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), both of which

may be computed from the quantization error using equation (2). We may iterate this expression through all the temporal levels to obtain the average distortion in the decoded video frames i.e. at level k=0 (after inverse temporal filtering) as

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

Note that this derivation can be considered as an approximation for the cases where sub-pixel 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 quantization noise along the temporal wavelet tree. Nevertheless, equation (3) shows that we can find a linear relationship between the average frame distortions within adjacent temporal levels:

$$\bar{\mathbf{d}}_I^{(k)} = A^{k+1} \bar{\mathbf{d}}_I^{(k+1)} + B^{k+1} \mathbf{d}_H^{(k+1)} \tag{5}$$

The average distortion for the original video frames may be expressed as:

$$\bar{\mathbf{d}}_I^{(0)} = \sum_{k=1}^{T} B^k \prod_{j=1}^{k-1} A^j \mathbf{d}_H^{(k)} + \prod_{j=1}^{k-1} A^j \bar{\mathbf{d}}_I^{(T)} \tag{6}$$

The parameters B^k and A^k are determined by training (curve-fitting) the model based on the measured distortions for two rate points (512 Kbps and 1024 Kbps) for one group of 4 frames each from the different sequences. These models are then used in the cross-layer optimization for all other groups of frames, i.e. the training is performed on 4 frames, while the testing is done on 40 frames from each sequence. Thus, the complexity of the cross-layer strategy is reduced significantly.

5.2. Performance Results

The distortions computed using the models are used in equation (1) to solve the joint optimization problem, and determine the optimal packet sizes and retry limits for each subband. We repeat our lossy scenario experiments and present experimental results over 40 frames for the optimization with the modeled distortions. We compare these results with those obtained for the exhaustive optimization as well as the reference scheme for the same four sequences, in Table 8.

Table 8. Performance Results: Model-based Optimization (1024 Kbps) $p_e = 10^{-4}$**)**

Seq.	Scheme	PSNR (dB)	Overhead
Foreman	Reference	34.90	27%
	Model-based	35.14	25%
	Optimized	35.66	25%
Football	Reference	28.01	35%
	Model-based	29.09	35%
	Optimized	29.14	36%
Mobile	Reference	24.29	29%
	Model-based	24.91	28%
	Optimized	25.10	27%
Coastguard	Reference	28.91	25%
	Model-based	29.43	23%
	Optimized	30.17	23%

The model-based scheme performance is between that of the exhaustive scheme and the reference

fixed-packetization scheme. We can see that the model-based scheme performance is very close to the exhaustive optimization especially for the Football and Mobile sequences, for which the model is very accurate. In our future work, we plan to adopt improved distortion models for the cross-layer optimization.

6. Conclusions

In this paper, we propose a content-aware cross-layer (Application and MAC) packetization and retransmission strategy for optimized multimedia transmission over wireless networks. We show that previously proposed state-of-the-art MAC-only optimization schemes lead to a sub-optimal performance for wireless multimedia. We conclude that both the packetization and retransmission strategies need to be optimized jointly based on content parameters, such as the distortion impact, as well as channel conditions. We formulate this joint optimization problem in terms of minimizing the expected distortion and rate overhead, and solve it numerically to determine the optimal packet sizes and retransmission limits for each spatio-temporal subband. To enable this content and channel aware adaptive packetization, we propose a simple real-time packetization algorithm for the deployed scalable video coder.

Subsequently, we show that the proposed optimized cross-layer aware packetization strategies can improve the PSNR over fixed packetization schemes by 0.4-1.8 dB under different channel loss scenarios. The proposed scheme can be adopted in conjunction with other compression schemes and protection and adaptation schemes at the lower layers of the protocol stack. Our joint optimization improves the multimedia performance under losses, especially for moderate and high transmission bit-rates. However, a disadvantage of the proposed scheme is the complexity incurred in performing this exhaustive optimization. In order to reduce this complexity, we propose to use content-based distortion models to drive the cross-layer strategies.

Summarizing, the main conclusions of this paper are threefold. First, the optimization of packet sizes solely at the MAC results in a sub-optimal performance for wireless video delivery, as it does not explicitly consider the content-based distortion impact at the application layer. Secondly, significant improvements in distortion can be obtained by optimizing the packet sizes in a cross-layer manner that explicitly considers the content characteristics and the resulting R-D performance under different channel conditions. In this way, cross-layer rate-distortion-resilience tradeoffs can be performed to determine the optimal packet sizes. Finally, our results indicate that the additional gains due to R-D optimized retransmission limit adaptation are not significant as compared to schemes that explicitly consider the codec features for packet prioritization and deploy layered priority queuing to drive the retransmission adaptation. Our further research will consider more sophisticated mechanisms for packetization, the use of better network loss models, better R-D models for on the fly cross-layer optimization etc.

7. References

[1] B. Girod and N. Farber, “Wireless Video,” in M.T. Sun and A.R. Reibman (eds.), “Compressed Video Over Networks”, Marcel Dekker, 2001.

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

[3] J. Rogers and P. Cosman, “Wavelet zerotree image compression with packetization,” *IEEE Signal Processing Letters*, vol. 11, 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, March 2001.

[5] S. Wenger, “H.264/AVC over IP,” *IEEE Trans. Circuits and Syst. Video Technology*, vol. 13, no. 7, pp. 645-56, July 2003.
[6] D. Qiao and S. Choi, “Goodput enhancement of IEEE 802.11a Wireless LAN via link adaptation,” Proc. IEEE, ICC 01, Helsinki, June 2001.
[7] M. van der Schaar, S. Krishnamachari, S. Choi, X. Xu, “Adaptive Cross-Layer Protection Strategies for Robust Scalable Video Transmission over 802.11 WLANs,” *IEEE Journal on Selected Areas of Communications*, 2003.
[8] T. Rusert, K. Hanke and J. Ohm, “Transition filtering and optimization quantization in interframe wavelet video coding,” VCIP, Proc. SPIE, vol. 5150, pp. 682-93, 2003.
[9] D. Singer, W. Belknap, G. Franceschini, “ISO Media File Format Specification - MP4 Technology under consideration for ISO/IEC 14496-1:2002 Amd 3,” Committee Draft, ISO/IEC JTC1/SC29/WG11 MPEG01/N4270-1, July 2001.
[10] S. McCanne, V. Jacobson, M. Vetterli, "Receiver-driven Layered Multicast," Proceedings of ACM SIGCOMM'96, pp. 117 – 130, 1996.
[11] A. Majumdar, D. Grobe Sachs, I. V. Kozintsev, K. Ramchandran "Multicast and Unicast Real-Time Video Streaming Over Wireless LANs", *IEEE Trans. Circuits and Syst. Video Technology*, vol.12, no.6, pp. 524 – 534, June 2002.
[12] M. Wang and M. Van der Schaar, “Operational Rate-Distortion modeling for wavelet video coders,” submitted to *IEEE Trans. Signal Processing*.
[13] D. Taubman, M. Reji, D. Maestroni, S. Tubaro, “SVC Core Experiment 1 – Description of UNSW Contribution”, MPEG document m11441, October 2004.
[14] D. S. Taubman and M. W. Marcellin, *JPEG 2000-Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, 2002.
[15] Q. Li, M. van der Schaar, “Providing Adaptive QoS to Layered Video over Wireless Local Area Networks through Real-Time Retry Limit Adaptation”, *IEEE Trans. on Multimedia*, vol. 6, no. 2, April 2004.
[16] Q. Li and M. van der Schaar, “A Flexible Streaming Architecture for Efficient Scalable Coded Video Transmission over IP Networks”, *ISO/IEC JTC 1/SC 29/WG 11/M8944*, Oct. 2002.
[17] G. Feideropoulou and B. Pesquet-Popescu, “Stochastic modeling of the spatio-temporal wavelet coefficients - Application to quality enhancement and error concealment,” accepted in EURASIP J. Signal Processing, Appl., 2004.