H.264/AVC Video Packet Aggregation and Unequal Error Protection for Noisy Channels

Kashyap K. R. Kambhatla[⋆,†], Sunil Kumar[⋆], Pamela Cosman[†] and John Matyjas[§]

[⋆] Electrical and Computer Engineering, San Diego State University
[†] Electrical and Computer Engineering, University of California, San Diego
[§] Air Force Research Laboratory, Rome, NY
E-mail:kkambhat@ucsd.edu, skumar@mail.sdsu.edu, pcosman@eng.ucsd.edu, john.matyjas@rl.af.mil

***Abstract*—The quality of H.264/AVC compressed video delivery over wireless channels is affected by packet losses. Aggregating H.264/AVC slices to form video packets with sizes adaptive to their importance can improve transmission reliability. Larger packets are more likely to be in error but smaller packets cause more overhead. A second method is assigning stronger channel code rates to more important slices. We use cross-layer dynamic programming to address both adaptive packet formation as well as RCPC-channel code rate allocation simultaneously, to improve received video quality. Simulation results show the advantages of the proposed scheme.**

***Index Terms*—H.264/AVC video compression, adaptive packet formation, RCPC codes, dynamic programming, unequal error protection.**

I. Introduction

Compressed video packet communication over wireless channels is influenced by errors due to fading, interference and noise. Lost video packets induce different levels of quality degradation due to temporal and spatial dependencies. To improve performance packet formation and unequal error protection (UEP) are employed at the medium access control (MAC) and physical (PHY) layers. Video slices of a H.264 compressed bitstream can be aggregated into packets at the application (APP) layer for transmission over wireless networks with a maximum transmission unit (MTU) size requirement [1]. These video packets can be decoded independently from one another at the receiver.

Adapting packet sizes to the channel error characteristics improves the successful packet transmission probability and involves a trade-off between reducing the overhead by adopting large sizes and reducing the transmission error rate by using small sizes. Maximizing throughput does not guarantee minimum received video distortion since lost video packets induce different amounts of distortion. Hence video packet sizes should be adaptive to the packet importance.

In this paper, our objective is minimizing the expected received video distortion by jointly optimizing the packet sizes at the APP layer and forward error correction (FEC) code rates allocated at the PHY layer. Past research proposed optimizing joint source channel coding of the video transmission at the APP layer [2], [3]. FEC at the APP layer provides more flexibility and avoids extra header bits for source significance information (SSI). However protection strategies have to be derived from channel parameters like fading depth, noise level variation, short term signal loss, or jammer activity. A joint APP-MAC-PHY cross layer interface [4] is desirable which allows the APP layer to assign both optimal packet sizes as well as estimate the optimal unequal FEC code rates using the channel state information (CSI), channel bit rate constraints and network packet size limitation.

In [5], code rate allocation with packet discarding at the APP layer was studied for H.264 video packets in a group of pictures (GOP). We extend the work in [5] by considering the network protocol headers and developing a novel joint optimization scheme to select optimal packet sizes (instead of a fixed packet size as in [5]) and efficient UEP code rates for all the slices of one GOP at a time.

This paper is organized as follows. Section II gives an overview of the proposed cross-layer approach. Section III discusses the dynamic programming approach for solving the joint optimization problem. Packet formation is discussed in Section III-A while optimal packet code rate allocation is discussed in Section III-B. Section IV discusses the problem formulation for other equal error protection (EEP) schemes. Simulation results are presented in Section V. Section VI concludes the paper.

II. Proposed Cross-Layer Approach

At the transmitter, the APP layer carries out two functions: cumulative mean squared error (CMSE) computation for slice prioritization, and optimal packet formation for pre-encoded H.264 video slices as discussed in Section III. The distortion contributed by a slice loss is computed in the encoder as the CMSE due to the loss of that slice, taking into consideration the error propagation within the entire GOP. The MTU size, RTP/UDP, IP and MAC layer headers which remain unchanged for a given network and the CSI and FEC configuration information from the PHY layer are sent to the optimal packet formation block. The optimal packet formation block in the APP layer uses this information to form variable-sized packets from pre-encoded slices ordered from highest to lowest CMSE contribution and estimate their corresponding optimal FEC code rates that can be applied at the PHY layer by using a dynamic programming approach proposed in Section III. The FEC configuration contains a mother code rate and a family of punctured rates which helps the APP layer to evaluate the error probability for the resulting packet sizes. The wireless network MTU size is 1500 bytes when using the IEEE 802.11 protocol [6]. Each packet at the APP layer is appended with RTP/UDP/IP overhead of 4 bytes after RObust Header Compression (RoHC) and 50 bytes of MAC and PHY layer headers.

III. Expected Video Distortion Minimization

We introduce a dynamic programming approach which requires *a priori* channel and video information: channel transmission bit rate of R_{CH} bits per second, video frame rate of f_s frames per second and total outgoing bit budget for a GOP of length L_G frames equal to $\frac{R_{CH} L_G}{f_s}$. We use n_s to denote the number of slices generated per GOP which is a constant. We use n_p to denote the number of packets formed from these slices; n_p is a variable. $S_p(i)$ is the i^{th} packet size before adding network headers of size h bits and parity bits from a punctured RCPC code rate selected from a candidate set, $\mathbf{R} = \{R_1, R_2, R_3, ..., R_K\}$. The number of packets discarded is n_{pd} which will be described in Section III-B.

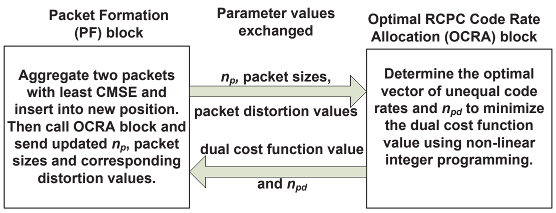

Fig. 1: Block diagram of proposed dynamic programming approach.

A. Packet Formation

The proposed scheme is a recursive process between two blocks shown in Figure 1. It 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 distortion contribution. The OCRA block determines the optimal RCPC packet code rates and the number of packets discarded n_{pd}, to minimize a dual cost function value which will be described in Section III-C. It forwards the computed parameters to the PF block as shown in Figure 1.

The PF block, considering the remaining packets not discarded by the OCRA block, aggregates the two packets with least CMSE contribution and inserts the aggregated packet into a new position based on its distortion computed as the sum of the CMSE values of both packets. This maintains the decreasing order of packet distortion. It calls the OCRA block again to determine optimal RCPC code rates for the new set of packets. The blocks in Figure 1 exchange the parameters recursively until aggregating packets is no longer beneficial to reduce the dual cost function value. As an example, Figure 2 shows one iteration of our proposed scheme in the PF block. After returning from the OCRA block, the number of packets is updated to $n_p = n_s - n_{pd}$ since n_{pd} packets were dropped in the OCRA block. The two least important packets are then aggregated and inserted into a new position while the remaining packets are simply retained. $n_p - 1$ packets with their sizes and distortion values are once again sent to the OCRA block, to estimate their new optimal packet code rates. The aggregated packet is at position $n_p - j$. The size of the aggregated packets is constrained by the MTU size. Aggregating packets reduces the total overhead from headers; the saved bits are used to increase the FEC protection to more important packets.

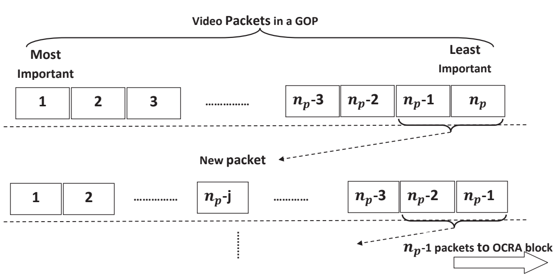

Fig. 2: Packet formation in PF block.

B. Distortion minimization over RCPC code rates: OCRA block

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 distortion due to channel packet loss and distortion from packets discarded at sender as in [5]. The distortion due to the compression is neglected in this formulation because the slices are pre-encoded and assumed to be at relatively high quality, so compression distortion is small compared to distortion from losses and discards.

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

$D_p(i)$ is the distortion caused due to loss of packet i and is computed as the sum of the CMSE of individual slices contained in the packet. $p_{pkt}(i)$ is the packet error probability and depends on the channel SNR, packet size and the selected RCPC code rate. For a given value of n_{pd}, the distortion due to the discarded packets in Equation 1 is a constant K_1. After appending packet i_{th} packet with h bits of network header followed by parity bits from code rate r_i the optimization problem for minimizing expected video distortion over the GOP is formulated as:

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

$p_b(SNR, r_i)$ is the bit error probability after channel decoding for code rate r_i. Constraint 1 in Eq. 2 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 Eq. 2 to an unconstrained problem [7]. 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 $\lambda_i \,\epsilon\, \mathbb{R}^+$, we construct the Lagrangian cost function as:

$$\begin{aligned}
&F_{GOP}(\mathbf{r}, \boldsymbol{\lambda}) \\
&= \sum_{i=1}^{n_p - n_{pd}} \left[1 - (1 - p_b(SNR, r_i))^{\left(\frac{h + S_p(i)}{r_i}\right)} \right] D_p(i) \\
&\quad + \lambda_1 \left(\sum_{i=1}^{n_p - n_{pd}} \frac{h + S_p(i)}{r_i} - \frac{R_{CH} L_G}{f_s} \right) + K_1 \\
&\quad + \sum_{i=2}^{n_p - n_{pd}} \lambda_i (r_{i-1} - r_i) \\
&\text{where} \quad \boldsymbol{\lambda} = \left[\lambda_1, \lambda_2, ..., \lambda_{n_p - n_{pd}} \right]
\end{aligned} \qquad (3)$$

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 which will be discussed in Section III-C. 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:

$$\begin{aligned}
&d_{GOP}(\boldsymbol{\lambda}) = \min_{\mathbf{r} \epsilon \mathcal{C}} \; F_{GOP}(\mathbf{r}, \boldsymbol{\lambda}) \\
&= \min_{\mathbf{r} \epsilon \mathcal{C}} \sum_{i=1}^{n_p - n_{pd}} \left\{ p_{pkt}(i) D_p(i) + \lambda_1 \left(\frac{h + S_p(i)}{r_i} \right) \right\} \\
&+ \sum_{i=2}^{n_p - n_{pd}} \lambda_i (r_{i-1} - r_i) + K_2
\end{aligned} \qquad (4)$$

$K_2 = K_1 - \lambda_1 \left(\frac{R_{CH} L_G}{f_s} \right)$ in Eq. 4 is a constant and the computation of $d_{GOP}(\boldsymbol{\lambda})$ can be further simplified as follows. Let $A(r_i) = D_p(i) \left(1 - (1 - p_b(SNR, r_i))^{\left(\frac{h + S_p(i)}{r_i}\right)} \right) + \lambda_1 \left(\frac{h + S_p(i)}{r_i} \right)$ and after algebraic manipulation the dual function can now be expressed as:

$$\begin{aligned}
&d_{GOP}(\boldsymbol{\lambda}) \\
&= K_2 + \min_{r_1 \,\epsilon\, \mathbf{R}} \{A(r_1) + \lambda_2(r_1)\} \\
&\quad + \sum_{i=2}^{n_p-n_{pd}-1} \min_{r_i \,\epsilon\, \mathbf{R}} \{A(r_i) + r_i(\lambda_{i+1} - \lambda_i)\} \\
&\quad + \min_{r_{n_p-n_{pd}} \,\epsilon\, \mathbf{R}} \left\{A(r_{n_p-n_{pd}}) - \lambda_{n_p-n_{pd}}(r_{n_p-n_{pd}})\right\} \\
&= K_2 + \sum_{i=1}^{n_p-n_{pd}} \min_{r_i \,\epsilon\, \mathbf{R}} \tilde{F}_{GOP,i}(r_i, \lambda_i)
\end{aligned} \quad (5)$$

where

$$\begin{aligned}
&\tilde{F}_{GOP,i}(r_i, \lambda_i) \\
&= \begin{cases} A(r_1) + \lambda_2(r_1)\ ,\ i = 1 \\ A(r_i) + r_i(\lambda_{i+1} - \lambda_i)\ ,\ i = 2, 3, ..., n_p - n_{pd} - 1 \\ A(r_{n_p-n_{pd}}) - \lambda_{n_p-n_{pd}}(r_{n_p-n_{pd}})\ ,\ i = n_p - n_{pd} \end{cases}
\end{aligned}$$

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 [5], [7].

C. Determination of $\boldsymbol{\lambda}$

We use the subgradient method [7] 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 [7]. 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, it may not be differentiable at all points. Nevertheless, subgradients can still be found and are used to compute the optimal value [7]. 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:

$$\lambda_i^{(k+1)} = \max(0, \lambda_i^k + s_k \xi_i^k / \|\boldsymbol{\xi}^k\|) \quad (6)$$

where s_k is the step size. Based on the derivation in [7], the subgradients $\boldsymbol{\xi}^k$ of $d_{GOP}(\boldsymbol{\lambda})$ at $\boldsymbol{\lambda}^k$ are

$$\begin{aligned}
&\xi_1^k = g(\mathbf{r}^k) - \frac{R_{CH} L_G}{f_s} = \sum_{i=1}^{n_p-n_{pd}} \left(\frac{h+S_p(i)}{r_i}\right) - \frac{R_{CH} L_G}{f_s} \\
&\xi_i^k = r_{i-1} - r_i \text{ for } i = 2, 3, 4, ..., n_p - n_{pd}
\end{aligned} \quad (7)$$

where $g(.)$ is the rate constraint function of the problem and $\mathbf{r}^k = [r_1^k, r_2^k, ..., r_{n_p-n_{pd}}^k]$ is the solution to $\min_{\mathbf{r} \,\epsilon\, \mathcal{C}}\ F_{GOP}(\mathbf{r}, \boldsymbol{\lambda}^k)$ in Equation 4 .

D. Discarding Packets

To allow discarding of less important packets or sending them unprotected, the candidate set of punctured code rates $\mathbf{R}$ is modified to $\{1, R_1, R_2, R_3, ..., R_K, \infty\}$. If $r_i = \infty$, then packet i is discarded. The induced distortion is accounted for in the overall expected distortion $E[\tilde{D}_{GOP}]$ through component K_1 in Equation 2. If $r_i = 1$, the video packet is transmitted uncoded over the channel.

IV. Problem Formulation of EEP schemes

We compare our proposed scheme denoted as **DYN-PROG-UEP** with the **Dual15** scheme proposed in [5] and two EEP schemes, namely **EEP-slice** and **EEP-slice-ENH**. **EEP-slice** treats every slice as a packet and does not discard any of them at the sender, i.e. $n_{pd} = 0$. Every slice in the GOP is attached with network protocol headers and we use the single strongest FEC code rate allowed for all the slices within the bit rate constraint. Though the final bit rate after adding header and parity bits does not exceed the bit budget, there is a possibility that not all of the available bits are utilized due to $\mathbf{R}$ being a small discrete set of punctured code rates. To be fair, we limit the bit budget of all other schemes to the number of bits used by **EEP-slice**.

The second scheme **EEP-slice-ENH** is similar to **DYN-PROG-UEP** in the way pre-encoded slices are aggregated to form packets with more important ones having smaller sizes and error probabilities and also the less important packets being discarded if necessary. However all packets are equally protected with the same code rate. The objective in this scheme is minimizing the expected received video distortion and can be formulated in a manner similar to Eq. 2:

$$\begin{aligned}
&\min_{r \,\epsilon\, \mathbf{R}} \sum_{i=1}^{n_p-n_{pd}} \left[1 - (1 - p_b(SNR, r))^{\left(\frac{h+S_p(i)}{r}\right)}\right] D_p(i) \\
&\quad + K_1 \\
&\quad \text{subject to} \quad \sum_{i=1}^{n_p-n_{pd}} \frac{h+S_p(i)}{r} \le \left(\frac{R_{CH} L_G}{f_s}\right)
\end{aligned} \quad (8)$$

As in Eq. 2, K_1 is the 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 dynamic programming approach described in Sections III-A and III-B is used to solve the optimization problem in Equation 8.

V. Simulation Results and Discussion

In this section we evaluate and compare the performance of **DYN-PROG-UEP**, **EEP-slice**, **EEP-slice-ENH** and **Dual15** schemes with video quality measured by PSNR and a perceptually based Video Quality Metric (VQM) discussed in [8]. 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.

A. Simulation Setup

Two CIF (352 x 288) sequences Foreman and Silent are used in our experiments. They are encoded using H.264/AVC JM 14.2 reference software [9] at 720 Kbps, frame rate 30 fps and transmitted over a 2 Mbps 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. Two reference frames are used for predicting the P and B frames. Error concealment including both temporal concealment and spatial interpolation is enabled for all the schemes evaluated in this section. For this the motion copy option provided in the JM 14.2 [9] decoder is used.

The total header size is 54 bytes per packet. 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. In the baseline **EEP-slice** scheme, the RCPC rates each packet can select from 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)}. For the remaining schemes two additional rates, $8/8$ corresponding to no coding and ∞ corresponding to discarding are also included.

B. Performance Comparison

Figure 3 shows the average PSNR (dB) and VQM performance over an AWGN channel. As the channel SNR increases, the packet error decreases and received videos achieve average PSNRs close to their error-free PSNR values. For baseline **EEP-slice** at channel SNRs $<$ 0 dB, the received videos do not decode due to a large number of packet errors. Since every slice is treated as an individual packet, greater overhead is incurred from network protocol header bits resulting in less protection. Since optimization is performed on every GOP, the optimal **EEP-slice** code rate varies according to the video bits generated in a GOP, and the lowest and highest optimal code rates derived across the GOP's was $\left[\frac{8}{18}\ \frac{8}{16}\right]$. **EEP-slice-ENH**

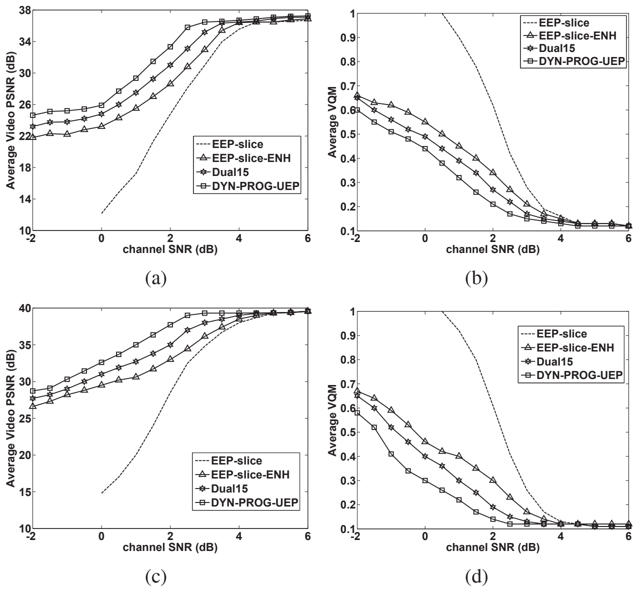

Fig. 3: Average Video PSNR (dB) and corresponding average VQM comparison computed over 100 realizations of each AWGN channel for Foreman:(a),(b) and Silent:(c),(d).

achieves higher average PSNR (dB) and corresponding lower VQM values as compared to **EEP-slice**. Aggregating slices and discarding lower priority packets allows **EEP-slice-ENH** to achieve code rates as low as $\frac{8}{32}$ and hence much better PSNR performance. For example at channel SNRs 0 and 4 dB, **EEP-slice** achieves an average PSNR of 12.2 dB and 35.6 dB whereas **EEP-slice-ENH** achieves 23.2 dB and 36.4 dB, respectively.

Dual15 does not consider packet formation through slice aggregation and only performs optimal RCPC code rate allocation for slices (considered as individual packets) of each GOP as discussed in Section III-B. It also discards least important slices if required. The packet error probability in the **Dual15** scheme is completely dependent on the optimal RCPC code rate allocated since the size of each packet is more or less the same. **DYN-PROG-UEP** takes advantage of both adaptive packet sizes as in **EEP-slice-ENH** as well as optimal RCPC packet code rate allocation as in **Dual15**. Figure 3 shows significant improvement in perceptual video quality of **Dual15** and **DYN-PROG-UEP** as compared to **EEP-slice** and **EEP-slice-ENH**. For example, at channel SNR of 2.5 dB, **EEP-slice**, **EEP-slice-ENH**, **Dual15** and **DYN-PROG-UEP** achieve average VQM of 0.42, 0.27, 0.22 and 0.17 and corresponding average PSNR values of 28.11 dB, 30.79 dB, 33.1 dB and 35.8 dB for Foreman.

Figure 4 shows the expected number of slices discarded per GOP for Foreman. Controlling the overhead due to the FEC parity bits allows **Dual15** to discard fewer slices per GOP as compared to **EEP-slice-ENH**. The gain in perceptual video quality of **DYN-PROG-UEP** is attributed to balancing both the overhead due to FEC parity bits as well as headers attached to packets, further reducing the number of discarded slices as compared to **Dual15**. **DYN-PROG-UEP** achieves maximum PSNR gain of 2.7 dB (2.5 dB) at channel SNR of 2.5 dB (1.5 dB) over **Dual15** for Foreman (Silent). **DYN-PROG-UEP** achieves maximum gains of 5 dB (4.7 dB) over **EEP-slice-ENH** at channel SNR of 2.5 dB (1.5 dB) for Foreman (Silent). Similar behavior is also observed in the VQM performance.

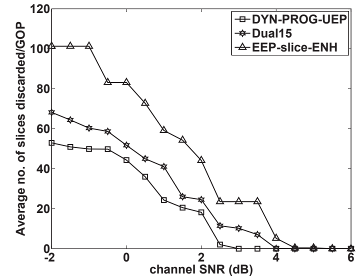

Fig. 4: Expected number of slices discarded per GOP in **EEP-slice-ENH**, **Dual15** and **DYN-PROG-UEP** for Foreman.

VI. Conclusion

An efficient joint optimization algorithm extending the work in [5] for packet formation and optimal RCPC code rate allocation was proposed to improve the quality of pre-encoded H.264/AVC bitstreams transmitted over error-prone channels. A dynamic programming approach was used where packets were formed through slice aggregation and optimal RCPC packet code rates were determined recursively over a GOP. The options of not coding or discarding less important packets were exploited through cross-layer information exchange between the PHY, MAC and APP layer to increase protection to more important packets and reduce expected received video distortion. The proposed scheme outperformed EEP schemes as well as the UEP scheme in [5], providing better video quality for different sequences.

VII. Acknowledgement

Approved for Public Release; Distribution Unlimited: 88ABW-2012-2762, 11 May 2012. This research was partially supported by awards from the U.S. Air Force Research Laboratory under agreements FA8750-08-1-0078 and FA8750-11-1-0048 at San Diego State University. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

References

[1] K. K. R. Kambhatla, S. Kumar, and P. Cosman, “Prioritized packet fragmentation for H.264 video,” in *IEEE ICIP*, September 2011, pp. 3233–3236.

[2] P. Chou and Z. Miao, “Rate-distortion optimized streaming of packetized media,” Microsoft Research, Tech. Rep. MSR-TR-2001-35, February 2001.

[3] J. Chakarakesi and P. Chou, “Application layer error correction coding for rate-distortion optimized streaming to wireless clients,” *IEEE Trans. on Communications*, vol. 52, no. 10, pp. 1675–1687, October 2004.

[4] M. van der Schaar and N. S. Shankar, “Cross-layer wireless multimedia transmission: challenges, principles and new paradigms,” *IEEE Wireless Communications Magazine*, vol. 12, no. 4, pp. 50–58, August 2005.

[5] T.-L. Lin and P. C. Cosman, “Efficient optimal RCPC code rate allocation with packet discarding for pre-encoded compressed video,” *IEEE Signal Processing Letters*, vol. 17, no. 5, pp. 505–508, May 2010.

[6] *Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard for Information Technology-Telecommunications and information exchange between systems Std., June 2007.

[7] D. Li and X. Sun, *Nonlinear Integer Programming*. International Series in Operations Research and Management Science, Springer, USA, 2006.

[8] M. H. Pinson and S. Wolf, “A new standardization method for objectively measuring video quality,” *IEEE Trans. on Broadcasting*, vol. 50, no. 3, pp. 312–322, 2004.

[9] *H.264/AVC reference software JM14.2*, ISO/IEC Std. [Online]. Available: http://iphome.hhi.de/suehring/tml/download/