Multimed Tools Appl (2016) 75:3235–3257
DOI 10.1007/s11042-014-2432-1

Cross-Layer prioritized H.264 video packetization and error protection over noisy channels

Kashyap K. R. Kambhatla · Seethal Paluri · John D. Matyjas · Sunil Kumar

Received: 21 April 2014 / Revised: 1 November 2014 / Accepted: 21 December 2014 /
Published online: 28 January 2015

Abstract Video transmission over wireless channels is affected by channel-induced packet losses. Distortion due to channel errors can be alleviated by applying forward error correction. Aggregating H.264/AVC slices to form video packets with sizes adapted to their importance can also improve transmission reliability. Larger packets are more likely to be in error but smaller packets require more overhead. We present a cross-layer dynamic programming (DP) approach to minimize the expected received video distortion by jointly addressing the priority-adaptive packet formation at the application layer and rate compatible punctured convolutional (RCPC) code rate allocation at the physical layer for prioritized slices of each group of pictures (GOP). Some low priority slices are also discarded to improve protection to more important slices and meet the channel bit-rate limitations. We propose two schemes. Our first scheme carries out joint optimization for all slices of a GOP at a time. The second scheme extends our cross-layer DP-based approach to slices of each frame by predicting the expected channel bit budget per frame for live streaming. The

Approved for Public Release; Distribution Unlimited: 88ABW-2014-5102, 4 Nov. 2014. This research was partially supported by awards from the U.S. Air Force Research Laboratory under contract #FA8750-08-1-0078 and #FA8750-11-1-0048. Opinions, interpretations, and conclusions are those of the authors and are not necessarily endorsed by the United States Government.

K. K. R. Kambhatla (✉)
Electrical and Computer Engineering, University of California San Diego, La Jolla, CA, USA
e-mail: kkambhat@ucsd.edu

S. Paluri
Computational Science Research Center, San Diego State University, San Diego, CA, USA
e-mail: spaluri@mail.sdsu.edu

J. D. Matyjas
Air Force Research Laboratory, Rome, NY, USA
e-mail: john.matyjas@us.af.mil

S. Kumar
Electrical and Computer Engineering, San Diego State University, San Diego, CA, USA
e-mail: skumar@mail.sdsu.edu

prediction uses a generalized linear model developed over the cumulative mean squared error per frame, channel SNR, and normalized compressed frame bit budget. The parameters are determined over a video dataset that spans high, medium and low motion complexity. The predicted frame bit budget is used to derive the packet sizes and corresponding RCPC code rates for live transmission using our DP-based approach. Simulation results show that both proposed schemes significantly improve the received video quality over contemporary error protection schemes.

Keywords H.264/AVC · Video compression · Cross-layer · Priority-adaptive packet formation · Packet discarding · Error protection · Live streaming

1 Introduction

Multimedia applications such as video streaming, which are delay sensitive and bandwidth intensive, are growing rapidly over wireless networks. However, existing wireless networks provide only limited and time-varying quality of service (QoS) support for these applications. Further, compressed video bitstream is vulnerable to packet losses over noisy wireless channels and the lost packets contribute different levels of video quality degradation due to temporal and spatial dependencies in the compressed bitstream. Video transmission can also tolerate some packet losses as the effect of lost packets can be concealed at the decoder [2]. Recent research has demonstrated the promise of cross-layer protocols for supporting the QoS demands of multimedia applications over wireless networks [44].

Adapting the packet size to channel error characteristics improves the successful packet transmission probability and reduces retransmissions [22, 26, 34]. It involves a trade-off between reducing the amount of overhead bits contributed by protocol headers at different layers by using large packet sizes and reducing the packet error rate by using small packet sizes. However, maximizing throughput in this manner does not guarantee the minimum received video distortion since lost video packets induce significantly different amounts of distortion. Hence, video packet size should also be adaptive to the packet importance. However, existing payload (i.e., packet size) adaptation schemes do not consider distortion contribution of the packet. Our recent cross-layer, priority-aware packet fragmentation schemes for H.264/AVC video at the medium access control (MAC) layer provided significant improvement in video quality over the priority-agnostic fragmentation schemes, thereby showing the advantages of adapting video fragment size to the packet importance [20]. However, the schemes in [20] did not use forward error correction (FEC) at the PHY layer for noisy wireless channels. MAC layer fragmentation usually requires that all the fragments of a packet are received error-free, otherwise the packet is discarded.

In this paper, our objective is minimizing the expected received video distortion by jointly optimizing the packet sizes (by aggregating the slices based on their importance) at the APP layer and estimating their FEC code rates to be applied at the PHY layer for noisy channels. Some low priority slices are also discarded in order to increase the protection to more important slices within the channel bit-rate constraints. Our proposed scheme ensures that higher priority slices, which contribute more distortion, are sent in smaller packets with stronger FEC coding. At the same time, it also controls the overhead incurred from the total protocol header bits and FEC codes associated with the formed packets. The distortion contributed by each slice is determined by its cumulative mean squared error (CMSE).

We propose two cross-layer dynamic programming (DP) based schemes. Our first scheme carries out joint optimization for all the slices of a group of pictures (GOP) together.

An earlier version of this scheme was presented in [21]. To avoid the delays associated with optimizing the packet sizes and their associated FEC code rates for entire slices of a GOP, our second scheme carries out the joint optimization on each frame independently by predicting its expected channel bit budget. For this, we train a generalized linear model (GLM) offline over the database of factors: (*a*) normalized CMSE per frame, (*b*) channel signal-to-noise ratio (SNR), and (*c*) normalized compressed frame bit budget allocated by the H.264 encoder. The factors are determined for a video dataset that spans high, medium, and low motion complexity. Simulation results show that both schemes achieve better received video quality than other contemporary schemes over noisy channels. Our scheme considers the point-to-point wireless communication environments which have also been considered in [3, 9, 17, 29, 46, 47]. Such communication environments are prevalent in military communication (such as the line of sight among airborne nodes or air to ground nodes), emergency communication, and device to device communication in certain industry applications (such as supervisory control and data acquisition (SCADA), critical industrial automation and control). In particular, a well-designed wireless network infrastructure cannot be assumed in the military and emergency communication, and the channel can also degrade due to node mobility, poor infrastructure (such as long distance between the nodes), and interference due to jamming etc. An important challenge in such applications is alleviating the channel-induced errors, which our scheme addresses by intelligently using the available channel resources to protect different video slices and frames in a way that the received video quality is improved over the noisy channels. The improved received video quality will also impact many content analysis applications, including the surveillance applications [13].

Contributions: Existing payload adaptation schemes [22, 26, 34] do not consider different distortion contributions (e.g., CMSE values) of video slices while computing their packet size nor do they discard low priority slices. Our approach has the following distinguishing features: (*i*) minimizes the video distortion by jointly optimizing the packet size and FEC code rate for all slices of a GOP, for a given source video bit rate, channel bit rate and channel SNR; (*ii*) adapts packet size and FEC code rate to the distortion contribution (i.e., CMSE values) of video slices; (*iii*) discards some low priority slices to improve protection to high priority slices while meeting the channel constraints; and (*iv*) performs optimization over slices of each frame (instead of slices of entire GOP) by using the predicted slice CMSE and frame overhead bit budget values for live streaming applications.

Section 2 discusses the related work. Section 3 gives an overview of our proposed cross-layer approach. Section 4 discusses the details of the joint optimization problem for the slices of a GOP. Section 5 discusses our joint optimization scheme for the slices of each frame. Details of the reference approaches and the performance comparison with our proposed schemes are presented in Section 6, followed by conclusions in Section 7.

2 Related work

Many schemes have been designed for fragmenting data units at the MAC layer [6, 11, 12, 19, 20, 23]. To address the variation in network conditions, solutions for adaptive packet size adjustments at the application layer have been discussed in [5, 7, 22, 25, 26, 30, 34, 39, 40, 43]. The effect of packet size on the loss rate and delay characteristics in a wireless streaming application was studied in [22]. It was shown that the application level packet size optimization could facilitate efficient use of wireless network resources, improving the service provided to the end users. Choudhury and Gibson [7] observed that payload length adaptation significantly improves the throughput at low channel SNRs.

Choi et al. [5] designed cross-layer schemes to study the effect of optimal packet size, MAC layer retransmissions, and application layer FEC on multimedia delivery over wireless networks. They noted that the packet size is tightly related to the packet delay and channel conditions. A mathematical framework to maximize a single user throughput by using the symbol rate, the packet length, and the constellation size of the modulation was described in [7]. A theoretical framework to optimize the single user throughput by adjusting the source bit rate and payload length as a function of channel conditions, without retransmission, was discussed in [8, 50].

Shih [39, 40] proposed a scheme which integrated the packet size control mechanism with the optimal packet-level FEC in order to enhance the efficiency of FEC over wireless networks. Both the degree of FEC redundancy and the transport packet size were adjusted simultaneously in accordance with a minimum bandwidth consumption strategy. To transmit video frames with delay bound and target frame error rate constraint, Lin et al. [30] formulated an optimization problem to minimize the required resource units for a single user by adjusting payload length, modulation, block size, and code rate for wireless channels. An adaptive packet and block length FEC control mechanism was discussed in [43]. An algorithm that allows an automatic repeat request (ARQ) protocol to dynamically optimize the packet size based on the wireless channel bit error rates was proposed in [34]. Lee at al. [25, 26] developed an analytic model to evaluate the impact of channel BER on the quality of streaming a MPEG-4 FGS scalable video. They proposed a video transmission scheme, which combines the adaptive assignment of packet size with unequal error protection to increase the end-to-end video quality. A cross-layer design considering retransmission was shown in [49]. Authors optimized the length of payload and suggested the associated physical transmission modes, which included modulation and coding scheme, for a given channel SNR.

The above mentioned schemes attempt to minimize the video distortion during wireless transmission without considering the different distortion contributions of video slices for jointly computing their packet size and FEC code rate. They also do not consider discarding some low priority slices which can further reduce the distortion compared to transmitting all the slices on bit-rate constrained channels.

3 Proposed cross-layer approach

Our scheme minimizes the expected received video distortion by using the slice priorities (i.e., slice CMSE values) and exploiting the trade-offs between the priority-adaptive packet sizes, and their RCPC code rates with the total incurred overhead (FEC + network protocol header) for a given source bit rate, channel SNR, and channel bit rate. Figure 1 illustrates a flow diagram of our proposed cross-layer approach at the transmitter. The APP layer carries out two functions: CMSE based slice prioritization and optimal packet formation (illustrated further in Fig. 2) for H.264 video slices.

3.1 CMSE computation/prediction of H.264 video slices

The video frames are encoded into a GOP using the fixed slice size configuration in H.264/AVC, where macroblocks of a frame are aggregated into slices with fixed size [14, 48]. The H.264 slices are prioritized based on their distortion contribution to the received video quality. The loss of a slice introduces error in the current reference frame and could propagate to other frames in the GOP. We compute the total distortion by using the CMSE

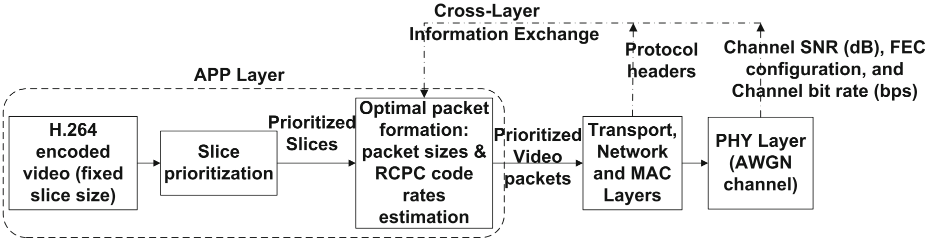

Cross-Layer
Information Exchange
APP Layer
Protocol headers
Channel SNR (dB), FEC configuration, and Channel bit rate (bps)
H.264 encoded video (fixed slice size)
Slice prioritization
Prioritized Slices
Optimal packet formation: packet sizes & RCPC code rates estimation
Prioritized Video packets
Transport, Network and MAC Layers
PHY Layer (AWGN channel)

Fig. 1 Flow diagram of proposed cross-layer system

introduced by a slice loss, since it takes into consideration the error propagation within the entire GOP. Suppose the video resolution is $H \times W$, represented in terms of the number of pixels along the height (H) and width (W) of a video frame. Let $\widehat{Pel}_{i,j,k}$ represent the pixel energy value at location (j, k) in the reconstructed frame i at the encoder without the slice loss and $\widetilde{Pel}_{i,j,k}$ represent the corresponding pixel energy value in the same frame decoded at the receiver with the slice loss. The CMSE contributed by the loss of the slice is computed in (1) as the sum of mean squared error (MSE) over the current and all the other frames in the GOP.

$$CMSE = \sum_{i=current\ frame\ with\ slice\ loss-t}^{last\ frame\ of\ GOP} \left\{ \frac{1}{H \times W} \sum_{j=1}^{H} \sum_{k=1}^{W} \left(\widehat{Pel}_{i,j,k} - \widetilde{Pel}_{i,j,k} \right)^2 \right\} \quad (1)$$

Here, t is the temporal duration of a reference frame in the backward direction. The bi-directionally predicted (B) frames in the backward temporal direction are also covered by (1).

The computation of slice CMSE introduces high computational overhead as it requires decoding the entire GOP for every slice loss. This overhead can be avoided by predicting the slice CMSE using our low-complexity generalized linear model (GLM) scheme proposed in [35]. This model reliably predicts the slice CMSE values by extracting the encoded frame and the error frame features. The encoded frame features consist of motion characteristics, signal characteristics, maximum residual energy, and number of macroblock sub-partitions. The error frame features consist of the temporal duration, initial mean square error, and initial structural similarity index. The slice contributing the highest distortion is the most important slice (i.e., highest priority). This process defines the relative importance order for the slices in the GOP [20]. Note that our joint video packetization and error protection

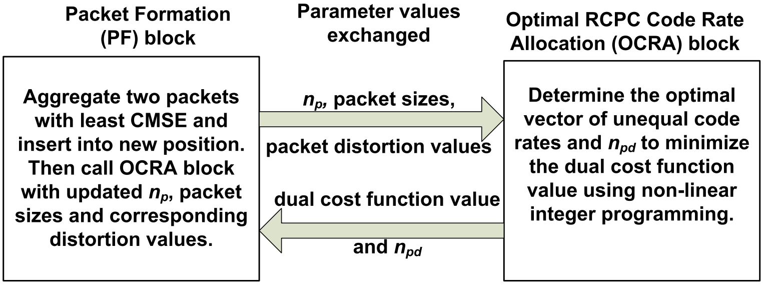

Packet Formation (PF) block
Parameter values exchanged
Optimal RCPC Code Rate Allocation (OCRA) block
Aggregate two packets with least CMSE and insert into new position. Then call OCRA block with updated n_p, packet sizes and corresponding distortion values.
n_p, packet sizes,
packet distortion values
dual cost function value
and n_{pd}
Determine the optimal vector of unequal code rates and n_{pd} to minimize the dual cost function value using non-linear integer programming.

Fig. 2 Block diagram of proposed dynamic programming approach

scheme proposed in this paper will also work well with other slice distortion computation schemes such as Li and Liu [27] and Schierl and Welzl [38].

3.2 Video packet formation

The optimal packet formation uses a joint optimization scheme to form variable-sized packets (by aggregating pre-encoded slices according to their CMSE) and estimating their corresponding optimal FEC code rates that can be applied at the PHY layer, in order to minimize the received video distortion.

The FEC configuration contains a mother code rate and a family of rate compatible punctured convolutional (RCPC) code rates [15]. We use BPSK modulation. The maximum transmission unit (MTU) size for the wireless network is 1500 bytes [31]. The optimal packet formation block uses the information about the MTU size, RTP/UDP, IP and MAC layer headers which remain unchanged for a given network, and the channel SNR, FEC configuration and channel bit rate information from the PHY layer. The RTP/UDP/IP overhead appended to each packet formed at the APP layer is 4 bytes after robust header compression (RoHC) [37]. Each packet is also appended with 50 bytes of MAC and PHY layer headers.

4 Expected video distortion minimization

We introduce a dynamic programming based optimization approach, denoted as DP-UEP(GOP) to minimize the expected video distortion for the slices of a GOP. The channel transmission rate is R_{CH} bits per second. The video is encoded at a frame rate of f_s frames per second. The total outgoing bit budget for a GOP of length L_G frames is $\frac{R_{CH} L_G}{f_s}$. n_s denotes the total number of slices generated within a GOP; n_s is a constant. n_p denotes the number of packets formed from these slices in the GOP; n_p is variable. $S_p(i)$ is the i^{th} packet size before adding network headers of size h bits and parity bits from the selected RCPC code. The RCPC code rates are chosen from a candidate set, **R**, of punctured code rates $\{R_1, R_2, R_3, ..., R_K\}$. The number of packets discarded is n_{pd}.

4.1 Packet size adaptation

The proposed scheme is a recursive process between two blocks: packet formation (PF) block and optimal RCPC code rate allocation (OCRA) block as shown in Fig. 2. The PF block initializes $n_p = n_s$ and $n_{pd} = 0$ and calls the OCRA block after sorting the $n_p = n_s$ packets of a GOP in descending order of their CMSE values. The OCRA block determines the optimal RCPC code rates for the packets and the number of packets discarded n_{pd}, to minimize a dual cost function value (computed over the GOP) which will be described in the next section. The number of packets is updated to $n_p = n_s - n_{pd}$. The OCRA block then forwards the computed parameters to the PF block as shown in Fig. 2 [21].

The PF block aggregates two packets with the least CMSE contribution from the remaining set of packets not discarded by the OCRA block. The aggregated packet is inserted into a new position in the sorted list based on its distortion computed as the sum of the CMSE values of both packets. This maintains the decreasing order of packet distortion. As an example, Fig. 3 shows one iteration of our proposed scheme in the PF block. The aggregated packet is at position $n_p - j$. The $n_p - 1$ packets with their sizes and distortion values are once

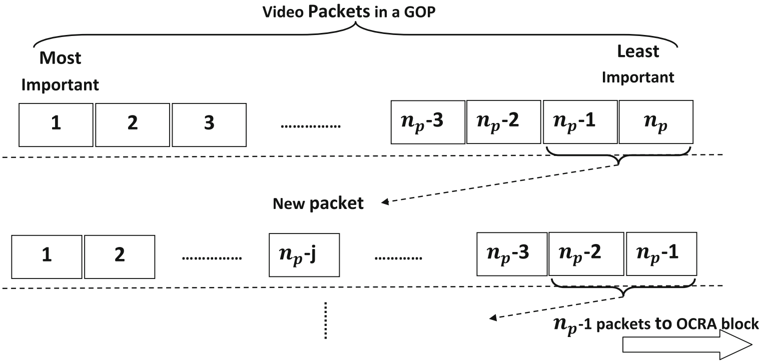


Fig. 3 Packet formation in PF block

again sent to the OCRA block which estimates their new optimal packet code rates. The parameters shown in Fig. 2 are exchanged recursively between the blocks until aggregating packets no longer reduces the dual cost function value [21].

The size of the aggregated packets is constrained by the MTU size for wireless networks. Aggregating packets reduces the total overhead from network protocol headers; the bits saved are used to increase the FEC protection to more important packets. Since the PF block aggregates the least important packets in each iteration, this ensures that packets contributing higher distortion are transmitted with smaller sizes, and the OCRA block ensures that they have stronger FEC hence lower packet error probabilities.

4.2 Distortion minimization with OCRA block

The distortion due to compression is neglected in this formulation because the slices are encoded at relatively high quality. Therefore, the distortion due to compression is small compared to distortion from slice losses and discards. The initial values are $n_p = n_s$ and $n_{pd} = 0$. The expected video distortion within a GOP, $E[\tilde{D}_{GOP}]$ is modeled as the sum of the distortion due to channel-induced packet loss and distortion from packets discarded at the sender as in [29]. The distortion due to the compression is neglected in this formulation because the slices are encoded at a relatively high quality. So, the distortion due to compression is small compared to distortion from packet losses and discards.

$$E[\tilde{D}_{GOP}] = \sum_{i=1}^{n_p - n_{pd}} E[\tilde{D}_p(i)] + \sum_{i=n_p-n_{pd}+1}^{n_p} D_p(i) \tag{2}$$

$D_p(i)$ is the distortion caused due to the loss of packet i and is computed as the sum of the CMSE of individual slices contained in the packet. Each video packet is appended with a h bit network header and the parity bits for a code rate r_i selected from the set $\mathbf{R}$. A video packet is in error if at least one bit is in error after channel decoding at the receiver. The packet error probability $p_{pkt}(i)$, which depends on the channel SNR, packet size, and the selected RCPC code rate is estimated as given in [3, 7, 29, 32]:

$$p_{pkt}(i) = 1 - (1 - p_b(SNR, r_i))^{\left(\frac{h+S_p(i)}{r_i}\right)} \tag{3}$$

where $p_b(SNR, r_i)$ is the bit error probability after channel decoding for code rate r_i. We use the packet error probability estimate in the OCRA block to determine the packet FEC rates. For a given value of n_{pd}, the distortion due to the discarded packets in (2) is a constant K_1. The optimization problem for minimizing expected video distortion over the GOP by allocating optimal code rates is formulated as in [21],

$$\begin{gathered}
\min_{\mathbf{r}} \left\{ \sum_{i=1}^{n_p - n_{pd}} \left[1 - (1 - p_b(SNR, r_i))^{\left(\frac{h+S_p(i)}{r_i}\right)} \right] D_p(i) + K_1 \right\} \\
= K_1 + \min_{\mathbf{r}} \left\{ \sum_{i=1}^{n_p - n_{pd}} \left[1 - (1 - p_b(SNR, r_i))^{\left(\frac{h+S_p(i)}{r_i}\right)} \right] D_p(i) \right\} \\
\text{subject to} \\
(C1) \quad \sum_{i=1}^{n_p - n_{pd}} \frac{h+S_p(i)}{r_i} \le \left(\frac{R_{CH} L_G}{f_s} \right) \\
(C2) \quad r_{i-1} \le r_i \quad \text{for} \quad i = 2, 3, 4, 5, 6, ..., (n_p - n_{pd}) \\
\text{where} \quad \mathbf{r} = \left[r_1, r_2, ..., r_{n_p - n_{pd}} \right] \text{ and } r_i \in \mathbf{R}
\end{gathered} \tag{4}$$

Constraint 1 in (4) is the channel bit rate constraint. Constraint 2 ensures that higher priority packets have code rates at least as good as those allocated to lower priority packets. This speeds up the optimization process by narrowing down the selection set of packet code rates. To solve this non-linear integer programming problem, we first relax the constrained optimization problem in (4) to an unconstrained problem [28]. By absorbing the constraints into the objective using Lagrange multipliers $\boldsymbol{\lambda} = \left[\lambda_1, \lambda_2, ..., \lambda_{n_p - n_{pd}}\right]$ with each $\lambda_i \in \mathbb{R}^+$, we construct the Lagrangian cost function as in [21],

$$\begin{aligned}
F_{GOP}(\mathbf{r}, \lambda) = K_1 + \sum_{i=1}^{n_p - n_{pd}} & \left[1 - (1 - p_b(SNR, r_i))^{\left(\frac{h+S_p(i)}{r_i}\right)} \right] D_p(i) \\
& + \lambda_1 \left(\sum_{i=1}^{n_p - n_{pd}} \frac{h+S_p(i)}{r_i} - \frac{R_{CH} L_G}{f_s} \right) + \sum_{i=2}^{n_p - n_{pd}} \lambda_i (r_{i-1} - r_i)
\end{aligned} \tag{5}$$

$$\text{where} \quad \lambda = \left[\lambda_1, \lambda_2, ..., \lambda_{n_p - n_{pd}} \right]$$

We form the dual cost function $d_{GOP}(\boldsymbol{\lambda})$ by minimizing the Lagrangian cost function for a given $\boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ is searched using a subgradient approach discussed in the next section. Let $\mathcal{C}$ be the space of all possible combinations of $r_i, i = 1, 2, ..., n_p - n_{pd}$ selected from $\mathbf{R}$ that can be applied to the packets before transmission. The dual function is computed as in [21],

$$\begin{aligned}
d_{GOP}(\lambda) &= \min_{\mathbf{r} \in \mathcal{C}} \; F_{GOP}(\mathbf{r}, \lambda) \\
&= \min_{\mathbf{r} \in \mathcal{C}} \; \sum_{i=1}^{n_p - n_{pd}} \left\{ p_{pkt}(i) D_p(i) \right\} + \lambda_1 \left(\sum_{i=1}^{n_p - n_{pd}} \frac{h+S_p(i)}{r_i} - \frac{R_{CH} L_G}{f_s} \right) \\
&\qquad + \sum_{i=2}^{n_p - n_{pd}} \lambda_i (r_{i-1} - r_i) + K_1 \\
&= \min_{\mathbf{r} \in \mathcal{C}} \left\{ \sum_{i=1}^{n_p - n_{pd}} \left(p_{pkt}(i) D_p(i) + \lambda_1 \left(\frac{h+S_p(i)}{r_i} \right) \right) + \sum_{i=2}^{n_p - n_{pd}} \lambda_i (r_{i-1} - r_i) \right\} \\
&\qquad + K_2
\end{aligned} \tag{6}$$

$K_2 = K_1 - \lambda_1 \left(\frac{R_{CH}L_G}{f_s}\right)$ in (6) is a constant and the computation of $d_{GOP}(\boldsymbol{\lambda})$ can be further simplified as follows. Let $A(r_i) = p_{pkt}(i)D_p(i) + \lambda_1 \left(\frac{h+S_p(i)}{r_i}\right)$. Then we can modify the first term in (6) as in [21],

$$
\begin{aligned}
&\min_{\mathbf{r}\,\epsilon\,\mathcal{C}} \left\{\sum_{i=1}^{n_p-n_{pd}} A(r_i) + \sum_{i=2}^{n_p-n_{pd}} \lambda_i(r_{i-1} - r_i)\right\} \\
&= \min_{\mathbf{r}\,\epsilon\,\mathcal{C}} \left\{A(r_1) + A(r_2) + \ldots + A(r_{n_p-n_{pd}}) + \lambda_2(r_1 - r_2) + \ldots + \lambda_{n_p-n_{pd}}(r_{n_p-n_{pd}-1} - r_{n_p-n_{pd}})\right\} \\
&= \min_{r_1\,\epsilon\,\mathbf{R}} \{A(r_1) + \lambda_2(r_1)\} + \min_{r_2\,\epsilon\,\mathbf{R}} \{A(r_2) - \lambda_2(r_2) + \lambda_3(r_2)\} \\
&\quad + \ldots + \min_{r_{n_p-n_{pd}-1}\,\epsilon\,\mathbf{R}} \left\{A(r_{n_p-n_{pd}-1}) - \lambda_{n_p-n_{pd}-1}(r_{n_p-n_{pd}-1}) + \lambda_{n_p-n_{pd}}(r_{n_p-n_{pd}-1})\right\} \\
&\quad + \min_{r_{n_p-n_{pd}}\,\epsilon\,\mathbf{R}} \left\{A(r_{n_p-n_{pd}}) - \lambda_{n_p-n_{pd}}(r_{n_p-n_{pd}})\right\} \\
&= \min_{r_1\,\epsilon\,\mathbf{R}} \{A(r_1) + \lambda_2(r_1)\} + \sum_{i=2}^{n_p-n_{pd}-1} \min_{r_i\,\epsilon\,\mathbf{R}} \{A(r_i) + r_i(\lambda_{i+1} - \lambda_i)\} \\
&\quad + \min_{r_{n_p-n_{pd}}\,\epsilon\,\mathbf{R}} \left\{A(r_{n_p-n_{pd}}) - \lambda_{n_p-n_{pd}}(r_{n_p-n_{pd}})\right\}
\end{aligned}
$$

The dual function can now be expressed in terms of function $A(r_i)$ as,

$$
\begin{aligned}
d_{GOP}(\lambda) &= K_2 + \min_{r_1\,\epsilon\,\mathbf{R}} \{A(r_1) + \lambda_2(r_1)\} + \sum_{i=2}^{n_p-n_{pd}-1} \min_{r_i\,\epsilon\,\mathbf{R}} \{A(r_i) + r_i(\lambda_{i+1} - \lambda_i)\} \\
&\quad + \min_{r_{n_p-n_{pd}}\,\epsilon\,\mathbf{R}} \left\{A(r_{n_p-n_{pd}}) - \lambda_{n_p-n_{pd}}(r_{n_p-n_{pd}})\right\} \\
&= K_2 + \sum_{i=1}^{n_p-n_{pd}} \min_{r_i\,\epsilon\,\mathbf{R}} \tilde{F}_{GOP,i}(r_i, \lambda_i)
\end{aligned}
\tag{7}
$$

$$
\text{where } \tilde{F}_{GOP,i}(r_i, \lambda_i) = \begin{cases} A(r_1) + \lambda_2(r_1) \text{ for } i = 1 \\ A(r_i) + r_i(\lambda_{i+1} - \lambda_i) \text{ for } i = 2, 3, 4, \ldots, n_p - n_{pd} - 1 \\ A(r_{n_p-n_{pd}}) - \lambda_{n_p-n_{pd}}(r_{n_p-n_{pd}}) \text{ for } i = n_p - n_{pd} \end{cases}
$$

The minimum of the dual cost function for a given $\boldsymbol{\lambda}$ can be found by minimizing the sub-Lagrangian cost functions $\tilde{F}_{GOP,i}(r_i, \lambda_i)$ individually. The solution space of the minimization of $F_{GOP}(\mathbf{r}, \boldsymbol{\lambda})$ is $(K+1)^{(n_p-n_{pd})}$. Since we can minimize the sub-Lagrangians individually, $d_{GOP}(\boldsymbol{\lambda})$ can be computed with only $(n_p - n_{pd})(K+1)$ evaluations of $\tilde{F}_{GOP,i}(r_i, \lambda_i)$ and comparisons [28]. This reduces the computational complexity involved in deriving the optimal set of packet sizes and their code rates. The frame based optimization scheme discussed in Section 5 uses only the slices of a frame (instead of a GOP) to form packets. Therefore, the optimization complexity is significantly reduced compared to the DP-UEP(GOP) scheme.

4.2.1 Determination of λ

We use the subgradient method [28] to search for the best $\boldsymbol{\lambda}$ over the space $\mathcal{C}$. The dual function $d_{GOP}(\boldsymbol{\lambda})$ is a concave function of $\boldsymbol{\lambda}$ even when the problem in the primal domain is not convex [28]. Therefore the optimal $\boldsymbol{\lambda}$ is found by solving $\max_{\boldsymbol{\lambda}\,\epsilon\,\mathbb{R}^+} d_{GOP}(\boldsymbol{\lambda})$. Since the dual is a piecewise linear concave function [28], it may not be differentiable at all points. Nevertheless, subgradients can still be found and are used to compute the optimal value [28]. It can be shown that the subgradient is a descent direction of the Euclidean distance to the set of maximum points of the dual function [28]. This property is used in the subgradient method for the optimization of a non-smooth function. The subgradient method is an iterative search algorithm for $\boldsymbol{\lambda}$. In each iteration, λ_i^{k+1} is updated by the subgradient ξ_i^k of $d_{GOP}(\boldsymbol{\lambda})$ at λ_i^k as in [21],

$$\lambda_i^{(k+1)} = \max(0, \lambda_i^k + s_k \xi_i^k / \|\xi^k\|) \tag{8}$$

where s_k is the step size. Based on the derivation in [28], the subgradients $\boldsymbol{\xi}^k$ of $d_{GOP}(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda}^k$ are

$$\begin{array}{l} \xi_1^k = g(\mathbf{r}^k) - \frac{R_{CH}L_G}{f_s} = \sum_{i=1}^{n_p-n_{pd}} \left(\frac{h+S_p(i)}{r_i}\right) - \frac{R_{CH}L_G}{f_s} \\ \xi_i^k = r_{i-1} - r_i \text{ for } i = 2, 3, 4, ..., n_p - n_{pd} \end{array} \tag{9}$$

where $g(.)$ is the rate constraint function of the problem and $\mathbf{r}^k = \left[r_1^k, r_2^k, ..., r_{n_p-n_{pd}}^k\right]$ is the solution to the term $\min_{\mathbf{r} \in \mathcal{C}} \ F_{GOP}(\mathbf{r}, \boldsymbol{\lambda}^k)$ in (6).

4.3 Discarding packets

By explicitly discarding a small number of low priority packets, our scheme gains additional room for packet size adaptation and FEC, and can achieve significant benefits overall [21]. To allow either the discarding of less important packets or sending them unprotected, the candidate set of punctured code rates $\mathbf{R}$ is modified to $\{1, R_1, R_2, R_3, ..., R_K, \infty\}$. This neither changes the objective function to be minimized in (4), nor does it affect the above optimization algorithm. If the code rate of packet i, $r_i = \infty$, then its probability of bit error $p_b(SNR, r_i) = 1$ causing it to be discarded. The induced distortion is accounted for in the overall expected distortion $E[\tilde{D}_{GOP}]$ through component K_1 in (4). If $r_i = 1$, the video packet is transmitted uncoded over the channel.

5 Frame-Level optimization using prediction

The DP-UEP(GOP) scheme was designed for a pre-encoded video and the cross-layer optimization was performed over the slices of each GOP. It's computational complexity and delay may not be suitable for live streaming applications such as live telecast of sports events. In this section, we extend our scheme to be applied over the slices of a single frame instead of the entire GOP, to reduce its computational complexity and delay. This requires the scheme, denoted as DP-UEP(frame) to perform optimization over the encoded slices of only one frame at a time. Since a typical GOP consists of different frame types (i.e., IDR, I, P and B), we require an estimate of the channel bit budget for each frame in order to allocate the protocol header and FEC bits to its packets. Moreover the different frame types generate different numbers of slices that contribute different amounts of distortion based on the error propagation and the video content. Therefore, we need to distribute the channel bit budget for a GOP among different frames. For this, we study the video factors which have the most influence on the expected channel bit budget estimate of a frame.

First, we analyze the channel bit budget allocation R_l for a frame l made by the DP-UEP(GOP) scheme. For this, its overhead bit budget proportion w_{ovh}^l is computed from the result of the DP-UEP(GOP) scheme as,

$$w_{ovh}^l = \frac{\text{\# FEC bits for frame } l}{\text{\# FEC bits for whole GOP}} \tag{10}$$

This quantity, while it is explicitly the fraction of FEC bits which a particular frame gets relative to the FEC bits for the whole GOP, is taken to be an estimate of overhead bits (both

FEC and protocol header bits) which the frame gets relative to the overhead bits for the whole GOP. For a video bit rate denoted by R_v, R_l is then evaluated as:

$$R_l = \sum_{i=1}^{n_s^l} S_p^l(i) + w_{ovh}^l \left\{ \frac{(R_{CH} - R_v)L_G}{f_s} \right\} \tag{11}$$

where $S_p^l(i)$ is the size of slice i and n_s^l is the number of slices in frame l. However, computing the w_{ovh}^l requires the knowledge of FEC bits allocated for the entire GOP which is not available in our frame-based approach. Therefore, we predict the value of w_{ovh}^l as discussed below.

5.1 w_{ovh}^l prediction

From the analysis of the DP-UEP(GOP) scheme, we observed that w_{ovh}^l for a frame l is dependent on the following video factors: (a) normalized compressed frame bit budget, w_c^l, (b) normalized frame CMSE, w_{cmse}^l, (c) channel SNR, and (d) video content. w_c^l is computed as the ratio of the size of the compressed frame l in bits to the total source bit rate for the GOP. w_{cmse}^l is computed as the ratio of the total CMSE contribution of all slices in frame l to the total CMSE contribution of all slices in the GOP.

$$w_c^l = \frac{\sum_{i=1}^{n_s^l} S_p^l(i)}{\left(\frac{R_v L_G}{f_s}\right)} \quad ; w_{cmse}^l = \frac{\sum_{i=1}^{n_s^l} D_p^l(i)}{\sum_{j=1}^{L_G} \sum_{i=1}^{n_s^j} D_p^j(i)} \tag{12}$$

where $D_p^l(i)$ is the actual measured distortion (i.e., CMSE) due to the loss of slice i in frame l.

To determine the channel bit budget for different frames in each GOP in real-time, we train a GLM to estimate the predicted overhead bit proportion of every frame l, $\hat{w}_{ovh}^l$ in real time. The GLM is trained offline over a database of the factors that were discussed above and derived for videos with different types of motion and content. We use a database of 15 CIF video sequences that span (a) low motion: *Silent*, *Mother-Daughter*, *Bridge*, *Akiyo*, and *Container*; (b) medium motion: *Table Tennis*, *Coastguard*, *Tempete*, *Foreman*, and *Hall Monitor*; and (c) high motion: *Soccer*, *Bus*, *Football*, *Stefan*, and *Whale Show*. First three sequences from each motion category are used for training and the last two from each category for testing. For a given source encoding rate R_v, we compute the factors w_{ovh}^l, w_{cmse}^l, w_c^l for the frames of each training video sequence by using the DP-UEP(GOP) scheme and store them in the database along with the channel SNR. The GLM, explained in the next section, is trained offline only once. $\hat{w}_{ovh}^l$ is then used to estimate the channel bit budget constraint (as shown in (11) and estimate the optimal packet sizes and code rates for the slices of frame l.

5.2 GLM for estimating $\hat{w}_{ovh}^l$

GLMs are an extension of classical linear models [33]. Let $\mathbf{Y} = [y_1, y_2, y_3, ..., y_N]$ be a vector of our response variable w_{ovh}^l from the database. Every data point y_i in $\mathbf{Y}$ is expressed as a linear combination of a known covariate vector $\left[1, x_{i1}, x_{i2}, x_{i3}, ..., x_{ip}\right]$, where p is the number of factors, and a vector of unknown regression coefficients $\boldsymbol{\beta} =$

$\left[\gamma, \beta_1, \beta_2, ..., \beta_p\right]^T$. The covariate vector is a row of matrix $\mathbf{X}$ of order $N \times (p+1)$ with elements x_{ij} for N observations and p factors also from the database.

$$f(\mathbf{Y}) = \mathbf{X}\boldsymbol{\beta} \quad ; \quad f(y_i) = \gamma + \sum_{j=1}^{p} x_{ij}\beta_j. \tag{13}$$

where $f(.)$ is called the link function. After estimating $\boldsymbol{\beta}$, we use it to derive the predicted response variable vector $\widehat{\mathbf{Y}} = \left[\hat{y}_1, \hat{y}_2, \hat{y}_3, ..., \hat{y}_N\right]$ computed as $f^{-1}(\mathbf{X}\boldsymbol{\beta})$; f^{-1} is the inverse of the link function and $\widehat{\mathbf{Y}}$ is a vector of $\hat{w}^l_{ovh}$.

5.2.1 *Response variable distribution*

To determine the link function for the GLM, we need to know the distribution family of our response variable. We evaluate the goodness of fit for ranking Weibull, Gamma and Gaussian fitted distributions of w^l_{ovh} by using three information criteria (IC): (a) SIC: Schwarz information criterion, aka Bayesian information criterion [41], (b) AIC: Akaike information criterion [1], and (c) HQIC: Hannan-Quinn information criterion [16]. Each information criterion depends on the (i) number of distribution parameters to be estimated, and (ii) number of observations of our response variable w^l_{ovh}, and the maximized log-likelihood estimate of the fitted distribution producing the set of observations.

We randomly chose $m = 5000$ observations from the vector of w^l_{ovh} values in the database, obtained from all the training videos at channel SNRs from -2 dB to 6 dB. These are divided into 100 bins from zero to one and the likelihood function is maximized for each of the three fitted distributions. The distribution parameters where the likelihood is maximized are: (a) Gaussian: mean = 0.05, standard deviation = 0.095, (b) Gamma: shape parameter = 1, scale parameter = 0.05, and (c) Weibull: shape parameter = 1, scale parameter = 0.05. Since the shape parameter of both Gamma and Weibull distributions is 1, they are in essence exponential distributions. In Table 1, the goodness of fit of all three information criteria are minimum for Weibull and Gamma distributions; therefore our response variable is exponential. Figure 4 also shows that the cumulative distributions of Weibull and Gamma are the same and closer to the cumulative distribution of the 5000 observations than the Gaussian cumulative distribution.

5.2.2 *Model fitting and validation*

We use the statistical software R [42] for fitting our GLM and its validation. We classified our response variable as a member of the exponential family of distributions with identity as its link function. The GLM model in R uses the AIC index [1] to determine the order in which three factors, w^l_{cmse}, w^l_c and channel SNR are fitted. Here the AIC index is defined as $2 \times p - 2 \times max(L)$, where p is the number of factors and L is the log-likelihood estimate

Table 1 Goodness of fit statistics for maximized likelihood function

IC/Fitted Distribution	Weibull	Gamma	Gaussian
SIC	23.71	23.71	25.79
HQIC	12.72	12.72	16.88
AIC	6.25	6.25	8.33

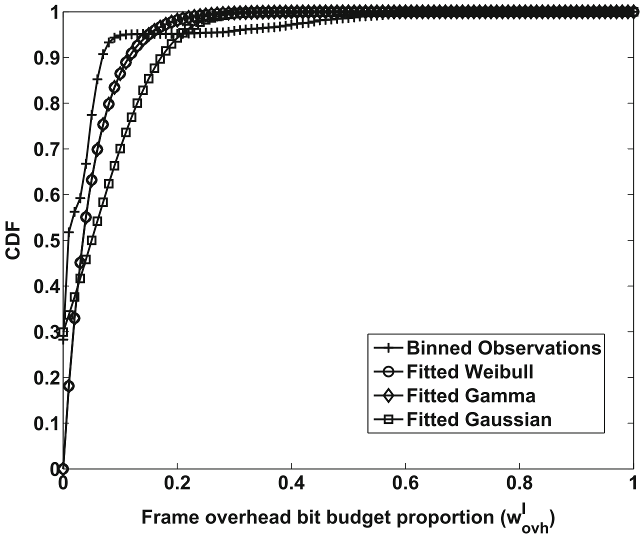


Fig. 4 Cumulative distribution function (CDF) for the binned observations and fitted distributions of w^l_{ovh}

for the model. We let $\mathbf{Y}^k$ represent the model with a subset of k factors (i.e. covariates). The i^{th} data point in $\mathbf{Y}^k$, y^k_i, where $i = 1, 2, ..., N$ is expressed as:

$$y^k_i = \gamma + \beta^k_1 x_{i1} + \beta^k_2 x_{i2} + ... + \beta^k_j x_{ij} + ... + \beta^k_k x_{ik}. \tag{14}$$

Here, γ is the intercept as considered in (13), $\beta^k_j = 1, 2, ..., k$ are the fitted coefficients for k factors, and x_{ij} represents the j^{th} factor value for the i^{th} observation in $\mathbf{Y}^k$. The forward stepwise approach is used to determine the order of our covariates [45]. The covariates and coefficients of our final model are shown in Table 2. We also introduced two interactions, w^l_c×channel SNR and w^l_{cmse}×channel SNR.

The goodness of fit for a GLM can be characterized by its deviance, which is a general term of variance [33]. A smaller deviance means a better model fit. Third column in Table 2 shows the reduction in deviance as each of the covariates in the first column is selected into the model. Model 1 is the best univariate model with w^l_c. Model 2 has both w^l_c and w^l_{cmse} covariates. In addition to these, Model 3 has channel SNR. Model 4 adds the first interaction between w^l_c and channel SNR, and Model 5 includes all the factors in Table 2.

Table 2 Final Model Factors and Coefficients

Covariate(Factor)	Coeff. for Final Model	Model Deviance
γ	-0.0193	167.18
w^l_c	1.3240	12.2
w^l_{cmse}	5.37×10^{-5}	11.7
channel SNR	0.0028	11.7
w^l_c× channel SNR	-0.0564	9.75
w^l_{cmse}× channel SNR	9.521×10^{-5}	9.7

5.2.3 *Normalized predicted CMSE* ($\hat{w}^l_{cmse}$)

Computing w^l_{cmse} for frame l is not feasible in real-time since it requires the decoding of the current and all other subsequent frames of the GOP which is computationally intensive and introduces about one GOP time delay. Therefore, we use the scheme proposed in [35] to predict the CMSE value of each slice i, $\hat{D}^l_p(i)$ in frame l. But, the predicted slice CMSE values of the future frames in the GOP will not be available during real-time transmission. We therefore use the sum of the predicted CMSE of all the slices of the previous GOP to compute the normalized predicted CMSE of the frame l, $\hat{w}^l_{cmse}$ in the current GOP using (12). It is reasonable to use the predicted CMSE of the previous GOP because for most GOPs there is a high correlation between the CMSE of adjacent GOPs. The predicted overhead bit budget for the frame l, $\hat{w}^l_{ovh}$ uses the $\hat{w}^l_{cmse}$ instead of w^l_{cmse} and other factors shown in Table 2. The predicted channel bit budget for the frame l is estimated as $\hat{w}^l_{ovh} \times \frac{R_{CH} L_G}{f_s}$. The proposed joint optimization in Section 4 is then used to compute the optimal packet sizes and RCPC code rates for the slices of frame l.

5.3 Computational complexity of DP-UEP(frame)

On a Core 2 Duo 2.6 GHz Intel processor with 4GB RAM, we observed that the average computation time across all test videos and channel SNR from -1 dB to 6 dB, is 75 ms for the IDR frame, 10.5 ms for the P frame, and 1.5 ms for the B frame. Since IDR frames have considerably more slices than P and B frames, and P frames have more slices than B frames, the computation time also varies accordingly. These low computational delays are acceptable in live streaming applications.

In this paper, we have used a GOP structure of IDR B P B P... with a length of 20 frames. Each GOP has one I-frame, nine P and 10 B-frames. The total time to process all the frames of a GOP in our scheme is $75 + 10.5 \times 9 + 1.5 \times 10 = 184.5ms$, which is much lower than 667 ms available to transmit these 20 frames at a frame rate of 30 frames/second. Also streaming applications buffer the frames at the decoder before starting the decoding process. This initial delay (known as the pre-roll delay) typically varies from 1-2s as mentioned in the literature [4, 10, 24, 48]. This process of smoothing the video transmission allows frame deadlines at the transmitter, which are way higher than the time required for the optimization process in our schemes.

6 Performance evaluation

6.1 Reference approaches

We compare our proposed DP-UEP(GOP) and DP-UEP(frame) schemes, with two reference schemes Dual15 [29], and EEP-slice-ENH. The Dual15 scheme is a representative of UEP schemes which use RCPC codes, and EEP-slice-ENH is a representative of the payload adaptation schemes. In the Dual15 scheme, every row of macroblocks in a H.264/AVC video frame is arranged in a slice. These slices typically have different sizes and CMSE values based on the spatial and temporal dependency of their macroblocks and frames. For example, the slice of an intra-coded frame typically has much larger size and higher CMSE value than that of a predicted frame. The Dual15 scheme treats every slice as a packet and does not carry out the payload adaptation. It finds the optimal RCPC code rates to protect the variable-sized packets based on their CMSE (i.e., using UEP), in order to minimize the expected received video distortion over an AWGN channel. It was shown in [29] that the Dual15 scheme outperforms other UEP schemes because it includes two additional options ‘not sent’ (i.e., using code rate of ∞) and ‘not coded’ (i.e., using code rate of $8/8$) in the RCPC set. Another interesting result in [29] was that the SortMSE scheme, which is derived from [18] and uses EEP with two additional options ‘not sent’ and ‘not coded’ in the RCPC set, also outperformed the UEP scheme which used the optimized RCPC codes without those two options in the set, for lower channel SNRs. The Dual15 thus represents the state-of-the-art in the class of UEP schemes.

The EEP-slice-ENH aggregates the pre-encoded slices for packet size adaptation by considering their priority, i.e., to form packets with more important ones having smaller sizes and error probabilities and also some least important packets being discarded (i.e., not sent) to meet the channel rate constraint. All packets in EEP-slice-ENH are equally protected with the best possible EEP code rate. This scheme is broadly similar to other packet (or payload) size adaptation schemes in literature [5, 18, 22, 30, 34]. The EEP-slice-ENH scheme, which incorporates the slice aggregation in SortMSE outperformed the latter.

The objective of EEP-slice-ENH is to minimize the expected received video distortion and this can be formulated in a manner similar to (4) [21]:

$$\begin{gathered}
\min_{r\,\epsilon\,\mathbf{R}}\left\{\sum_{i=1}^{n_p-n_{pd}}\left[1-(1-p_b(SNR,r))^{\left(\frac{h+S_p(i)}{r}\right)}\right]D_p(i)+K_1\right\}\\
=K_1+\min_{r\,\epsilon\,\mathbf{R}}\left\{\sum_{i=1}^{n_p-n_{pd}}\left[1-(1-p_b(SNR,r))^{\left(\frac{h+S_p(i)}{r}\right)}\right]D_p(i)\right\}\\
\text{subject to}\quad \sum_{i=1}^{n_p-n_{pd}}\frac{h+S_p(i)}{r}\le\left(\frac{R_{CH}L_G}{f_s}\right)
\end{gathered}\tag{15}$$

Constraint 2 in (4) is not valid here since r is no longer a vector. As in (4), K_1 is the permanent distortion caused by the discarded packets and is constant for a given value of n_{pd}. Apart from the change that only a single λ and r value needs to be determined, the same DP-based approach described in Section 4 is used to solve the optimization problem in (15).

6.2 Simulation setup

CIF (352 x 288) resolution (*i*) low motion sequences Akiyo and Container, (*ii*) medium motion sequences Foreman and Hall Monitor, and (*iii*) high motion sequences Stefan and Whale Show are used in our experiments. They are encoded using H.264/AVC JM reference software [14] at an encoding rate of 720 Kbps, frame rate 30 frames per second (fps) and transmitted over a 2 Mbps discrete-time AWGN channel. The GOP structure is IDR B P B P B, ..., P B, IDR with a length of 20 frames, and the slice size is 300 bytes. Error concealment, including both temporal concealment and spatial interpolation, is enabled for all the schemes evaluated in this paper. For this, the motion copy option provided in the JM [14] decoder is used. The error concealment in a frame depends on the frame type and the type of losses encountered. If an entire frame is lost, first the motion vectors and reference indices of the co-located macroblocks in the previously decoded reference frame are copied and motion compensation is used to reconstruct the lost frame based on the copied motion information [2]. If some slices of a predicted (P or B) frame are lost, the decoder verifies the availability of motion vector information for the lost macroblocks. If the motion vectors are available, the motion copy is performed else co-located macroblocks of the previous reference frame are directly copied. If some slices of an IDR frame are lost, the corresponding macroblocks are concealed using spatial interpolation.

The total network protocol header size is 54 bytes per packet as discussed in Section 3.2. The mother code of the RCPC code has rate $\frac{1}{4}$ with memory M=4 and puncturing period P=8. Log-likelihood ratio (LLR) is used in the Viterbi decoder. The initial RCPC rates available are {(8/9), (8/10), (8/12), (8/14), (8/16), (8/18), (8/20), (8/22), (8/24), (8/26), (8/28), (8/30), (8/32)}. Two additional rates, 8/8 corresponding to no coding and ∞ corresponding to discarding are also included. The performance evaluation of the schemes is based on a bit-level simulation of the compressed videos using the derived packet sizes and FEC code rates over 100 realizations of every AWGN channel SNR.

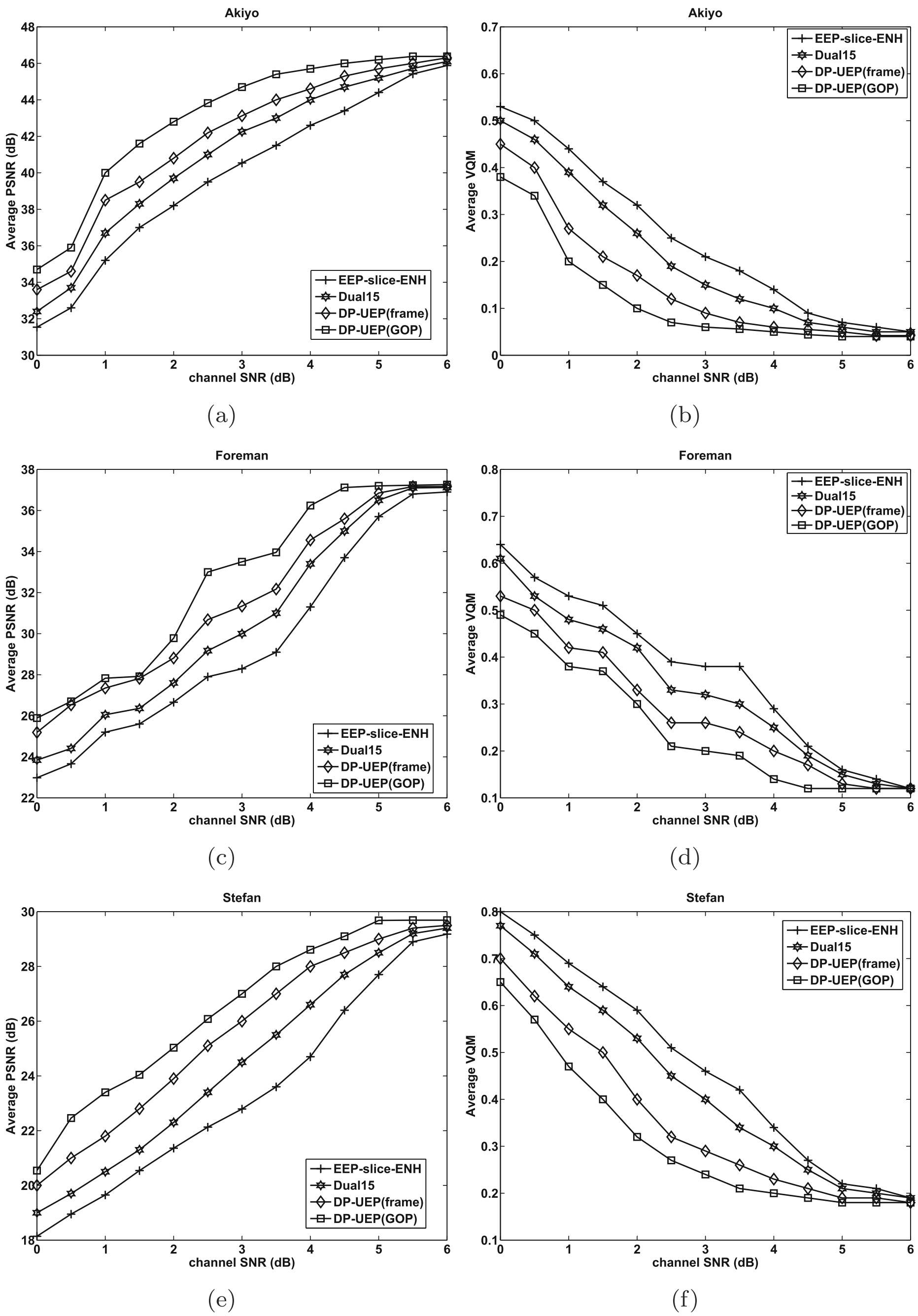
6.3 Performance of DP-UEP(GOP)

Figure 5 shows the video quality performance of DP-UEP(GOP), DP-UEP(frame), Dual15, and EEP-slice-ENH schemes in terms of PSNR and a perceptually based Video Quality Metric (VQM). VQM is reported as a single number for the entire sequence and has a nominal output range from zero to one, where one represents the worst quality [36]. The error-free PSNR values of Akiyo, Foreman, and Stefan compressed at 30 fps and 720 Kbps are 46.5 dB, 37.3 dB, and 29.7 dB, respectively. We observe that as the channel SNR increases to 6 dB, all the schemes are able to achieve the error-free PSNR values of the individual sequences. This is because the channel errors are few and with motion concealment at the decoder, the best possible video PSNR value is achievable.

EEP-slice-ENH does not perform as good as other schemes in Fig. 5. Though it adapts the packet size to the video priority by aggregating the slices and also discarding lower priority packets, it is still limited to providing equal protection to all the packets formed. The lowest and highest optimal EEP code rates derived across GOP's were $\left[\frac{8}{20}\ \frac{8}{14}\right]$.

Dual15 does not consider packet size adaptation and only performs optimal (UEP) RCPC code rate allocation to the slices (considered as individual packets) of each GOP, also discarding some least important slices [29]. Our proposed DP-UEP(GOP) takes advantage of both the priority-adaptive packet sizes and optimal RCPC packet code rate allocation. It assigns optimal code rates as low as $\frac{8}{32}$ to the high priority packets with small packet sizes and higher code rates to the lower priority packets with larger packet sizes within every GOP. However, all packet sizes are restricted by the network MTU size of 1500 bytes. At a channel SNR of 3 dB for Foreman, EEP-slice-ENH, Dual15 and DP-UEP(GOP) achieve average PSNR values of 28.3 dB, 30 dB, and 33.5 dB, respectively and average VQM values of 0.38, 0.32, and 0.2, respectively. DP-UEP(GOP) achieves maximum PSNR gains of 3.3 dB for Akiyo, 3.5 dB for Foreman, and 2.9 dB for Stefan over Dual15 at channel SNRs of 2 dB, 3 dB, and 2 dB, respectively. DP-UEP(GOP) also achieves maximum gains of 4.6 dB for Akiyo, 5.2 dB for Foreman, and 4.4 dB for Stefan over the EEP-slice-ENH scheme at channel SNRs of 3 dB, 2 dB, and 3.5 dB. Similar performance was observed for Container, Hall Monitor, and Whale Show sequences.

The considerable improvement in video quality achieved by our DP-UEP(GOP) can be explained by the following two factors: (*i*) the lower number of slices discarded per GOP shown in Fig. 6, and (*ii*) the composition of the final transmitted bits in terms of the compressed source bits, network protocol headers, and FEC bits shown in Fig. 7. Balancing the overhead due to the FEC parity bits allows Dual15 to discard fewer slices per GOP as compared to EEP-slice-ENH. DP-UEP(GOP) further reduces the number of discarded slices as compared to Dual15 by balancing both the overhead due to FEC parity bits as well as the network protocol headers attached to the packets formed by aggregating slices. For example in Fig. 6, at a channel SNR of 3 dB, DP-UEP(GOP) does not discard any slices whereas



(a) (b)

(c) (d)

(e) (f)

Fig. 5 Average Video PSNR (dB) and average VQM comparison computed over 100 realizations of each AWGN channel for Akiyo: (**a**),(**b**), Foreman: (**c**),(**d**) and Stefan: (**e**), (**f**). The error-free PSNR values are: 46.5 dB for Akiyo, 37.3 for Foreman and 29.7 for Stefan

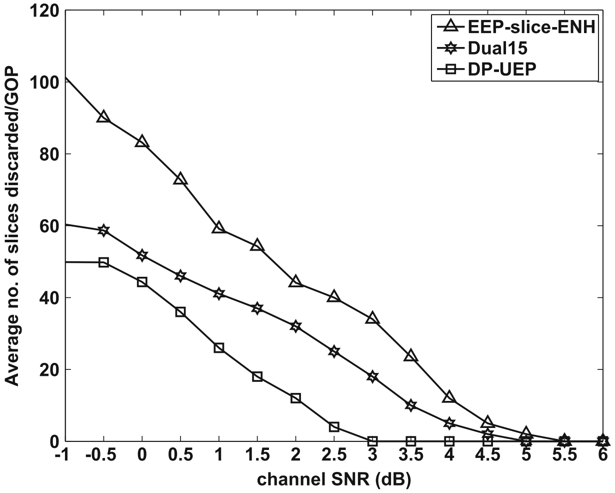


Fig. 6 Average number of slices discarded per GOP in EEP-slice-ENH, Dual15 and **DP-UEP** for Foreman

20 and 35 slices are discarded per GOP by Dual15 and EEP-slice-ENH, respectively. This shows that though we encode the video at a target bit rate of 720 Kbps, every scheme adjusts this bit rate by discarding some low-priority slices in order to minimize the expected received video distortion under the given channel SNR condition and bit budget constraints.

Figure 7 shows the bit contribution of the source, network protocol headers, and FEC to the total bits transmitted over a 2 Mbps channel at 3 dB channel SNR for Foreman. DP-UEP(GOP) transmits more source bits (i.e., a relatively higher bit rate) than the other two schemes by reducing the network protocol overhead as well as allocating optimal RCPC code rates based on packet priority. It also uses only 5.5 % bits for the network protocol overhead, compared to 8.5 % and 11.5 % overhead bits for EEP-slice-ENH and Dual15, respectively. Further 61.3 % bits are allocated for FEC overhead in DP-UEP(GOP) com-

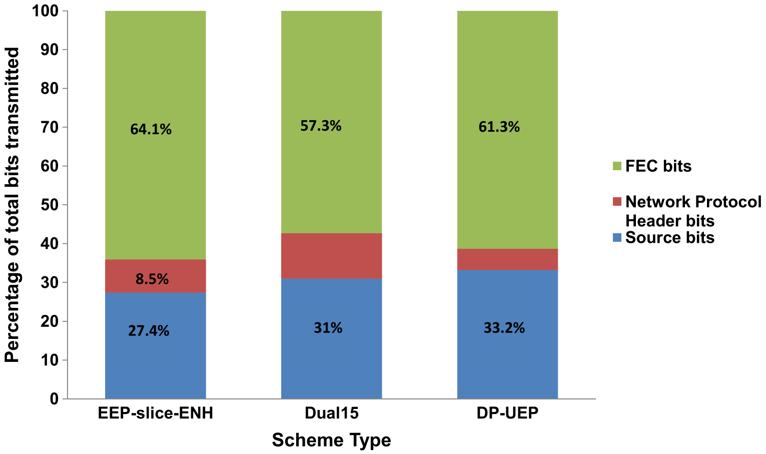


Fig. 7 Distribution of the final output bits for Foreman at 3 dB channel SNR in EEP-slice-ENH, Dual15, and **DP-UEP** schemes

pared to 57.3 % in Dual15, thus providing better FEC protection. Although EEP-slice-ENH uses 64.1 % FEC bits, it uses EEP which ignores packet priority. The DP-UEP(GOP) scheme sends the highest percentage of source bits (i.e., 33.2 %) which also correlates to no slices being discarded at 3 dB channel SNR, shown earlier in Fig. 6. A similar trend is also observed for other sequences and at other channel SNRs.

6.4 Performance of DP-UEP(frame)

DP-UEP(GOP), Dual15, and EEP-slice-ENH schemes use the measured CMSE values and perform optimization over the slices of the entire GOP. On the other hand, DP-UEP(frame) uses the predicted CMSE and frame overhead bit budget values, and the joint packet size and code rate optimization is carried out over slices of each frame. Thus, DP-UEP(frame) enables real-time packet formation and transmission of videos for live streaming applications which is not possible with the other three schemes. The DP-UEP(frame) still achieves considerable PSNR and VQM gains over Dual15 and EEP-slice-ENH schemes. For example, the maximum PSNR gains achieved by DP-UEP(frame) over Dual15 are 1.8 dB for Akiyo at 1 dB channel SNR, 2.1 dB for Foreman at 1 dB channel SNR, and 1.7 dB for Stefan at channel SNR of 2.5 dB. Similar trends are also observed in the VQM performance of the three test videos in Fig. 5. Further, simulation results of Whale show, Hall Monitor, and Container also showed trends similar to those in Fig. 5. The performance of DP-UEP(frame) is, however, still lower than DP-UEP(GOP) because the latter uses the measured values of CMSE and performs optimization over all slices of a GOP. So, optimization over slices of each frame in DP-UEP(frame) is suboptimal compared to the DP-UEP(GOP) scheme.

7 Conclusion

An efficient joint optimization algorithm for packet formation and optimal RCPC code rate allocation was proposed to improve the quality of H.264/AVC bitstreams transmitted over noisy channels. The proposed algorithm used a cross-layer information exchange between the APP, MAC, and PHY layers. A dynamic programming approach (DP-UEP(GOP)) was used where packets were formed through slice aggregation and the optimal RCPC packet code rates were determined recursively over a GOP. The options of not coding or discarding some less important packets were exploited to reduce the expected received video distortion by increasing protection to more important packets. The dynamic programming approach was also extended to work on each video frame (DP-UEP(frame)) instead of the entire GOP. It has very low computational complexity and can be used in live streaming applications. The frame bit budget prediction used a GLM model developed using three factors - normalized compressed frame bit budget, normalized frame CMSE and channel SNR over a database of videos. Both proposed schemes outperformed contemporary state-of-the-art schemes, providing significantly better video quality for different video sequences. Our proposed schemes can work well with current wireless network standards such as IEEE 802.11n with MTU packet size restrictions.

References

1. Akaike H (1974) A new look at the statistical model identification. IEEE Trans Autom Control AC-19(6):716

2. Bandyopadhyay SK, Wu Z, Pandit P, Boyce JM (2006) An error concealment scheme for entire frame losses. IEEE Sarnoff Symp:1–4
3. Chakarakesi J, Chou PA (2004) Application layer Error-Correction coding for rate-distortion optimized streaming to wireless clients. IEEE Trans Commun 52(10):1657
4. Chen Z, Reznik Y (2012) Analysis of video codec buffer and delay under time-varying channel. In: IEEE VCIP. pp 1–6
5. Choi BY, Song S, Wang Y, Park EK (2009) Using RTT variability for adaptive cross-layer approach to multimedia delivery in heterogeneous networks. IEEE Trans Multimedia 11(6):1194
6. Connie AT, Nasiopoulos P, Leung VCM, Fallah YP (2008) Video packetization techniques for enhancing H.264 video transmission over 3G networks. IEEE CCNC:800–804
7. Choudhury S, Gibson JD (2007) Payload length and rate adaptation for multimedia communications in wireless LANs. IEEE J Sel Areas Commun 25(4):796
8. Choudhury S, Gibson JD (2008) Throughput optimization for wireless LANs in the presence of packet error constraints. IEEE Commun Lett 12(1):11
9. Chou PA, Miao Z (2006) Rate-distortion optimized streaming of packetized media. IEEE Trans Multimedia 8(2):390
10. Duffield NG, Ramakrishnan KK, Reibman AR (1998) SAVE: an algorithm for smoothed adaptive video over explicit rate networks. IEEE INFOCOM:1093–1102
11. Fallah YP, Darrell K, Avideh S, Faizal K, Nasiopoulos P (2007) A cross layer optimization mechanism to improve H.264 video transmission over WLANs. IEEE CCNC:875–879
12. Fallah YP, Nasiopoulos P, Alnuweiri H (2008) Efficient transmission of H.264 video over multirate IEEE 802.11e WLANs. EURASIP J Wirel Commun Netw 2008:1
13. Garibotto G, Murrieri P, Capra A, Muro SD, Petillo U, Flammini F, Esposito M, Pragliola C, Leo GD, Lengu R, Mazzino N, Paolillo A, Urso MD, Vertucci R, Narducci F, Ricciardi S, Casanova A, Fenu G, Mizio MD, Savastano M, Capua MD, Ferone A (2013) White paper on industrial applications of computer vision and pattern recognition. Image Analysis and Processing ICIAP 8157:721–730
14. H.264/AVC reference software jm18.5. http://iphome.hhi.de/suehring/tml/download/
15. Hagenauer J (1988) Rate-compatible punctured convolutional codes (RCPC Codes) and their Applications. IEEE Trans Commun 36(4):389
16. Hannan EJ, Quinn BG (1979) The determination of the order of an autoregression. J Royal Stat Soc B 41(2):190
17. Horne U, Stuhlmuller K, Link M, Girod B (1999) Robust internet video transmission based on scalable coding and unequal error protection. J Image Commun Special issue on Real-Time Video over the Internet 15(1-2):77
18. Huang YZ, Apostolopoulos JG (2007) A joint packet selection/omission and FEC system for streaming video. In: IEEE ICASSP. Honolulu, pp I–845 – I–848
19. Jelenkovic PR, Tian J (2008) Dynamic packet fragmentation for wireless channels with failures. In: ACM MobiHoc, pp 73–82
20. Kambhatla KKR, Kumar S, Paluri S, Cosman PC (2012) Wireless H.264 video quality enhancement through optimal prioritized packet fragmentation. IEEE Trans Multimedia 14(5):1480
21. Kambhatla KKR, Kumar S, Cosman PC (2012) H.264/AVC video packet aggregation and unequal error protection for noisy channels. In: IEEE international conference on image processing, ICIP. Orlando, pp 1649–1652
22. Korhonen J, Wang Y (2005) Effect of packet size on loss rate and delay in wireless links. In: IEEE WCNC, vol 3, pp 1608–1613
23. Ksentini A, Naimi M (2006) Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross layer architecture. IEEE Commun Mag 44(1):107
24. Lakshman ARR, Tabatabai A (1992) A scheme for smoothing delay sensitive traffic offered to ATM networks. In: IEEE INFOCOM. pp 776–785
25. Lee CW, Yang CS, Su YC (2009) Low-complexity adaptive packet size assignment schemes for real-time scalable video transmission over WLANs. In: ACM IWCMC. Leipzig, pp 1062–1066
26. Lee CW, Yang CS, Su YC (2006) Adaptive UEP and packet size assignment for scalable video transmission over burst-error channels. EURASIP J App Signal Process 2006:1
27. Li F, Liu G (2009) Compressed-domain-based transmission distortion modeling for precoded H.264/AVC video. IEEE Trans Circuits Syst Video Technol 19(12):1908
28. Li D, Sun X (2006) Nonlinear Integer Programming (International Series in Operations Research and Management Science, Springer. USA
29. Lin TL, Cosman PC (2010) Efficient optimal RCPC code rate allocation with packet discarding for pre-encoded compressed video. IEEE Signal Process Lett 17(5):505

30. Lin H, Wu TY, Huang CY (2011) Payload length adaptation for wireless video transmission in multicarrier systems. In: IEEE VTC. San Francisco, pp 1–5
31. Local and Metropolitan Area Networks-Specific Requirements Part 11 (2007) Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications
32. Luo H, Ci S, Wu D (2011) A cross-layer design for the performance improvement of real-time video transmission of secondary users over cognitive radio networks. IEEE Trans Circuits Syst Video Technol 21(8):1040
33. McCullagh P, Nelder JA (1989) Generalized Linear Models, 2nd edn. Monographs on Statistics and Applied Probability (Chapman & Hall/CRC)
34. Modiano E (1999) An adaptive algorithm for optimizing the packet size used in wireless ARQ. Wirel Netw J 5(4):279
35. Paluri S, Kambhatla KKR, Kumar S, Bailey B, Cosman P (2012) Predicting slice loss distortion in H.264/AVC video for low complexity data prioritization. In: IEEE ICIP. Orlando, pp 689–692
36. Pinson MH, Wolf S (2004) A new standardization method for objectively measuring video quality. IEEE Trans Broadcast 50(3):312
37. Rohc (2001) RFC:3095, RoHC: Robust Header Compression
38. Schier M, Welzl M (2012) Optimizing selective ARQ for H.264 live streaming: a novel method for predicting loss-impact in real time. IEEE Trans Multimedia 14(2):415
39. Shih CH (2012) Enhancing packet-level forward error correction for streaming video in wireless networks. Int J Comput Sci Issues (IJCSI) 9(5):146
40. Shih CH (2010) Adaptive forward error correction combined with packet size control for wireless video. In: IEEE Sixth IIH-MSP. Darmstadt, pp 256–259
41. Schwarz G (1978) Estimating the dimension of a model. Ann Stat 6(2):461
42. The R Project for Statistical Computing. http://www.r-project.org/
43. Tsai MF, Chilamkurti N, Shieh CK (2011) An adaptive packet and block length forward error correction for video streaming over wireless networks. Springer Wireless Pers Commun 56(3):435
44. van der Schaar M, Shankar NS (2005) Cross-layer wireless multimedia transmission: challenges, principles and new paradigms. IEEE Wireless Commun Mag 12(4):50
45. Venables WN, Ripley BD (2002) Modern Applied Statistics with S., 4th edn. Springer
46. Vosoughi A, Testoni V, Cosman P, Milstein L (2013) Joint source-channel coding of 3D video using multiview coding. In: IEEE ICASSP. pp 2050–2054
47. Wiegand T, Farber N, Stuhlmuller KS, Girod B (2000) Error-resilient video transmission using long-term memory motion-compensated prediction. IEEE J Sel Areas Commun 18(6):1050
48. Wiegand T, Sullivan GJ, Bjontegaard G, Luthra A (2003) Overview of the H.264/AVC video coding standard. Trans IEEE Circuits Syst Video Technol 13(7):560
49. Wu TY, Chuang TT, Huang CY (2011) Optimal transmission of high definition video transmission in WiMedia systems. Wirel Netw J 17(2):291
50. Yoo T, Lavery RJ, Goldsmith A, Goodman DJ (2006) Throughput optimization using adaptive techniques. IEEE Commun Lett:1–7

Dr. Kashyap K. R. Kambhatla received his B.Tech. degree in Electronics and Communication Engineering from Jawaharlal Nehru Technological University, Hyderabad, India in 2004 and the M.S. degree in Electrical and Computer Engineering from Clarkson University, Potsdam, NY, in 2006. He received his Ph.D. with honors in Electrical and Computer Engineering jointly awarded by the Jacobs School of Engineering, University of California San Diego and the College of Engineering, San Diego State University. During the summer of 2006, he worked on video telephony in 3G EV-DO Rev. A and B cellular networks as a research intern in the Next-Gen Wireless Technology Group at Sprint Advanced Technology Labs, Burlingame, CA. His research interest is in the area of real-time cross-layer quality-ofservice (QoS) aware video streaming over ad-hoc wireless communication networks through packet size and physical layer FEC code rate adaptation. He also works on network optimization problems and developing efficient medium access control protocols for reliable video transmission. He is also a member of Phi Kappa Phi and Scholars without Borders.

Seethal Paluri received her B.S. in Electrical and Computer Engineering from University of the West Indies, Trinidad and Tobago in 2005 and her M.S. degree in Electrical Engineering from University of Southern California, Los Angeles in 2008. She is Ph.D. candidate in Computational Sciences with major in Mathematics and minor in Electrical Engineering, jointly awarded by San Diego State University and Claremont Graduate University. Her research interests are in the areas of error resilient image and video compression, error concealment techniques, and region of interest (ROI) based coding. She also works on developing statistical modeling techniques for video quality prediction and prioritization for reliable real-time Quality of Service (QoS) aware streaming over wireless networks. She is also a member of Scholars without Borders.

Dr. John D. Matyjas received his Ph.D. in electrical engineering from State University of New York at Buffalo in 2004. Currently, he is serving as the Connectivity & Dissemination Core Technical Competency Lead at the Air Force Research Laboratory (AFRL) in Rome, NY. His research interests include dynamic multiple-access communications and networking, spectrum mutability, statistical signal processing and optimization, and neural networks. He serves on the IEEE Transactions on Wireless Communications Editorial Advisory Board. He is also the recipient of the 2012 IEEE R1 Technology Innovation Award, 2012 AFRL Harry Davis Award for "Excellence in Basic Research," and the 2010 IEEE Int'l Communications Conf. Best Paper Award. He is an IEEE Senior Member, chair of the IEEE Mohawk Valley Signal Processing Society, and member of Tau Beta Pi and Eta Kappa Nu.

Dr. Sunil Kumar obtained his B.E. in Electronics Engineering from S.V. National Institute of Technology in 1988, and his M.E. and Ph.D. in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, India, in 1992 and 1997, respectively. From 1997-2002, he was a postdoctoral fellow and adjunct faculty in Electrical Engineering Systems department at the University of Southern California, Los Angeles, CA. From 2002 to 2006, he was an Assistant Professor in Electrical and Computer Engineering at Clarkson University, Potsdam, NY. Since 2006, he is an Associate Professor and Thomas G. Pine Faculty Fellow in the Electrical and Computer Engineering Department at the San Diego State University, San Diego, CA. His research interests include QoS-aware cross layer wireless network protocols and robust multimedia compression techniques, including H.264. He has published more than 100 research articles in refereed journals and conferences, including two book/book chapters. His research has been supported by NSF, U.S. Air Force Research Lab, Dept. of Energy, California Energy Commission, and industry. He was a guest editor of the April 2006 special issue of Elsevier Journal of Visual Communication and Image Representation on Emerging H.264/AVC Video Coding Standard". He has organized workshops and technical sessions in several international conferences.