A JOINT PACKET SELECTION/OMISSION AND FEC SYSTEM FOR STREAMING VIDEO

Ying-zong Huang[*]

Electrical Engineering and Computer Science
Massachusetts Institute of Technology
zong@mit.edu

John G. Apostolopoulos

Streaming Media Systems Group
Hewlett-Packard Labs, Palo Alto, CA
japos@hpl.hp.com

ABSTRACT

Media delivery over packet networks is often plagued by packet losses which limit its utility to end users. Forward Error Correction (FEC) based techniques are important for overcoming this problem. This paper further develops an FEC-based technique [1] to maximize the expected received media quality by jointly choosing which packets to send and which packets to protect – including discarding packets to make additional room for protection. We describe a straight-forward implementation leveraging existing FEC system components. Comprehensive experiments demonstrate that significant gains in PSNR of several dB are achieved when sending H.264/MPEG-4 AVC coded video over a packet erasure channel.

***Index Terms*—** Video streaming, forward error correction, FEC

1. INTRODUCTION

A challenging problem in media delivery over wired and wireless networks is how to reliably deliver media when the network may be afflicted by packet loss. A variety of techniques have been developed to overcome this problem, including forward error correction (FEC), retransmission-based techniques, error-resilient coding, error concealment, and combinations of the above. Depending on the specific situation, one or another technique may be more appropriate.

We restrict our attention to FEC-based methods. The most basic FEC approach is to treat all packets equally and provide equal protection for all of them. We refer to this class of methods as Protect All, e.g., [2]. Another approach is to focus the resources for error correction on only a subset of the packets, which increases the likelihood that the protected subset of packets can be reconstructed if lost in transmission. We refer to these schemes as Protect Subset. When the media is scalably coded, another approach is to give different levels of protection to data of different layers, e.g., with unequal error protection (UEP) where the most important data is given the highest level of protection. Such techniques for transmitting scalably coded images and video have been extensively studied, e.g., [3, 4]. Differential protection for the I, P, and B frames in conventional MPEG coded video (a form of scalable coding) is likewise based on the same principle.

While FEC is intended to *protect* data as it travels through an unreliable channel, it can also be beneficial to *discard* video packets prior to conventional FEC coding [1]. In particular, by explicity discarding data, we gain additional room for FEC, and can derive significant benefits over all. In this paper, we extend our proposed approach, which we call Discard & Protect as in [1], by showing how to design a practical system, and assessing it using extensive experimental results with H.264 video. More concretely, we view the system that results from the proposed approach as a packet-level joint source-channel coding system that optimizes the amount of data to send and the amount of FEC checks to produce jointly. It is most useful in the scenario when video is pre-encoded to a certain rate and stored at a sender and is to be transmitted over a packet erasure channel that either does not have the channel capacity to reliably deliver the video or has time-varying throughput. Additionally, we do not wish to perform a full re-encoding or transcoding of the source video to the desired rate, because in many situations this may be complex, costly, delay-infeasible, or even impossible to do (e.g., the video may be encrypted). The most relevant prior work that we are aware of involves scalable image coding and UEP [5], where given an embedded bitstream the authors decide what fraction of the bitstream to keep and how much FEC to add, in order to provide graceful degradation in the case of packet loss.

In the following, Section 2 describes the problem model in more detail. Section 3 describes the solution we propose for Discard & Protect and Section 4 gives an algorithm for practically implementing the technique. We follow with experimental results that validate the proposed approach and compare it against other techniques in Section 5. A summary wraps up the discussion.

[*]This work was performed during a summer internship at HP Labs.

2. PROBLEM STATEMENT

We assume a pre-encoded media stream with a data rate of K packets per unit time. This is stored at the sender. As the sender, we wish to transmit the stream through a packet erasure channel that has a throughput of N packets per unit time and which erases each packet independently of others with a known probability $p > 0$, referred to as the packet loss rate (PLR). The channel output is decoded one time unit at a time (i.e., in blocks of N packets) by an FEC decoder. Depending on the strength of error protection, some packets may be irrecoverable by the FEC decoder and unavailable to the media decoder. The media decoder decodes the available packets to produce the reconstructed media.

A number of FEC-based techniques, including our own, share the above system components.

- Protect All: Protect all of the data packets equally with the available redundancy
- Protect Subset: Protect a subset of the packets (typically the most important) with the available redundancy
- Protect Multiple Subsets: Protect different subsets of packets with different levels of protection, e.g., UEP often used for graceful degradation (not examined in this paper)
- Discard & Protect: Proposed technique to discard a subset and protect another subset of the packets

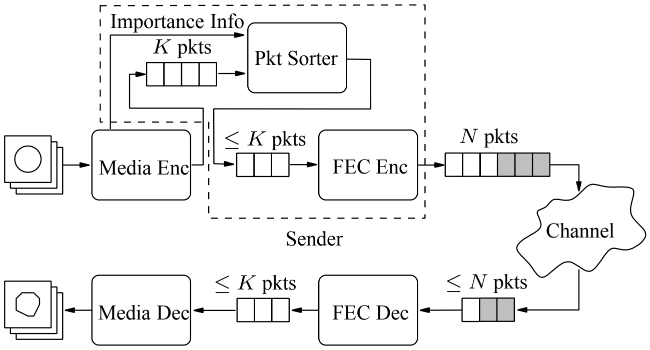

Fig. 1. System diagram (modification in dotted box)

The system model that we consider for Discard & Protect (Figure 1) is based on conventional systems that support FEC-based reliable media delivery schemes (e.g., RFC 2733 - FEC over RTP) and is a drop-in modification of existing systems. We add a "packet sorter" at the sender, which makes decisions about what to do with the video packets, which can be discarded, left unprotected, or protected with an FEC code.

As shown in Figure 1, the total additional MSE due to decoding without a particular packet – which we call the "importance" of the packet – is an additional piece of information known to the sender (either included with the source data or conveyed separately) for our scheme. Using this information, the problem is to minimize the expected end-to-end distortion by making decisions in the packet sorter.

3. PROPOSED SOLUTION

If we insist on sending all of the pre-encoded data through the channel, then the number of checksums we can fit into the channel and hence the number of erasures we can correct is limited to $N - K$ per block. When more than $N - K$ erasures may potentially occur and when losing some packets is more costly than losing others, i.e., when packets are of differing importance, we can do better by judiciously discarding data to make room for more checksums to be sent. Intuitively, it is worthwhile to sacrifice a low-importance data packet's original contents and fill it with a checksum to allow correction of one additional erasure among the higher-importance data packets.

Denote the encoder-computed importance of packet i in a block by $D_i, i \in \{1, ..., K\}$. For simplicity, we assume an additive distortion model where the distortion incurred by multiple missing packets is given by the sum of the distortions of the packets missing individually. Therefore, each packet's importance is computed as in [6]. We propose the following design. At the sender:

- At the packet sorter, discard the k_d data packets corresponding to the lowest values of D_i out of the K data packets entering, and mark the k_p packets corresponding to the highest values of D_i for protection;
- In the FEC encoder, code the k_p packets generating the highest values of D_i using a $(N - K + k_d + k_p, k_p)$ systematic Reed-Solomon code applied *across* packets[1];
- Emit the $N - K + k_d + k_p$ channel-coded packets, as well as the remaining $k_u = K - k_d - k_p$ unprotected packets, filling out a block of N channel packets.

The operation at the receiver side is conventional FEC decoding:

- Recover erased packets in the k_p protected data packets by decoding the Reed-Solomon code across packets; if the number of erasures exceeds the error correction capabilities of the code, do nothing;
- Emit the recoverable data packets along with the unerased data packets, numbering no more than $K - k_d$ packets.

This way of arranging packets was discussed in [1]. There we also showed that under simplifying assumptions where the number of losses is known and the number of packets is allowed to be fractional, then one level of error protection is sufficient and no packets are left unprotected. Furthermore, we suggested a "gap-closing" approach in optimizing for the number of packets to discard and to protect, namely, discard packets until the incremental cost of discarding packets is no longer less than the incremental gain of the increased protection. Here, we retain the design decision of one level of error protection from [1] (see relatedly [7]), and describe a "gap-closing" algorithm to find optimal values for k_d, k_p, and k_u, but without either of the simplifying assumptions.

4. DISCRETE ALGORITHM

4.1. Analysis

We index the packets by ascending values of importance D_i. Recall from Section 3 that we discard packets $1, ..., k_d$, leave packets $k_d + 1, ..., K - k_p$ unprotected, and protect packets $K - k_p + 1, ..., K$.

The expected end-to-end distortion $\bar{D}_{\text{end-to-end}} = \bar{D}_{\text{discarded}} + \bar{D}_{\text{unprotected}} + \bar{D}_{\text{irrecoverable}}$ is composed of three components depending on the treatment of the packets in question, where:

1. Expected distortion from discarded packets is:

$$\bar{D}_{\text{discarded}} = \sum_{i=1}^{k_d} D_i$$

2. Expected distortion from erasures on unprotected packets is:

$$\bar{D}_{\text{unprotected}} = p \sum_{i=k_d+1}^{K-k_p} D_i$$

3. Expected distortion from erasures on coded packets is:

$$\bar{D}_{\text{irrecoverable}} = \left\{ \frac{k_p}{n} \sum_{y=n-k_p+1}^{n} \binom{n}{y} p^y (1-p)^{n-y} \right\} \sum_{i=K-k_p+1}^{K} D_i$$

where $n = N - K + k_d + k_p$.

The final expressions of both 2 and 3 are, in each case, the expected proportion of unavailable packets among those of that type, multiplied by the cost of losing all of them.

The optimization problem is to minimize $\bar{D}_{\text{end-to-end}}$ subject to $k_d, k_p \geq 0$, and $k_d + k_p \leq K$.

[1] We use Reed Solomon codes as building blocks for our FEC-based scheme because they can recover from the maximal number of erasure losses in the MDS sense – up to the number of excess checksums. The use of systematic codes is critical because when recovery is impossible in the event of too many erasures, the data portion of the code that is unerased is still useable.

4.2. Proposed Algorithm

A procedure to search for the k_d^*, k_p^* that minimizes the objective $\bar{D}_{\text{end-to-end}}$ with a value of Λ^* is given below. It is practical, requiring at most $2K$ evaluations of the objective. In the outer loop, we increment k_d by one at a time. In the inner loop, for each k_d, we find the minimal loss Λ_d as we increment k_p by one at a time.

$\Lambda^* \Leftarrow \infty$
$k_d \Leftarrow 0$
$k_p \Leftarrow 0$
while $k_d + k_p \leq K$ **do**
  $\Lambda_d \Leftarrow \infty$
  while $k_d + k_p \leq K$ and $\bar{D}_{\text{end-to-end}} < \Lambda_d$ **do**
    $\Lambda_d \Leftarrow \bar{D}_{\text{end-to-end}}$
    $k_p \Leftarrow k_p + 1$
  end while
  $k_p \Leftarrow k_p - 1$
  if $\Lambda_d < \Lambda^*$ **then**
    $\Lambda^* \Leftarrow \Lambda_d$
    $k_d^* \Leftarrow k_d$
    $k_p^* \Leftarrow k_p$
  end if
  $k_d \Leftarrow k_d + 1$
end while

---

5. EXPERIMENTAL RESULTS

To evaluate the proposed approach we consider video coded with H.264/MPEG-4 Part 10 Advanced Video Coding (AVC) using JM 10.2 reference software. We consider a variety of realistic coding scenarios, and four standard test sequences: Carphone, Foreman, Mother & Daughter, and Salesman. Each is coded at a constant quantization level for an average PSNR of about 36 dB, 30 fps, and has at least 350 frames. Slices are periodically intra updated to improve error-resilience by reducing error propagation, corresponding to an intra update period of 36 frames. The decoder performs frame copy error concealment on lost packets. IID packet loss parameterized by the PLR p is assumed and 100 channel realizations are run to compute each data point.

The experiments compare the performance of several FEC schemes (assuming knowledge of p for each test point): (1) Protect All, (2) Protect Subset, where the most important packets are protected and the number of protected packets is chosen such that $N - K$ is the mean number of erasures that occurs in the protected and checksum packets, and (3) the proposed Discard & Protect. For Discard & Protect both the predicted and actual (empirically measured) performance are given. The following performance bounds are also identified: (4) Oracle is an upper bound computed by assuming that the number of erasures is exactly the mean for PLR p and the erasure positions are known via omniscience. The packet sorter is then permitted to rearrange the packets such that the least costly ones are erased. (5) Protect None provides a lower bound where the data packets are transmitted without protection. (6) Perfect Channel shows the case when the sequence is received without losses, corresponding to purely source-coding performance.

5.1. P- and B-frames

In this experiment (Figure 2), the video is coded with an initial I-frame followed by repeated units of BBP, or two B-frames and a P-frame. Every 4 frames a slice is intra updated and there are 9 slices in a frame, therefore the intra-update period is 36 frames. We assume that the initial I-frame is always correctly received to simplify the analysis. Every P-frame and B-frame fits within a single 1500 byte packet, hence in these experiments the loss of one packet corresponds to the loss of one frame. We expect a large gain for the proposed approach because the B-frames are of low importance and cause no error propagation if missing, facilitating their discarding.

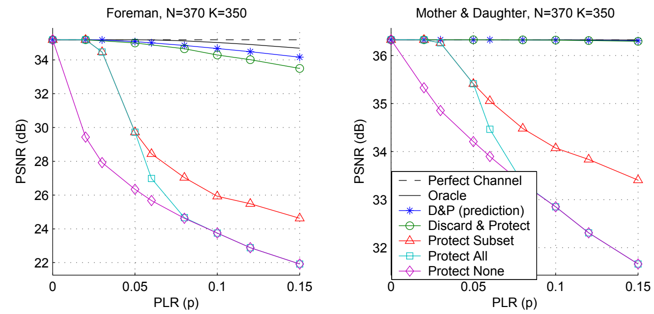

Fig. 2. IBBPBBP, QCIF with one packet per coded frame

Similar behavior was observed for each sequence, and because of the limited space only results for Foreman and Mother & Daughter are plotted. The presence of both P- and B-frames together induces a large variation in the relative importance between packets which is exploited by the proposed approach. In Mother & Daughter, for example, the performance of the proposed scheme is almost indistinguishable from perfect channel even at 15% erasure rate.

Note that in these experiments, $N - K$ is slightly more than 5% of N, hence if the realized packet loss is $\leq$ 5% then Protect All would provide complete recovery (PSNR of perfect channel). However, since the loss rate has a distribution, this is not the case and Protect All provides significantly lower performance at 5% PLR.

The following table provides an illustrative example of the performance of the various schemes at 8% PLR. Since Discard & Protect exploits the differences in importance of the coded video packets, the larger the differences between the importances of various packets, the larger gains typically observed. To give a sense of the range of importances within a particular sequence, we show the importance inter-quartile ratio (IIQR) which is the ratio between the third and first quartiles of the packet importances computed for each sequence. More details about the range of importance of packets is given in [6]. The remaining performance values are PSNR in dB.

	Foreman	M & D	Salesman	Carphone
IIQR	43.68	84.70	44.50	18.13
Protect None (LB)	24.63	33.33	33.04	29.03
Protect All	24.66	33.35	33.06	29.06
Protect Subset	27.04	34.48	33.87	31.06
Discard & Protect	34.66	36.32	35.06	35.76
D&P (Prediction)	34.86	36.32	35.06	35.83
Oracle (UB)	35.13	36.33	35.07	35.94
Perfect Channel	35.20	36.33	35.07	35.97

5.2. P-frames Only

In this experiment (Figure 3), the setup is similar to the previous one, except the video is coded with an initial I-frame followed by all P-frames, and no B-frames. We choose to code the video with all P frames in order to produce coded frames and associated packets that have a homogeneous coding dependency structure and therefore do not suggest a natural prioritization of frames (besides for the ear-

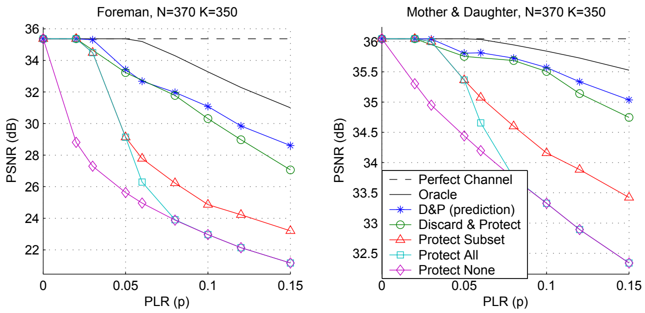

Fig. 3. IPPP, QCIF with one packet per coded frame

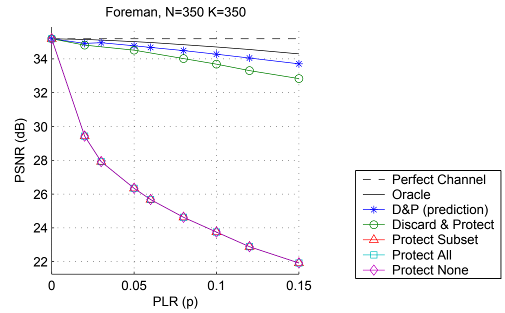

Fig. 5. IBBPBBP, QCIF with one packet per coded frame, $N = K$

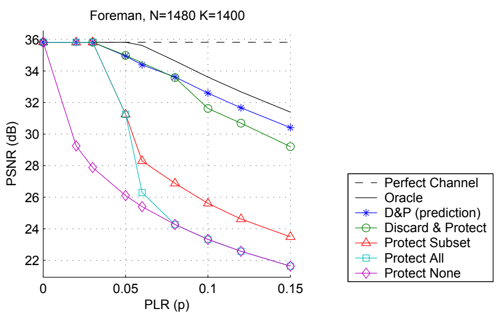

Fig. 4. IPPP, CIF with four packets per coded frame

lier P-frames being more important than the later ones), in contrast to conventioanl I, P, and B frame coding of video as discussed before or to scalably coded video. This homogeneous coding structure would appear to be a nice match for FEC designed to protect all the packets. Also, the presence of only P frames, and no B-frames, suggests a much smaller variation in the relative importance between packets than in the previous case and therefore much smaller gains. Nonetheless, even with a homogeneous coding structure, P-frames can also differ in importance from one another by a very significant amount depending on the video source – leading to significant gains.

5.3. Multiple Packets Per Frame

In this experiment (Figure 4), the CIF resolution video is coded with an initial I-frame followed by all P-frames, and no B-frames. Each P-frame is coded into four packets. Every 2 frames a slice is intra updated and there are 18 slices in a frame, therefore the intra-refresh period is still 36 frames. Every lost packet results in the loss of a quarter of a frame. It is replaced by the last correctly received corresponding quarter of the frame. Once again Discard & Protect provides sizable gains.

5.4. An Unusual Case: $N \leq K$

The derivation and the algorithm of our proposed method do not require that $N > K$, which is the normal regime of operation for FEC-based techniques. In fact, it works even for $N \leq K$ with minor modifications to the feasible set of the optimization. Figure 5 examines the unusual case when the number of transmittable packets in a block is equal to the number of data packets ($N = K$). Because there is no room for adding checksums, FEC schemes like Protect All and Protect Subset are not applicable and provide no protection against erasures (same performance as Protect None). However, Discard & Protect is able to exploit the differing importance of the data packets, and at each PLR discard the appropriate number of the least important packets and replace them with check packets, for a significant overall gain. This ability to gain FEC capabilities where there are none or in regimes where one would not normally consider applying FEC demonstrates a new flexibility enabled by this approach.

6. CONCLUSION

This paper examined the problem of improving the quality of media sent over a packet erasure channel using FEC methods. In contrast to conventional FEC techniques which protect all of the media packets or a subset of the most important media packets, the proposed technique of Discard & Protect explicitly discards the least important packets to make additional room for FEC that protects the most important packets, in order to minimize the expected end-to-end distortion. The proposed method is implementable on existing FEC transmission systems by adding a low-complexity modular packet processing block at the sender. A variety of experiments with H.264 coded video demonstrate that significant performance improvements can be achieved as compared to conventional FEC approaches. Furthermore, the proposed technique provides greater flexibility by expanding the range of PLR's for which FEC can be beneficially applied, i.e., for $N \leq K$ and PLR $\geq (N - K)/N$.

7. REFERENCES

[1] Y. Huang and J. Apostolopoulos, "Making packet erasures to improve quality of FEC-protected video," *IEEE ICIP*, October 2006.

[2] P. Frossard and O. Verscheure, "Joint source/FEC rate selection for quality-optimal MPEG-2 video delivery," *IEEE Transaction on Image Processing*, December 2001.

[3] R. Hamzaoui, V. Stankovic, and Z. Xiong, "Optimized error protection of scalable image bit streams," *IEEE Signal Processing Magazine*, November 2005.

[4] W. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," *IEEE Trans. Circuits and Systems for Video Technology*, March 2001.

[5] A.E. Mohr, E.A. Riskin, and R.E. Ladner, "Graceful degradation over packet erasure channels through forward error correction," *Data Compression Conference*, March 1999.

[6] J.G. Apostolopoulos, "Secure media streaming & secure adaptation for non-scalable video," *IEEE ICIP*, October 2004.

[7] S.S. Karande and H. Radha, "Partial Reed Solomon codes," *IEEE Information Theory Workshop*, April 2003.