Classification-Based System For Cross-Layer Optimized Wireless Video Transmission

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

***Abstract*—Joint optimization strategies across various layers of the protocol stack have recently been proposed for improving the performance of real-time video transmission over wireless networks. In this paper, we propose a new, low complexity system for determining the optimal cross-layer strategies for wireless multimedia transmission based on classification and machine learning techniques. We first determine offline the optimal cross-layer strategy for various video sequences and channel conditions (training data). Subsequently, we extract relevant and easy to compute content features, encoder-specific parameters, and channel resources from the training data, and train a statistical classifier based on these optimal results. At run-time, we predict using the classifier the optimal cross-layer compression and transmission strategy using these simple, on-the-fly computed features. Hence, we consider the complex problem of finding the optimal cross-layer strategy during the training phase only, and rely at transmission-time on low-complexity classification techniques. We illustrate the proposed classification-based system by performing MAC-application layer optimizations for video transmission over 802.11a wireless LANs. Specifically, we predict the optimal MAC retry limits for the various video packets and compare our results against both optimal and conventionally used *ad-hoc* cross-layer solutions. Our results indicate that considerable improvements can be obtained through the proposed cross-layer techniques relying on classification as opposed to optimized *ad-hoc* solutions. The improvements are especially important at high packet-loss rates (5% and higher), where deploying a judicious mixture of strategies at the various layers becomes essential. Furthermore, our proposed classification-based system can be easily modified to include other layers from the OSI stack during the cross-layer optimization.**

***Index Terms*—Classification for delay-constrained video transmission, cross-layer optimization.**

I. Introduction

DUE to their flexible and low cost infrastructure, wireless LANs (WLANs) [1] are poised to enable a variety of delay-sensitive multimedia transmission applications, such as videoconferencing, emergency services, surveillance, telemedicine, remote teaching and training, augmented reality, and distributed gaming. However, existing wireless networks provide dynamically varying resources with only limited support for the Quality-of-Service (QoS) required by the *delay-sensitive, bandwidth-intense and loss-tolerant multimedia applications* [2]. Cross-layer optimization strategies [3]–[10] have been proposed as a solution for improving the performance of multimedia streaming applications over WLANs. These solutions include joint PHY-MAC-Application layer optimizations such as robust wireless video transmission over 802.11 a/b/g using adaptive retransmission [3], [8], [9], adaptive modulation for various priority layers [5], adaptive packetization [11], MAC-PHY link layer adaptation for improved goodput [12], power optimized transmission etc. In general, the cross-layer optimization problem is nontrivial to solve because

- analytically deriving the relation between quality, delay and rate is difficult and sometimes only nondeterministic (worst or average case) relationships can be established;
- strategies at different OSI layers depend on strategies deployed at the same or other layers;
- channel conditions as well as multimedia traffic and sequence characteristics vary dynamically.

Consequently, previously proposed solutions for cross-layer optimization, including those in references [3]–[11] solve the problem using either *ad-hoc* heuristic approaches or rely on complex optimizations that often cannot be performed in real-time. In this paper, we explore classification-based algorithms as they exhibit low run-time complexity and have been shown to perform well for complex optimizations in image and video analysis applications.

A. Cross-Layer Problem Formulation and Challenges

We formulate the cross-layer design problem as an optimization that has as objective the joint selection of strategies across multiple OSI layers.[1] Assuming that N_{PHY}, N_{MAC}, and N_{APP} denote the number of adaptation and protection strategies available at the physical PHY, MAC, and application APP layers and that these strategies are denoted as PHY_i, MAC_i, APP_i etc., we define the joint cross-layer strategy S as $S = \{PHY_1, \ldots, PHY_{N_{PHY}}, MAC_1, \ldots, MAC_{N_{MAC}}, APP_1, \ldots, APP_{N_{APP}}\}$. The cross-layer optimization problem attempts to find the optimal composite strategy

$$S^{opt}(\mathbf{c}, \mathbf{v}) = \arg\max_{S} Q\left(S(\mathbf{c}, \mathbf{v})\right) \quad (1)$$

Manuscript received September 8, 2005; revised January 5, 2006. This work was supported by the National Science Foundation under Career CCF-0541867 and by grants from the UC Micro Program, IBM, Intel IT Research, and Samsung.

M. van der Schaar is with the Electrical Engineering Department, University of California Los Angeles (UCLA), Los Angeles, CA 90095-1594 USA (e-mail: 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).

R. Wong is with the Electrical and Computer Engineering Department, University of California, Davis, CA 95616 USA (e-mail: rswong@gmail.com).

Digital Object Identifier 10.1109/TMM.2006.879827

[1]In this optimization, the Transport and Network layers are not considered. UDP is the most used transport protocol for real-time video streaming applications. Since UDP does not provide error protection and rate adaptation like TCP, the impact of the transport layer on the cross-layer strategy is minimal (only the packetization overheads are affected). Similarly, since we focus on video transmission for a single hop, the impact of the network layer on the cross-layer optimization is also minimal.

1520-9210/$20.00 © 2006 IEEE

which maximizes the multimedia quality Q (PSNR/perceived) subject to rate (R) and delay constraints:

$$R(S(\mathbf{c},\mathbf{v})) \le R_{\max}$$
$$Delay(S(\mathbf{c},\mathbf{v})) \le Delay_{\max}. \quad (2)$$

In the above equations, $\mathbf{c}$ represents the underlying channel conditions, and $\mathbf{v}$ is derived based on video codec and content characteristics (see our discussion in Section II-B, where we group $\mathbf{c}$ and $\mathbf{v}$ into feature vector $\mathbf{f}$). $R_{\max}$ is the maximum transmission bit-rate and $Delay_{\max}$ is the maximum tolerable delay at the application layer. Note that various compression and cross-layer strategies lead to different delays, which impact the multimedia quality Q for low-delay applications such as video-conferencing or surveillance, where $Delay_{\max}$ can be as low as 150 ms.

To determine the optimal strategy S^{opt}, we can exhaustively examine all the possible strategies for each layer, evaluate the utility (e.g., quality under the given constraints) corresponding to each strategy, and select the one with the highest utility. However, such a solution is impractical in real-time due to its complexity. Depending on the granularity at which the adaptation is performed—per video flow, layer, frame, slice or video coding unit (e.g., video packet)—and the time-varying wireless channel conditions and multimedia content characteristics, the complexity of these strategies is further increased. In this paper, we focus on jointly determining the optimal strategy for a set of video packets contained within one Group of Pictures (GOP) of video (see Section 2.1.2 for more details).

We propose to use classification techniques to solve this complex real-time cross-layer optimization problem effectively. Our system is based on the observation that we can identify features (that can be easily determined by the video encoder and wireless card) that are good indicators of the optimal joint strategy. As an illustration of the proposed classification-based approach, we consider the problem of selecting optimal MAC retransmission limits for each application-layer video packet. Nevertheless, the proposed framework can easily be extended to include other layers and/or cross-layer strategies.

B. Related Research

The MAC-application layer optimization has been solved using both *ad-hoc* heuristics [4], [8] and optimal non real-time approaches [11]. Alternatively, we propose a classification-based system that has a performance comparable to that of the optimal approaches, while having a run-time complexity comparable to that of low-cost *ad-hoc* solutions.

A plethora of video applications that support users in browsing and retrieving digitized video, use automatic classification [26]. These classification approaches include neural networks or decision-tree based algorithms that use features such as color histograms, motion and texture features etc. Content-based classification techniques have also been extended for selection among the various MPEG encoding parameters and different scalability options [16]. In [13], [14], a QoS framework has been proposed that maps categorized video packets onto the relative differentiated service (DiffServ) provided by the wireless channel using a predetermined pricing model. However, while classification-based techniques have been often used in practice for video retrieval applications or to drive the encoding and transmission parameters at the application layer, a joint classification methodology that considers video characteristics, codec-dependent priority classes and wireless channel resources for cross-layer optimization has not yet been designed.

C. Contributions and Outline of This Paper

The main contribution of this paper is the classification-based system to determine the optimal cross-layer strategy for wireless video transmission. We formulate the problem of maximizing the utility (video quality under delay and rate constraints) for cross-layer optimization as a standard classification problem. We then examine two different classification strategies: based on minimizing the probability of misclassification, and minimizing the cost of misclassification (in terms of distortion). We compare the proposed system for joint MAC-application layer optimization against existing *ad-hoc* and optimal (exhaustive) cross-layer strategies for different video sequences, encoding parameters and channel conditions.

The paper is organized as follows. We describe our proposed classification based cross-layer optimization in Section II. In Section III, we validate the proposed cross-layer system under different scenarios. Our conclusions are summarized in Section IV.

II. Proposed Classification-Based Cross-Layer Optimization

We first focus on determining the optimal MAC retry limit for each video packet given the maximum available bit-rate ($R_{\max}$), the maximum tolerable delay $Delay_{\max}$ and the experienced bit error rate (P_e). The proposed classification-based wireless video transmission system is depicted in Fig. 1. It consists of an offline training module followed by online processing. The former includes modules for class definition and classifier learning, and the latter mainly involves classification and real-time cross-layer strategy prediction for video packets. The major steps in the approach are as follows.

Step 1) *Generate ground truth (offline)*. We first collect a set of packets from a variety of video sequences under different representative channel conditions and identify the entire set of cross-layer strategies available at the wireless station. For each packet in this training set and collection of encoding parameters, we extract the compressed-domain content features (CF) and packet types (PT) determined based on the specific encoder configuration/parameter set. Our feature sets also include the wireless channel conditions (WCC)—$R_{\max}$[2] and P_e. Subsequently, the optimal strategy S^{opt} resulting in the best quality for the different training sequences, packet types and channel conditions is determined using dynamic programming (see Section II–A2).

Step 2) *Train Classifier (offline)*. The key is to determine, for each packet j, a mapping from the composite feature vector $\mathbf{f}(j)$ to class label l_j, corresponding to a specific optimal strategy

[2] $R_{\max}$ can be determined based on both the video encoding rate, delay constraint and the PHY rate used for transmission (determined based on the modulation strategy, etc.) [9], [10].

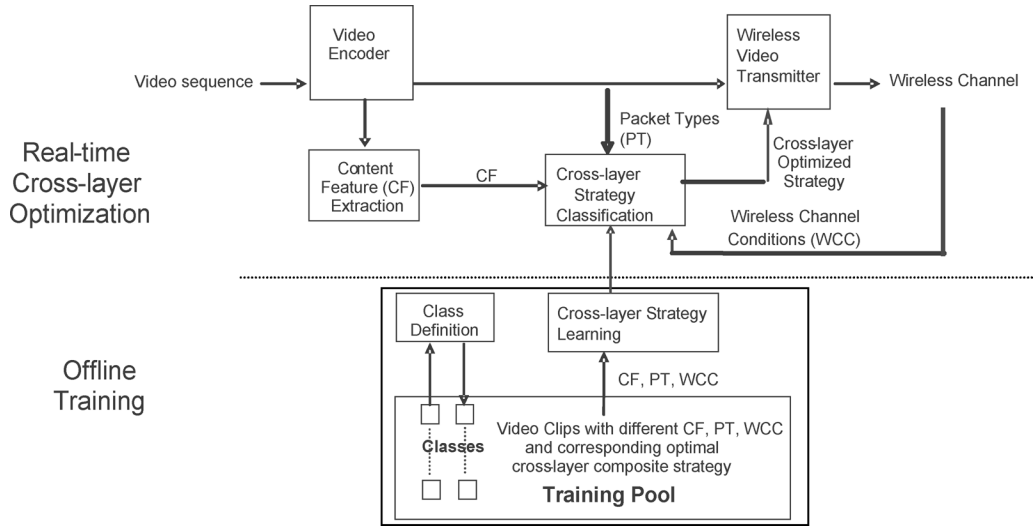

Fig. 1. Classification-based cross-layer system for wireless video.

S^{opt}. During training, supervised clustering methods (see Section II-C for details) are used to map the composite features to the corresponding class label. We examine two different classification strategies aimed at minimizing the probability of misclassification, and minimizing the cost of misclassification (in terms of video distortion), respectively.

Step 3) *Real-time strategy selection based on classification*. The optimal strategy S^{opt} for a sequence of incoming video packets given the instantaneous wireless channel conditions/characteristics can be then determined, by the trained classifier, on a packet-by-packet basis using the composite feature vectors. The selected strategy is used to determine the optimized parameters and configurations of the wireless multimedia system.

The various steps outlined above are described in more detail in the subsequent sections.

A. Ground Truth Generation

Although the concepts proposed in this paper can potentially be deployed with state-of-the-art nonscalable coding solutions [23], [24], this usually entails smaller granularity for real-time packet prioritization and adaptive retransmissions. Scalable video coders provide graceful degradation and adaptability to a large range of wireless channel conditions and power constraints. Hence, we deploy a recently-proposed $\mathrm{t}+2\mathrm{D}$ scalable video codec based on Motion Compensated Temporal Filtering (MCTF) [15], [16] and block-based content adaptive arithmetic coding similar to that of JPEG-2000 [26].

In typical MCTF-based video compression, rate allocation for the scalable bitstream is performed independently per GOP. During ground truth generation, we fix the encoding parameters: we set both spatial and temporal levels to 4 for GOPs of 16 frames. Each spatio-temporal subband (or group of subbands)[3] is partitioned into independently decodable units, which are encoded in a rate scalable manner. Hence, the bitstream consists of multiple quality layers (corresponding to different decoding bit-rates) for each decoding unit. The codec performs an R–D optimization to determine the bit allocation for each layer of a decoding unit, based on its distortion impact. After the R–D optimization, $R^q_{s,b}$ bits are assigned to decoding unit b of subband s at layer q. For instance, if during the encoding we specify four possible transmission bit-rates, the codec creates four layers for each decoding unit. When the video is transmitted at the lowest bit-rate, only the lowest layer from each decoding unit is transmitted, while at the highest bit-rate, all layers of each decoding unit are sent. We packetize the bitstream separately for each transmission bit-rate by grouping together all bits required to transmit a decoding unit at this target bit-rate (i.e., we aggregate the layers corresponding to the target bit-rate). Subsequently, multiple decoding units are packetized together into packets of maximum size 500 bytes.[4]

We label the size of packet j as $L(j)$ and the corresponding distortions based on whether a packet was received or lost as $D^{Quant}(j)$ and $D^{loss}(j)$, respectively. To determine the distortion due to packet j at the target bit-rate (i.e., $D^{Quant}(j)$), we can decode other video packets within a GOP losslessly except for decoding units within packet j, which we decode at the target bit-rate.[5] Consequently, $D^{Quant}(j)$ represents the total distortion impact of packet j across all the decoded frames within a GOP (i.e., it includes the distortion propagation across the temporal decomposition tree). Similarly, to compute $D^{loss}(j)$, we can decode all other video packets within a GOP losslessly, except for packet j, which we discard. The resulting distortion, which is independent of the transmission bit-rate, corresponds to $D^{loss}(j)$.

In a real transmission scenario, the expected total distortion $\overline{D}(m,j)$ due to packet j, given a retry limit m, depends on both the $D^{Quant}(j)$ and $D^{loss}(j)$ values of that packet:

$$\overline{D}(m,j) = P_{succ}(m,j)D^{Quant}(j) + P_{fail}(m,j)D^{loss}(j) \tag{3}$$

[3]In this paper, we constrain each decoding unit to contain data from precisely one frame and one spatial resolution.

[4]To preserve the spatio–temporal scalability, we do not packetize decoding units from different subbands in the same packet. Note, however, that other packetization strategies could also be deployed exhibiting other merits for the cross-layer optimization problem, but they are beyond the scope of this paper.

[5]The deployed wavelet codec determines the distortion impact of each quality layer for each decoding unit in order to solve the rate allocation problem during encoding. The packet loss and quantization distortions may be computed from these distortions in real time.

TABLE I
PERCENTAGE OF PACKETS ASSIGNED A SPECIFIC RETRY LIMIT S^{opt} FOR THE MOBILE SEQUENCE

	$R_{\max}$=512kps				$R_{\max}$=1024kps			
m	P_L=1%	P_L=3%	P_L=5%	P_L=10%	P_L=1%	P_L=3%	P_L=5%	P_L=10%
5	8.97%	13.21%	17.88%	21.81%	4.98%	8.12%	12.87%	16.12%
1	50.5%	51.51%	62.34%	62.37%	31.8%	43.05%	45.92%	47.47%
0	35.32%	29.12%	12.67%	7.32%	55.1%	38.61%	30.1%	23.12%
Discard	5.21%	6.16%	7.11%	8.5%	8.12%	10.22%	11.11%	13.29%

where $P_{succ}(m,j)$ is the probability of successfully receiving the packet and $P_{fail}(m,j)$ is the probability of losing the packet ($P_{succ}(m,j) + P_{fail}(m,j) = 1$). Hence, quality-resilience tradeoffs need to be considered when determining the distortion-optimal cross-layer strategy. In this paper, we determine the optimal retry limit $T^{opt}(j)$ for packet j based on minimizing the total expected distortion.

In delay-sensitive wireless video applications, there are two reasons for packet loss: link packet erasure—P_{error}, and missed transmission deadlines (i.e., due to incurred delay)—P_{delay}. In our analysis we make the following assumption: packets that miss their transmission deadline are discarded at the application layer. Hence, the overall packet loss rate may be written as

$$P_{fail}(m,j) = P_{error}(m,j)\,(1 - P_{delay}(m,j)) + P_{delay}(m,j). \tag{4}$$

If we assume that the wireless link is a memoryless packet erasure channel [2], [4], such that the packets are dropped independently,[6] we can compute the link packet erasure rate for packet j with retry limit m as

$$P_{error}(m,j) = 1 - \sum_{n=1}^{m+1} P_{L(j)}^{n-1}\left(1 - P_{L(j)}\right) = P_{L(j)}^{m+1}. \tag{5}$$

where $P_{L(j)} = 1-(1-P_e)^{L(j)}$, $L(j)$ is the size in bits of packet j and P_e is the bit error probability controlled by the physical layer, based on the channel SNR, channel coding and modulation strategy used, etc. Note that while the PHY parameters are not explicitly considered in our classification-based cross-layer optimization, we assume that link adaptation mechanisms such as those discussed in [10], [12] and currently implemented in the wireless cards are deployed in our system. Hence, the P_e and PHY layer rate $Rate_{PHY}$ were determined in our optimization assuming that link adaptation was performed per packet. Once a packet is added to the MAC layer buffer, the mean number of transmissions for it equals:

$$\overline{m}_j = \sum_{t=1}^{m} tP_{L(j)}^{t-1}\left(1 - P_{L(j)}\right) + (m+1)P_{L(j)}^{m}. \tag{6}$$

Given the PHY rate $Rate_{PHY}(j)$, the average time to transmit a packet j is given by

$$Time_{Avg}(m_j,j) = \hat{m}_j\left(\frac{L(j)}{Rate_{PHY}(j)} + Time_O\right) \tag{7}$$

where

$$\hat{m}_j = \begin{cases} \overline{m}_j; & \text{if packet } j \text{ is added to the MAC buffer} \\ 0; & \text{if packet } j \text{ is discarded at APP layer} \end{cases} \tag{8}$$

and $Time_O$ is the timing overhead spent, which can be approximated based on [31] and [32] and includes the time of waiting for acknowledgements, duration of empty slots, expected backoff delays, etc.

Since packets are transmitted sequentially, in order to determine whether a packet violates its delay constraint, we need to consider the expected time taken by all packets transmitted before it. Since each packet takes (on average) time $Time_{Avg}(m_j,j)$ to be transmitted, the delay constraint for packet k is not violated if

$$\sum_{j=1}^{k} Time_{Avg}(m_j,j) \le Deadline(k) \tag{9}$$

where $Deadline(k)$ is the time deadline for the kth packet in order for it to be decoded and displayed.[7] The $Deadline(k)$ is determined based on the coding dependencies between frames (and thus, the encoding structure and parameters) and it also includes the maximum tolerable delay $Delay_{\max}$. If the condition in (9) is not violated, $P_{delay}(m_j,j)$ is 0, otherwise it is 1.

The cross-layer optimization, aimed at determining jointly the optimal retry limits for each packet j—$T^{opt}(j)$—is solved offline, using dynamic programming. Dynamic programming is used to efficiently examine all possible combinations of retransmission limits (selected from a set of possible values) and packet discards at the application layer, and to determine the selection that minimizes the overall expected distortion. Note that the complexity associated with exhaustively determining the optimal cross-layer strategy is high. In the proposed classification based system, these optimizations are performed only offline and the resulting decisions are used as ground truth for our real-time classification system.

Tables I and II show the obtained optimal retry limits assigned to packets under different channel conditions. While the actual packet loss probability varies per packet (based on its length), we define an equivalent P_L to represent the underlying channel conditions, where P_L is computed from P_e assuming a packet size of 500 bytes. In the experiment below, for illustration simplicity, only a retry limit of 0, 1, or 5 could be selected for each packet. The maximum value (in this case 5) can be determined based on the maximum tolerable delay $Delay_{\max}$

[6]In [36] it was shown that the Gilbert–Elliot model for bit-errors in wireless channels generates packet loss patterns that are statistically similar (within $\sim$0.4%) to naïve independent bit-by-bit simulations. Using this argument we translate the independent loss model to an independent packet loss probability and generate results using one Bernoulli experiment per packet. This also reduces the computational complexity of our experiments significantly (a factor of 100).

[7]During the offline dynamic programming based optimization (for classifier training), expected transmission times were used. At run-time, a packet is discarded only if its actual (not expected) deadline has passed.

TABLE II
PERCENTAGE OF PACKETS ASSIGNED A SPECIFIC RETRY LIMIT S^{opt} FOR THE *FOREMAN* SEQUENCE

	$R_{\max}$=512kps				$R_{\max}$=1024kps			
m	P_L=1%	P_L=3%	P_L=5%	P_L=10%	P_L=1%	P_L=3%	P_L=5%	P_L=10%
5	10.34%	12.43%	15.78%	20.12%	6.44%	9.47%	12.88%	17.42%
1	28.8%	42.99%	46.98%	48.89%	13.86%	27.5%	33.56%	56.97%
0	55.71%	37.37%	29.13%	20.18%	71.74%	51.67%	42.19%	13.11%
Discard	5.15%	7.21%	8.12%	10.81%	7.95%	11.36%	11.36%	12.5%

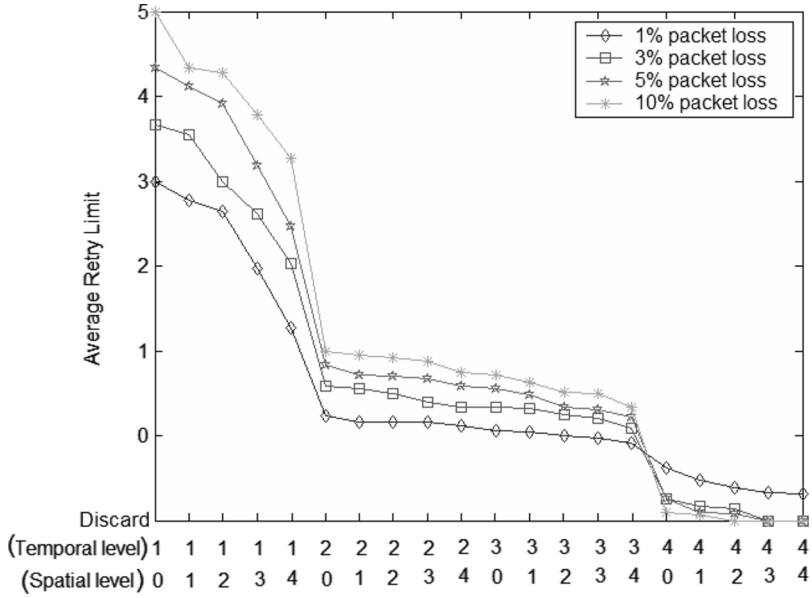

Fig. 2. Average retry limit for different packet types (*Foreman* sequence at 1024 kps).

[19]. The results show that S^{opt} depends on the sequence characteristics (*Mobile* and *Foreman* sequences have different results), and channel-dependent parameters P_L and $R_{\max}$. Consequently, these features are part of our classification system.

B. Feature Selection

For video packet j, we need to identify a feature vector $\mathbf{f}(j)$ that can predict the optimal decision with low complexity (i.e., with features that can be easily computed/extracted at run-time). The wireless channel condition (WCC) features P_e (equivalently P_L) and $R_{\max}$ can be determined in real-time based on information that can be easily extracted from the wireless card driver. For example, the transmitter can use the received signal strength indicator (RSSI) of previously received MAC acknowledgement frames, as well as MAC acknowledgement reports to determine these features [10], [33]. Among content features (CF) we select the packet energy, which may be used to distinguish among sequences with different levels of spatio-temporal detail. The energy for packet j is calculated by summing up the squared wavelet coefficients $coeff$ belonging to the packet

$$E_j = \sum_{i \in W_j} (coeff(i))^2 \tag{10}$$

where W_j is the set of decoded coefficients (collected from all the decoding units within the packet) belonging to packet j. At encoding time, the energy of each decoding unit is computed for the various quality layers. During transmission, we compute the packet energy simply by aggregating the relevant energies corresponding to the target bit-rate.

The codec-specific (PT) features include the spatial and temporal level of the data in packet. This is because packets belonging to distinct spatio-temporal bands have a different impact on the overall distortion and require different protection (retry limits). Fig. 2 depicts the average retry limit for various packets (computed using dynamic programming) belonging to different spatio–temporal levels.

As expected, the retry limit of the packet decreases with increasing spatio–temporal level. Most *ad-hoc* cross-layer strategies are based on this simple classification criterion for selecting the retry limit, i.e., the spatio–temporal level (or frame-type for conventional video coders). However, these schemes do not use either the content characteristics, or the channel conditions, which directly impact the optimal retry limit. In order to test the suitability of the CF and PT features,[8] we compute the correlation coefficient between them and the optimal decision sequence (i.e., the choice of optimal retry limit per packet). If feature i from packet j is called $\mathbf{f}_i(j)$ and the optimal retry limit (decision) for this packet is $T^{opt}(j)$, then the correlation coefficient between the feature sequence and the decision sequence may be defined as

$$\rho_i = \frac{\sum_{j=1}^{N_p} \mathbf{f}_i(j) T^{opt}(j)}{\sqrt{\sum_{j=1}^{N_p} (\mathbf{f}_i(j))^2 \sum_{j=1}^{N_p} (T^{opt}(j))^2}} \tag{11}$$

where N_P is the number of packets in the GOP.

[8]We exclude the WCC features as they are common to all the packets.

TABLE III
MOBILE: CORRELATION COEFFICIENTS ρ_i

Features	$R_{\max}$= 512kps				$R_{\max}$= 1024kps			
	P_L=1%	P_L=3%	P_L=5%	P_L=10%	P_L=1%	P_L=3%	P_L=5%	P_L=10%
Packet energy	0.79	0.82	0.82	0.83	0.75	0.80	0.81	0.81
Temporal level	0.76	0.86	0.90	0.93	0.72	0.82	0.85	0.89
Spatial Level	0.62	0.73	0.75	0.80	0.60	0.67	0.73	0.76

TABLE IV
PAIR-WISE MI FOR THE CHOSEN FEATURE SET

	Packet energy	Temporal level	Spatial level	$R_{\max}$	P_L
Packet energy	3.4	1.37	1.34	1.3	1.19
Temporal level	1.37	1.90	0.07	0.03	0
Spatial level	1.34	0.07	2.12	0.01	0
$R_{\max}$	1.30	0.03	0.01	1.57	0
P_L	1.19	0	0	0	2

TABLE V
ACCURACY OF THE CLASSIFIER BASED ON SINGLE FEATURE

	Packet energy	Temporal level	Spatial level	P_L	$R_{\max}$
Percentage of accuracy	52%	58%	48%	52%	48%

Table III shows the correlation coefficients for these different features with the optimal retry limit, for the *Mobile* sequence. Similar results were obtained for other video sequences. The large values (close to 1) of the coefficients ρ_i in Table III show that for given channel conditions, the selected features are well correlated with the optimal decision sequence.

In order to examine the redundancy in our feature set, we compute the mutual information[9] (MI) between pairs of features and present these results in Table IV. We can see from Table IV that while there is some redundancy among the features, especially between packet energy and the rest of the features, each feature contains nonredundant information (for a majority of cases, the MI is significantly lower than the feature entropy). Allied with this is the fact that none of these features are computationally complex to determine, and hence we use the complete set of features in our system.

Finally, in order to validate these features for the actual classification task, we also examine the classifier performance with each of these individual features. Table V shows the classifier accuracy results with each individual feature.

The temporal level feature leads to the best classification performance, while the video rate and the spatial level have the worst classification performance. We use this knowledge to design an *ad-hoc* strategy (similar to that used in [4], [8] but for different video coders) to determine the packet retransmission limits. We will discuss this in more detail in the Section III. Finally, as we use these features jointly, the classifier accuracy increases to $\sim$83%. While additional features can be used (e.g., the motion vectors, available bits per frame, number of bit-planes per frame at various bit-rates, etc.), these will increase the complexity of the real-time system with only limited possible improvement in the classification performance (see the results section).

[9]The MI between two random variables X and Y with distributions $p(x)$ and $p(y)$ and joint distribution $p(x,y)$ is defined as $MI(X,Y) = \sum_x \sum_y p(x,y) \log(p(x,y)/p(x)p(y))$.

Summarizing, the feature extraction step needs to be kept at a low complexity, because it is also performed online. Consequently, we select content and encoder specific features that are already computed during the encoding process (i.e., no additional complexity is needed for feature extraction). These features can be prestored in metadata files together with the video bitstreams. Hence, at transmission time, only the channel features need to be determined based on the RSSI and MAC acknowledgement frames. These values can readily be accessed from device drivers of existing wireless cards (see, e.g., Intel PRO/Wireless 2915 ABG Network Connection and Intel PRO/Wireless 2200 BG Network Connection mini PCI adapters).

C. Classifier Design

The cross-layer optimization problem involves assigning a retry limit to each packet such that the expected overall decoded distortion is jointly minimized. Let us assume that we have M available retry limits $\{X_1, \ldots X_M\}$ (i.e., an M-class classification problem), and N_t packets in our training set. From the data within each packet j, we extract an F-dimensional feature vector $\mathbf{f}(j) \in \mathbb{R}^F$ (in our case $F = 5$). We provide the classifier with feature vectors $\mathbf{f}(j)$, $1 \leq j \leq N_t$, and the associated optimal retry limit $T^{opt}(j) \in \{X_1, \ldots X_M\}$ for each packet. The classifier then partitions the feature space $\mathbb{R}^F$ into M nonoverlapping regions $G_1, \ldots G_M$, with region G_i associated with a unique optimal retry limit X_i, such that the error in classification (i.e., probability of misclassification) on the training data is minimized. This may be written as

$$\{G_1, \ldots, G_M\}^{opt} = \arg\min_{G_1, \ldots G_M} \sum_{j=1}^{N_t} \left[1 - B\left(\mathbf{f}(j) \in G_i | T^{opt}(j) = X_i\right)\right] \quad (12)$$

where $B(\mathbf{f}(j) \in G_i | T^{opt}(j) = X_i)$ is a binary-valued function that takes value 1 when vector $\mathbf{f}(j)$ is correctly classified, i.e., if

$\mathbf{f}(j)$, with optimal retry $T^{opt}(j)(=X_i)$ is inside region G_i, and zero otherwise. In the remainder of this paper, we refer to this classification method as classification algorithm 1 ("*classif_1*").

While minimizing the previously defined classification error, the classifier views all feature vectors in the training set equivalently. However, in reality, the feature vectors do have different importance because misclassifying different feature vectors can lead to different penalties in the total distortion. Clearly, some packets have a higher impact on distortion, e.g., packets belonging to the temporal low-pass subbands are likely to impact the total decoded distortion more than packets belonging to the temporal high-pass subbands. Since we want to finally minimize the decoded distortion, we need to modify the classifier to take the distortion impact into account. Let the importance of packet j with feature vector $\mathbf{f}(j)$ be determined by the cost of misclassifying it $C(j)(\geq 0)$. We will discuss this cost in more detail later. Then, the classifier design problem may be written as

$$\{G_1,\ldots,G_M\}^{opt} = \arg\min_{G_1,\ldots G_M} \sum_{j=1}^{N_t} C(j)\left[1-B\left(\mathbf{f}(j)\in G_i|T^{opt}(j)=X_i\right)\right] \quad (13)$$

or, alternatively, as

$$\{G_1,\ldots,G_M\}^{opt} = \arg\min_{G_1,\ldots G_M} \sum_{j=1}^{N_t}\sum_{k=1}^{C(j)} \left[1-B\left(\mathbf{f}(j)\in G_i|T^{opt}(j)=X_i\right)\right]. \quad (14)$$

Equation (14) has the same form as the optimization problem in (12) where instead of providing the classifier with vector $\mathbf{f}(j)$, we provide it vector $\mathbf{f}(j)$ repeated $C(j)$ times.[10] Hence, by modifying the training set in such a manner, we can use the minimized-classification-error classifier to minimize the cost of misclassification. In the remainder of this paper, we label this minimization of cost (distortion-based) classification method as classification algorithm 2 ("*classif_2*").

The cost of misclassification $C(j)$ in our cross-layer problem needs to be defined in terms of the increase in distortion when packet j is assigned the wrong retry limit. From Section II-A2, we know that if packet j is assigned a retry limit X_i, then the expected distortion due to this packet is

$$\overline{D}(X_i,j) = P_{succ}(X_i,j)D^{Quant}(j) + P_{fail}(X_i,j)D^{loss}(j). \quad (15)$$

Furthermore, this retry limit X_i affects the retry limit of the subsequent packets within the GOP and, hence, the expected resulting cumulative distortion (assuming that all the subsequent packets are assigned their optimal retry limits, given that this packet was assigned a retry limit X_i) may be computed as

$$\overline{D}_{cum}(X_i,j) = \overline{D}(X_i,j) + \sum_{k=j+1}^{N_p} \overline{D}\left(T^{opt}(k)|T(j)=X_i,k\right) \quad (16)$$

[10]In general it is not necessary that all the costs $C(j)$ are integers, however without loss of generality we can scale them appropriately to make them integers.

where N_p is the number of packets in the GOP and $\overline{D}(T^{opt}(k)|T(j) = X_i,k)$ is the expected distortion due the packet $k(k > j)$ in the GOP incurred due to selecting the retry limit X_i for packet j. Hence, for each available retry limit X_i that we assign to packet j, we can compute the expected cumulative distortion resulting from this choice. The optimal strategy selects $T^{opt}(j)$ by minimizing such a cumulative distortion. Hence,

$$T^{opt}(j) = \arg\min_{X_i}\left(\overline{D}_{cum}(X_i,j)\right). \quad (17)$$

When instead of this optimal retry limit, we assign the packet a different retry limit X_k the corresponding increase in distortion incurred is $(\overline{D}_{cum}(X_k,j) - \overline{D}_{cum}(T^{opt}(j),j)) \geq 0$. Hence, the total cost $C(j)$ of misclassifying the packet, in terms of distortion may be computed as

$$C(j) = \sum_{\substack{k=1\\ X_k\neq T^{opt}(j)}}^{M} \left(\overline{D}_{cum}(X_k,j) - \overline{D}_{cum}\left(T^{opt}(j),j\right)\right). \quad (18)$$

We use the above cost while training the classifier in the scheme "*classif_2*".

We adopted a supervised nonparametric classification technique using support vector machines (SVM) [20]–[22], [28]–[30] for our classifier. SVM is a method for partitioning the feature space using an optimal hyperplane and has several advantages for our classification problem. Firstly, SVMs minimize structural risk, i.e., the probability of misclassifying a previously unseen data point drawn randomly from a fixed but unknown probability function. Secondly, it is known that SVMs outperform most other classifiers when the training data set is limited, and hence we can train the classifier with a small amount of training data and still expect reasonable performance. Finally, an SVM can deal with nonlinearly separable clusters in the F dimensional feature space, as it provides a nonlinear function approximation by mapping the input vectors into higher dimensional spaces and partitioning these using special hyperplanes. Note though that an issue of concern is that the optimization technique associated with SVM training requires a considerable amount of computation. However, since training is performed offline, prior to the real-time transmission stage, this does not affect the real-time performance of the proposed system.

D. Real-Time Classification

After the classifier has been trained, we deploy it for the real-time determination of the retransmission limit. This decision is made packet by packet, based on its computed feature vector. Additionally, we can refine the obtained retransmission limits (using the classifier) such that packets are not transmitted beyond their deadline, since the wireless card driver can provide timely access to the actual time taken for the packet transmission. In particular, if the retransmission limit for packet j determined by the classifier is $T^{class}(j)$, the current time (measured based on the actual packet transmission so far) is $Time_{Act}$, and

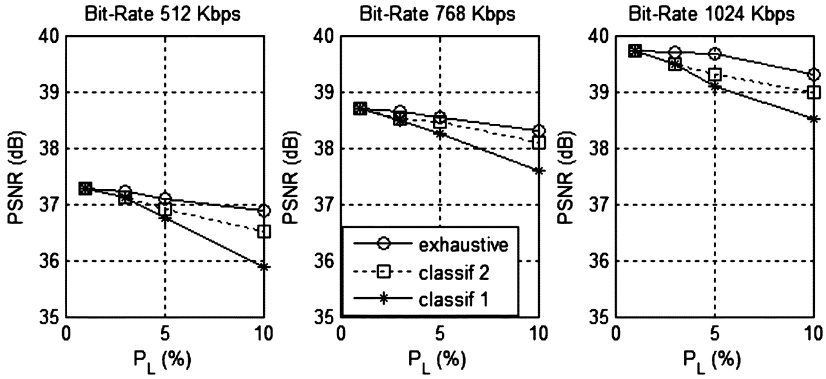

Fig. 3. Results of optimal versus classification-based cross-layer optimizations: *Foreman*.

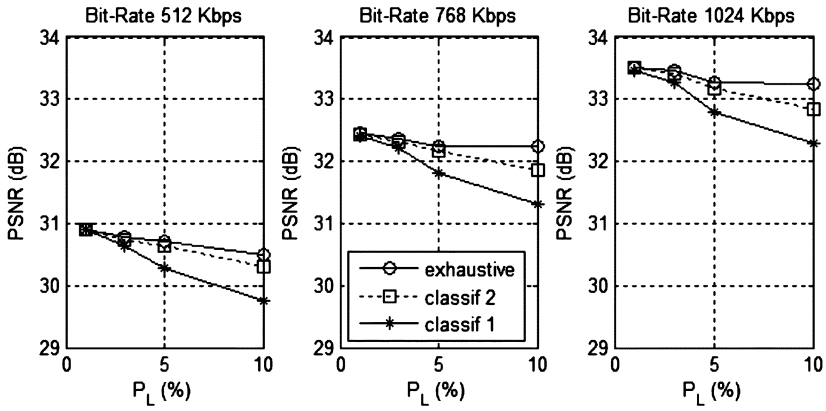

Fig. 4. Results of optimal versus classification-based cross-layer optimizations: *Mobile*.

the packet deadline is $Deadline(j)$, we may tune the retransmission limit as

$$T^{online}(j)=\min\left(T^{class}(j),\left\lfloor\frac{Deadline(j)-Time_{Act}}{\frac{L(j)}{Rate_{PHY}}+Time_O}\right\rfloor\right) \tag{19}$$

where $\lfloor\circ\rfloor$ is the floor operation. In (19), we compute the number of maximum transmission possible for the packet (based on the actual elapsed time, and the packet deadline), and use the $min(\,)$ operation to ensure that packets are not transmitted beyond their deadline.

III. Simulation Results

A. Training Data and Classifier Performance

Our training data set includes five GOPs each from the *Foreman* and *Mobile* sequences at CIF (352×288) resolution and 30 frames/s. Each GOP has 16 frames, which were decomposed into four spatial and four temporal levels. We packetize these GOPs into packets of maximum size 500 bytes, ensuring that data from different subbands is not included in one packet. We consider three different bit-rates, 512 kps, 768 kps and 1024 kps and four different channel conditions, corresponding (after the link adaptation) to an equivalent P_L of 1%, 3%, 5%, and 10%. We use a five-dimensional feature vector with the content and channel features as described in Section II-B. The encoding delays are ignored and the maximum tolerable delay $Delay_{\max}$ was set to 200 ms. This limits the choice of the maximum retransmission limit possible to five. With this configuration, we have a total of $\sim$ 26 000 training feature vectors. We compute the optimal decision for each feature vector exhaustively, using the expected transmission times, and the expected distortion impact. We then use the SVM to partition this feature space into five distinct classes corresponding to retransmission limits of $\{5, 2, 1, 0, Discard\}$, where the *Discard*-class corresponds to dropping (not transmitting) the packet at the sender side. On the training data set, the proportion of correctly classified feature vectors is 87% for scheme *classif_1* and 93% for scheme *classif_2*. Of the 13% misclassified feature vectors (for *classif_1*) a majority of the misclassifications ($\sim$80%) result in the selection of lower retransmission limits than the optimal solution, and we observe a similar trend for *classif_2*.

B. Evaluation Against Optimal Strategies—Decoded PSNR

Figs. 3 and 4 show the resulting decoded PSNR for a test set comprising five GOPs (not included in the training set) from *Foreman* and *Mobile*. We compare *classif_1* and *classif_2* against the optimal solution (labeled "*exhaustive*") obtained by dynamic programming as discussed in Section II-A2. These results correspond to the PSNR averaged across ten runs having the same P_L, but different packet loss patterns.

For the *Foreman* sequence, *classif_1* has an accuracy of 83% and *classif_2* has accuracy 91% on the test data set. *Classif_2* performs worse than the exhaustive strategy by at most 0.3 dB,

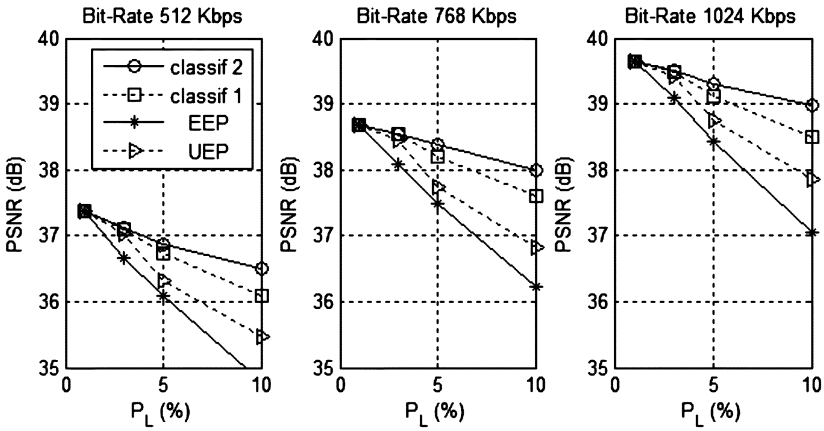

Fig. 5. Performance comparison—classification schemes versus *ad-hoc* schemes: *Foreman*.

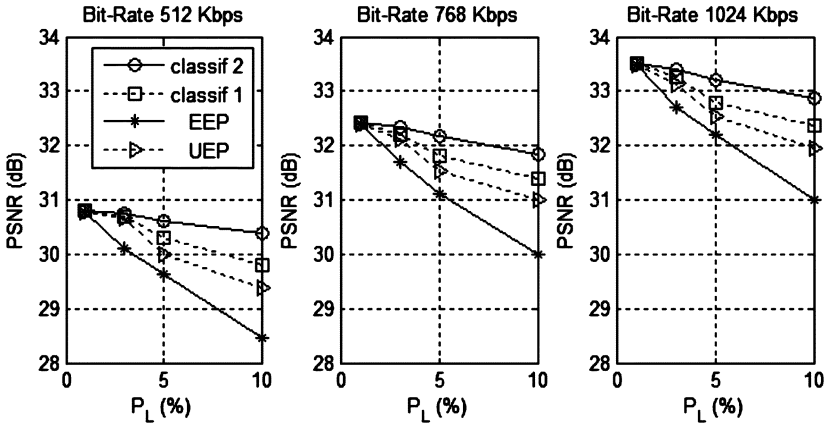

Fig. 6. Performance comparison—classification schemes versus *ad-hoc* schemes: *Mobile*.

which indicates that classification based algorithms can approach the performance of complex exhaustive algorithms (that are optimal in terms of minimizing the expected distortion). Finally, the PSNR results of *classif_2* are up to 0.6 dB better than those of *classif_1*.

C. Evaluation Against Ad-hoc Strategies—Decoded PSNR

We also compare these schemes against the *ad-hoc* protection strategies typically used in wireless multimedia streaming systems. We first compare against an *ad-hoc* scheme for equal error protection (EEP), which corresponds to the case where no classification is used, and all packets have a fixed retry limit of two. Subsequently, we also compared our proposed cross-layer system performance against an *a priori* determined unequal error protection (UEP) scheme, that assigns different retry limits based on the temporal level that the packet belongs to (the most accurate feature as determined in Table IV). As our test set, we used five new GOPs from *Foreman* and *Mobile* that were not part of the training set, and present the average PSNR in Figs. 5 and 6.

Clearly, the EEP has the worst performance while classification-based schemes achieve the highest PSNR performance. The improvements achieved by the classification-based schemes are higher at higher packet loss rates and bit-rates, where a judicious allocation of the redundancy/protection has a large impact on the distortion. At packet loss rates of 10%, *classif_1* outperforms the UEP scheme by ∼0.5–0.7 dB, and *classif_2* outperforms UEP by ∼1.2 dB. Since the performance of the UEP scheme (that is similar to the classification scheme, but uses only one feature) is close to the classification-based schemes at lower loss rates, we can further reduce the complexity of our schemes by using UEP when the packet loss rate is lower than 5%, and employ the full classifiers only at higher loss rates.

D. Training Test Mismatch

Next, in order to examine the performance of our schemes under train-test mismatch, we show results on data significantly different from the training set, in terms of: content characteristics; channel characteristics; encoder parameters and settings.

- *Content Characteristics Mismatch*
  We present results on five GOPs of the *Football* and *Flowergarden* sequences, with our classifier trained on the *Foreman* and *Mobile* sequences in Figs. 7 and 8.
  It is clear from these results that our PSNR results for *Foreman* and *Mobile* also extend to the *Flowergarden* and *Football* sequences, even though no data from these sequences was used to train the classifier (the actual classifier accuracy drops to 79% for *classif_1* and 89% for *classif_2*). This shows that the classifier can be trained

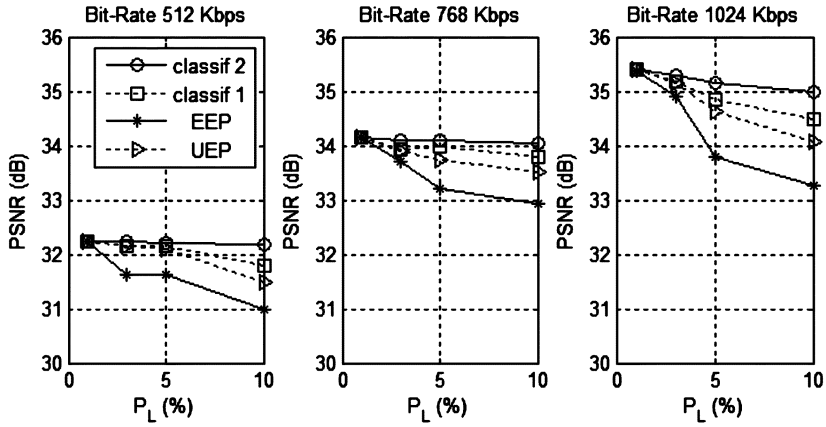

Fig. 7. Content characteristics mismatch: *Flowergarden*.

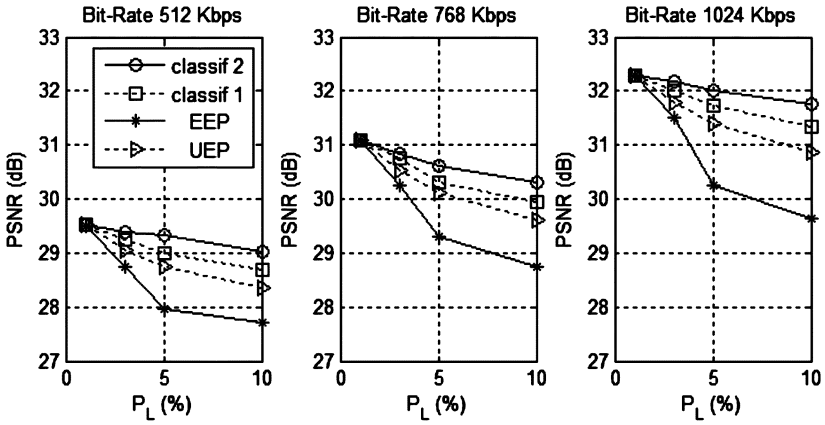

Fig. 8. Content characteristics mismatch: *Football*.

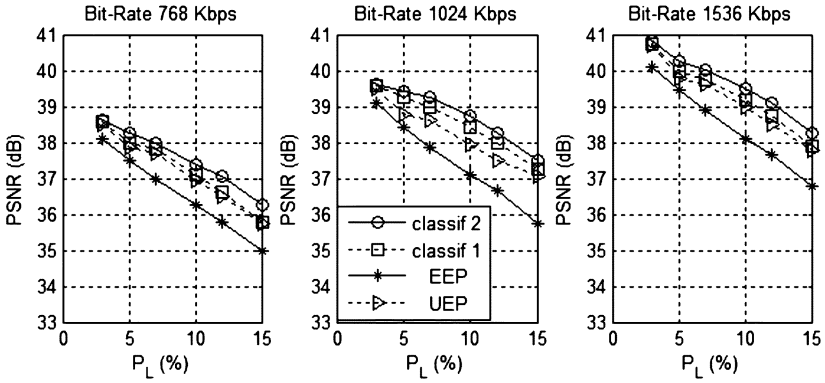

Fig. 9. Channel parameters mismatch: *Foreman*.

with a small representative set of data and be used on a much wider set without significant loss of performance. It must be mentioned that the performance of the classification-based schemes can be further improved through online learning, where the classifier is tuned in real-time to the specific characteristics of the test sequence.

- *Channel Parameters Mismatch*
  To assess the efficiency of the proposed scheme when the channel parameters at transmission time do not match the range of loss-rates and bit-rates used during training, we present a new set of results. Specifically, we use training data from the *Foreman* and *Mobile* sequences at only 512 kps and 1024 kps and at P_L corresponding to 3% and 10%. We present results at packet loss rates and bit-rates both within these ranges as well as outside these ranges. We show results for five GOPs (that were not part of the training set) from the *Foreman* and *Flowergarden* sequences (to show the variation in performance when we use sequences with different levels of similarity with the training set) in Figs. 9 and 10.
  We can see from these results that the classification-based schemes continue to outperform the UEP even when a

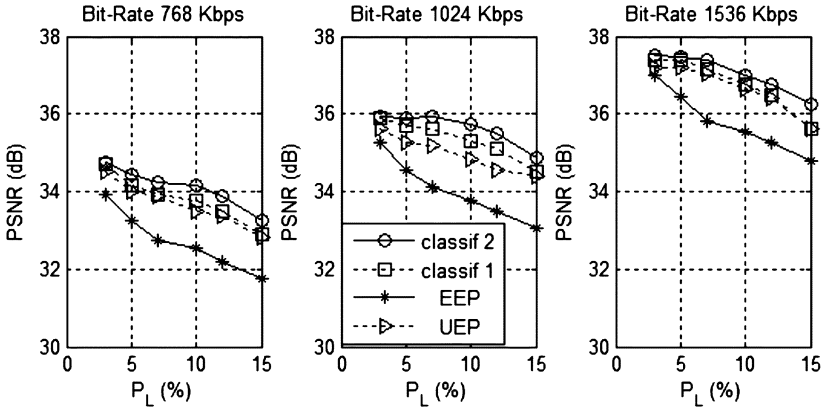

Fig. 10. Content and channel parameters mismatch: *Flowergarden*.

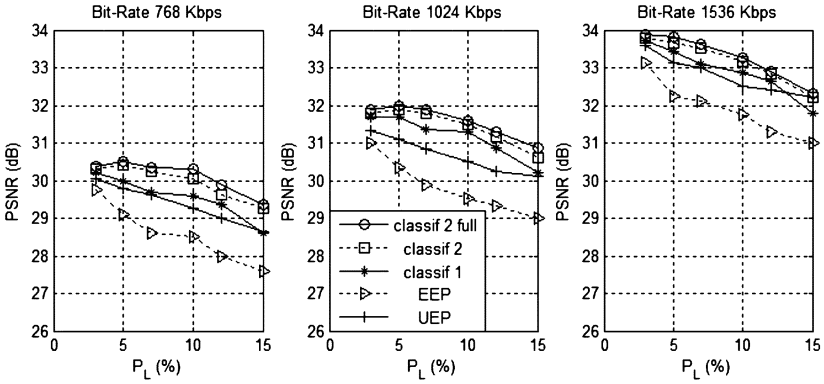

Fig. 11. Content and channel parameters mismatch (using various training sets): *Football*.

severe mismatch occurs between the training and testing channel conditions. However, the improvements over UEP decrease from ∼1 dB (in experiments with no channel train-test mismatch) to ∼0.5 dB. Again, this is significant, as we can train the classifiers at only a small set of channel conditions without sacrificing the performance significantly. Online learning can also help in this case to improve the performance of the cross-layer optimization in real-time, if a significant mismatch is detected. Of course, as the number of training points increases, the performance of the classification based schemes will also increase. In order to highlight this, in Fig. 11, we additionally show the performance (for the *Football* sequence) that can be obtained by *classif_2*, if the training set (using *Foreman* and *Flowergarden*) also included additional packet-loss rates (1%, 3%, 5%, 10%) and bit-rates (512 kps, 768 kps and 1024 kps). To illustrate this point, we superimpose these results on the results obtained with the limited training set. *Classif_2* (full) corresponds to the results of *classif_2* with the expanded training set, and we can see that these results are better than using the reduced set. The improvements over the UEP scheme are around 1 dB. An interesting observation is that the UEP results approach the results of the classification based strategies at a packet loss rate of 15% (especially at 1536 Kbps). This is because the packet loss rate is so high that the classifier can have only a limited impact over the *ad-hoc* UEP, as most packets are discarded, and only the most important packets (that are indicated by the temporal level) are assigned a nonzero retry limit.

- *Encoding Parameter Mismatch*
  Finally, we also consider encoding parameter train-test mismatch. In particular, for training we used *Foreman* and *Mobile* sequences with a GOP of 16 frames with four temporal decomposition levels, and we test the performance on sequences with GOPs of eight and 32 frames, corresponding to three and five temporal levels, respectively. Figs. 12 and 13 show the resulting PSNR performance. We plotted the results for 10% packet-loss rates only, where the importance of selecting the right cross-layer strategies is essential in terms of distortion impact. Changing the GOP structure clearly affects our most important feature, i.e., the (number of) temporal levels. However, since we order the temporal levels in decreasing order of importance, our classifier based schemes automatically adjust to a different number of temporal levels/frames by grouping together frames from less important levels. The experiments were performed for data from *Foreman* and *Flowergarden* with a GOP structure of 32 frames (Fig. 12) and eight frames (Fig. 13), respectively.
  The results indicate that the proposed classification solution maintains its performance even if the high-level encoding parameters (GOP structure) change. We are investigating the impact of mismatches in other encoding parameters as a direction for future research.

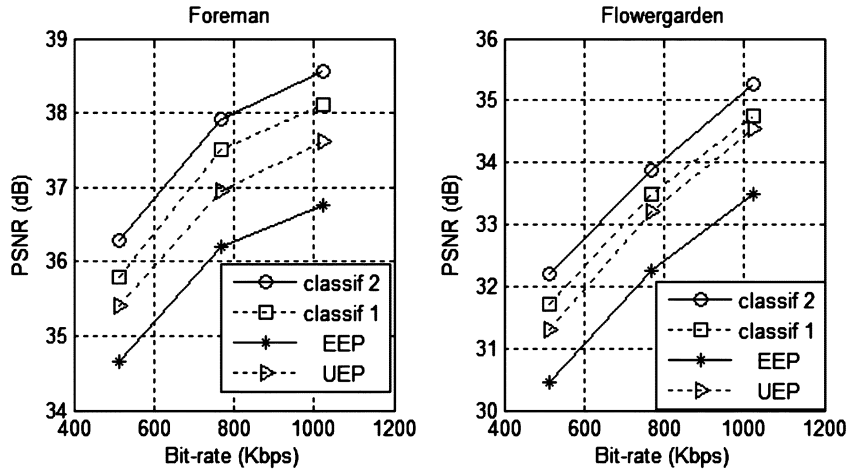

Fig. 12. Content and encoding parameter mismatch $(\mathrm{GOP} = 32) P_L = 10\%$.

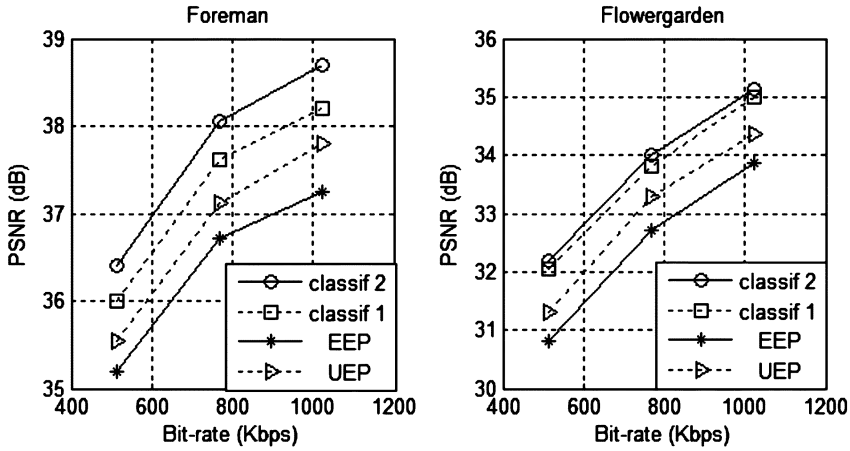

Fig. 13. Content and encoding parameter mismatch $(\mathrm{GOP} = 8) P_L = 10\%$.

E. Validation Experiments Using Real Wireless Channel Conditions

We validate the efficiency of the proposed classification-based system using our wireless streaming test-bed. (The used test-bed was described partially in [9]). In this way, we can assess the efficiency of our system under real wireless channel conditions and link adaptation mechanisms currently deployed in state-of-the-art 802.11 a/g wireless cards. Link adaptation selects one appropriate physical-layer mode (modulation and channel coding) depending on the link condition, in order to continuously maximize the experienced goodput [9], [33]. We captured the packet-loss pattern under different channel conditions (described here by the link SNR) using our wireless streaming test-bed. We compare the performance of our proposed classification-based cross-layer strategy against the optimal exhaustive strategy for these real wireless channel traces in Table VI. The results demonstrate that under varying SNR, the proposed cross-layer mechanism leads to a decrease in PSNR of $\sim$0.7 dB as compared with the optimal strategy. The obtained classification-based results outperformed by 3–5 dB current *ad-hoc* retransmission strategies available in the wireless card. Similar results were observed for a variety of alternative sequences and transmission scenarios.

F. Validation Experiments for Alternative Video Codec

To further assess the performance of the proposed classification based scheme, we compare its performance against the exhaustive optimal cross-layer approach for a different video coding scheme. Specifically, we use the scalable MCTF-based video coding scheme [36], which is based on $2D+t$ rather than t $t+2D$ spatio–temporal decomposition structures (see [16] for a summary of the key differences between $2D+t$ and $t+2D$ wavelet video coding structures). The alternative scheme uses significantly different motion-estimation, compensation (the estimation and MCTF are performed in the overcomplete wavelet domain rather than in the spatial domain like in the original coder used) and, importantly, packetization and entropy coding techniques. However, the GOP and temporal prediction structure was kept similar with that of the original $t+2D$ coder in order to preserve similar delay deadlines for the various frames. Importantly, to highlight the robustness and efficiency of the proposed cross-layer strategy, we used the same classification scheme and existing classes of the original video coder. Also, the features we use for our classifier may be easily computed for this new codec as well.

The results shown Table VII highlight that the proposed classification-based scheme is competitive as compared with the op-

TABLE VI
DECODED PSNR FOR REAL WIRELESS PACKET LOSS TRACES

Measured channel SNR	Foreman PSNR (dB)		Mobile PSNR (dB)	
	Classification	Exhaustive	Classification	Exhaustive
Poor channel conditions (12-15dB)	33.81	34.29	26.31	26.50
Average channel conditions (15-20dB)	35.90	36.06	28.48	29.26
Very good channel conditions (20-25dB)	38.64	38.66	31.82	32.01

TABLE VII
PERFORMANCE COMPARISON BETWEEN OPTIMAL AND CLASSIFICATION-BASED CROSS-LAYER STRATEGIES FOR AN ALTERNATIVE VIDEO CODER

Packet-loss rate	Foreman		Mobile	
	Classification	Exhaustive	Classification	Exhaustive
3%	37.25	38.09	28.90	28.93
5%	36.29	37.08	28.35	28.60
10%	30.38	31.36	22.54	23.03

timal approach (within 1 dB) for a variety of sequences and packet-loss rates. The average packet-size was 500 bytes, and the transmission rate was 1024 Kbps in these experiments. This is important, since it shows that different video coders deploying similar prediction structures and prioritization algorithms can use the same classification engine for cross-layer optimization.

IV. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we address the problem of cross-layer optimization for wireless video in real-time, using a classification-based framework. In this framework, instead of using complex dynamic programming or *ad-hoc* solutions, a learning based approach is applied, where content, coder-specific and channel features are used to efficiently predict the optimal cross-layer strategy. Specifically, we investigate the cross-layer problem of assigning optimal MAC retry limits for video transmission under delay constraints. The deployed features are easily extracted (from available metadata) or determined in real-time. Our results indicate that the proposed classification methods perform within 0.3 dB of the optimal solution, and outperform *ad-hoc* approaches by $\sim$1 dB. We observe that the performance of our schemes does not degrade significantly with train-test mismatches and can conclude that the designed classifier schemes are robust. We further validate our results using real wireless channel traces. Finally, we also investigate the extensibility of our approach by presenting results with two different scalable video coders.

There are several open issues requiring further exploration. Firstly, though our framework is general and can be applied to a variety of cross-layer optimizations, a comprehensive study involving various strategies and parameters at different layers of the OSI stack is needed for a completely optimized wireless transmission system. Secondly, investigating the performance of the proposed solution in conjunction with different nonscalable video coders such at MPEG-2 or H.264 is essential in order to determine which set of features and parameters are adequate for such widely used coders. Finally, alternative techniques for classification, including neural nets, logistic regression, decision trees etc., need to be explored, especially in order to consider dynamically varying systems where online tuning of the classification-based strategies may be required. For instance, in [34], the authors compare ten algorithms in terms of different performance metrics such as threshold or ordering metrics as well as probability metrics. These results can be used as a basis for future research aimed at exploring efficiency-cost tradeoffs for our proposed cross-layer classification system as they provide criteria of selecting the appropriate algorithm given the underlying data characteristics, as well as the optimization goal.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the associate editor for their suggestion and comments, which significantly improved this manuscript.

REFERENCES

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Ref. ISO/IEC 8802-11:1999(E), IEEE Std. 802.11, 1999.

[2] *Compressed Video Over Networks*A. Reibman and M.-T. Sun, Eds.. New York: Marcel Dekker, 2000.

[3] L. Qian, D. L. Jones, K. Ramchandran, and S. Appadwdula, "A general joint source-channel matching method for wireless video transmission," in *Proc. DCC*, Snowbird, Mar. 1999, pp. 414–423.

[4] D. Majumdar, 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.

[5] Y. Pei and J. Modestino, "Multi-layered video transmission over wireless channels using an adaptive modulation and coding scheme," in *Proc. IEEE ICIP 2001*, Thessaloniki, Greece, Oct. 2001.

[6] Y. Shan and A. Zakhor, "Cross layer techniques for adaptive video streaming over wireless networks," in *Proc. IEEE ICME 2002*, Lausanne, Aug. 2002.

[7] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer design for wireless networks," *IEEE Commun. Mag.*, Oct. 2003.

[8] 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.

[9] 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. Sel. Areas Commun.*, vol. 21, no. 10, pp. 1752–1763, Dec. 2003.

[10] M. van der Schaar and S. Shankar, "Cross-layer wireless multimedia transmission: challenges, principles and new paradigms," *IEEE Wireless Commun. Mag.*, vol. 12, no. 4, pp. 50–58, Aug. 2005.

[11] D. Turaga and M. van der Schaar, "Cross-layer aware packetization strategies for optimized wireless multimedia transmission," in *Proc. IEEE ICIP 2005*, Genova, Italy, Sep. 2005.

[12] 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. 278–292, Oct.–Dec. 2002.

[13] W. Kumwilaisak, T. Hou, Q. Zhang, W. Zhu, C.-C. J. Kuo, and Y.-Q. Zhang, "A cross-layer quality of service mapping architecture for video delivery in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 10, pp. 1685–1698, Dec. 2003.
[14] J. Shin, J. Kim, J. Kim, and C.-C. J. Kuo, "Dynamic QoS mapping control for streaming video in relative service differentiation networks," *Eur. Trans. Telecommun.*, vol. 12, no. 3, pp. 217–230, 2001.
[15] J.-R. Ohm, "Advances in scalable video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 42–56, Jan. 2005.
[16] J.-R. Ohm, M. van der Schaar, and J. Woods, "Interframe wavelet coding—motion picture representation for universal scalability," *Signal Process.: Image Commun., Special Issue on Digital Cinema*, vol. 19, no. 9, pp. 877–908, Oct. 2004.
[17] S.-J. Choi and J. W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 155–167, Feb. 1999.
[18] D. S. Turaga and T. Chen, "Classification based mode decisions for video over networks," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 41–52, Mar. 2001.
[19] R. Wong, S. Shankar, and M. van der Schaar, "Integrated application-MAC modeling for cross-layer optimized wireless video," in *Proc. ICC 2005*, Oct. 2005, vol. 2, pp. 1271–1275.
[20] M. Aizerman, E. Braverman, and L. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning," *Automat. Remote Contr.*, vol. 25, pp. 821–837, 1964.
[21] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[22] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods)*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[23] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[24] M. Karcewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 637–644, Jul. 2003.
[25] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Norwell, MA: Kluwer, 2002.
[26] N. Dimitrova, H. J. Zhang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor, "Applications of video content analysis and retrieval," *IEEE Multimedia*, vol. 9, no. 3, pp. 42–55, Jul.–Sep. 2002.
[27] A. Girgensohn and J. Foote, "Video classification using transform coefficients," in *Proc. ICASSP*, Mar. 1999, vol. 6, pp. 3045–3048.
[28] Statistical Pattern Recognition Toolbox for Matlab [Online]. Available: http://cmp.felk.cvut.cz/~xfrancv/stprtool/
[29] C.-W. Hsu and C.-J. Lin, "A comparison on methods for multi-class support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.
[30] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl.Disc.*, vol. 2, no. 2, pp. 1–47, 1998.
[31] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
[32] H. Zhu and I. Chlamtac, "An analytical model for IEEE 802.11e EDCF differential services," in *ICCCN'03*, Oct. 2003.
[33] 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. 278–292, Oct. 2002.
[34] R. Caruana and A. Niculescu-Mizil, An Empirical Comparison of Supervised Learning Algorithms Using Different Performance Metrics Cornell Univ., Ithaca, NY, Tech. Rep. 2005-1973, 2005 [Online]. Available: http://www.cs.cornell.edu/~alexn/comparison.ecml05.submitted.pdf
[35] Y. Andreopoulos, A. Munteanu, J. Barbarien, M. van der Schaar, J. Cornelis, and P. Schelkens, "In-band motion compensated temporal filtering," *EURASIP Signal Process.: Image Commun., Special Issue on "Subband/Wavelet Interframe Video Coding")*, vol. 19, no. 7, pp. 653–673, Aug. 2004.
[36] J.-P. Ebert and A. Willig, A Gilbert-Elliot Bit Error Model and the Efficient Use in Packet Level Simulation Telecommunication Networks Group, Tech. Univ. Berlin, Germany, Tech. Rep. TKN-99-002, Mar. 1999.

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

Prior to joining the Electrical Engineering Department faculty at the University of California, Los Angeles (UCLA) on July 1, 2005, between 1996 and June 2003, she was a Senior Researcher at Philips Research in the Netherlands and the USA, where she led a team of researchers working on multimedia coding, processing, networking, and streaming algorithms and architectures. From July 1, 2003 until July 1, 2005, she was an Assistant Professor in the Electrical and Computer Engineering Department at the University of California, Davis. She has published extensively on multimedia compression, processing, communications, networking and architectures and holds 27 granted U.S. patents and several more pending. Since 1999, she was an active participant to the ISO Motion Picture Expert Group (MPEG) standard 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 MPEG-21 Scalable Video Coding, and also co-chairing the MPEG *ad-hoc* group on Multimedia Test-bed.

Dr. van der Schaar was elected as a Member of the Technical Committee on Multimedia Signal Processing of the IEEE Signal Processing Society. She was an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA and SPIE *Electronic Imaging Journal* from 2002- to 2005. Currently, she is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEM FOR VIDEO TECHNOLOGY (T-CSVT) and an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS. She received the NSF CAREER Award in 2004, IBM Faculty Award in 2005, 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. 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 35 journal and conference papers and two book chapters. He has 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. He received the IEEE T-CSVT 2006 Transactions Best Paper Award (with M. van der Schaar and B. Pesquet-Popescu), and is a coauthor for the 2006 IEEE ICASSP Best Student Paper (with H. Tseng, O. Verscheure, and U. Chaudhari).

Raymond Wong received the M.Sc. degree in 2001 from the University of California, Davis, where he is currently pursuing the Ph.D. degree.

From 1999 to 2002, he was with the Wireless Group at Advance Micro Devices (AMD). He is currently with the Verification Division at Cadence Design Systems, Inc., San Jose, CA.