MAKING PACKET ERASURES TO IMPROVE QUALITY OF FEC-PROTECTED VIDEO

Ying-zong Huang[*]

Electrical Engineering & Computer Science Dept.
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 lossy packet networks is a challenging problem, and Forward Error Correction (FEC) based techniques are an important technique for overcoming packet loss. Conventional FEC-based media delivery techniques protect all packets equally, or protect a subset of the packets, or protect different subsets of packets with different levels of protection, e.g., scalable coding with unequal error protection (UEP). This paper proposes an FEC-based technique to maximize the expected received media quality by explicitly discarding packets, when beneficial, in order to provide additional room for FEC. Given knowledge of the importance of each packet, we show that there is a simple and intuitive criterion for the optimal selection of which packets to discard and which to protect, as well as the level of protection, to minimize the expected distortion experienced at the receiver. The proposed approach provides significant gains over the conventional approaches, and these gains are illustrated for the case of sending H.264 coded video data over a packet erasure channel with known packet loss rate.

Index Terms— Video streaming, forward error correction, FEC

1. INTRODUCTION

Media delivery over wired and wireless networks is important today and will likely increase in importance in the future. A challenging problem in this context is how to reliably deliver media over a lossy packet network. 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 various combinations of the above [1, 2, 3, 4]. Depending on the specific situation, one or another technique may be more appropriate. For example, if real-time encoding is performed and the encoder receives timely feedback about which packets are received by the decoder, or about the channel loss characteristics, then a variety of feedback-aware techniques can be applied [4].

This paper examines the use of FEC for reliable media delivery over a lossy packet network, and for simplicity of discussion we consider the case when the media is pre-encoded and stored at the sender. In addition, we assume that the probability of packet loss is known to the sender. Given the pre-encoded content, one typical FEC approach is to add sufficient FEC packets so that the receiver may recover all of the transmitted media packets. This approach treats all packets equally and provides equal error protection across all packets in an attempt to recover all packets – we refer to this class of techniques as "protect all", e.g., [5]. When the media is scalably coded, another approach is to apply unequal error protection (UEP) where the more important data is provided greater protection than the less important data. The application of UEP with scalably coded images and video has been extensively studied [6, 7], and this has also been applied to provide differential protection for the I, P, and B frames in conventional MPEG coded video (also a form of scalable coding). Under this approach, when the packet loss rate is higher than the amount of redundancy which can be applied, the redundancy is allocated to protect the most important data, e.g., applied to protect the base layer of a scalably coded media or to the I and P frames for an MPEG coded video (and the B frames are not protected). We refer to this class of schemes as "protect subset".

This paper examines the problem of determining what is the best FEC strategy to minimize the expected distortion at the receiver. We propose an alternative technique to improve FEC performance as compared to the "protect all" and "protect subset" classes of approaches described above. Specifically, when necessary, we propose to protect a subset of the pre-encoded data by first *discarding* another subset of the data. While it may appear illogical to intentionally introduce additional losses as the first step of delivering data, we show that appropriate selections of what to discard and what to protect can provide significant benefits. In particular, by explicitly discarding data, we gain additional room for FEC, which is then used to provide improved protection for the more important remaining data.

This paper continues in Section 2 by describing the specific problem to be examined and different strategies for overcoming it. Section 3 examines the proposed strategy in detail, and analytically determines the optimal solution assuming the number of packets is given by a continuous value – which provides beneficial insight into the solution. The case of integer number of packets is examined in Section 4. Experimental results validating the proposed technique are given in Section 5, followed by a summary.

2. PROBLEM STATEMENT

In the following, we assume that the media is coded into k packets per unit of time, and each packet is independently decodable. We wish to transmit the coded media through a link with throughput of n packets per unit time, and which erases packets in an i.i.d. manner with a known probability $p > 0$, referred to as the packet loss rate (PLR). The media is assumed to be decoded one time unit at a time (i.e., block of up to n received packets), by an FEC decoder followed by a media decoder. Depending on the strength of error protection, some packets may be irrecoverable by the FEC decoder and unavailable to the media decoder. The quality of reconstructed media is determined by the packets which are received, FEC decoded, and available to the media decoder. The distortion due to the unavailability of each individual packet is assumed to be known, and expressed by $D(i)$ where i is the packet index. In addition, we as-

[*]This work was performed while attending the Researchers in Residence program at HP Labs.

sume that the total quality loss (i.e. total amount of distortion) due to unavailable packets is additive in the distortions caused by the individual lost data packets.

There are a number of FEC-based strategies for overcoming the distortion consequences of packet erasure losses, which can roughly be classified as below:

- *Protect All* : Protect all of the packets equally with the available FEC
- *Protect Subset* : Protect a subset of packets (typically the most important) with the available FEC
- *Protect Different Subsets with UEP* : Protect different subsets of packets with different levels of protection (UEP)
- *Discard & Protect* : Proposed technique to discard a subset and protect subset(s) of packets (also possibly with UEP)

Clearly, if the available FEC is sufficient to protect all of the packets from potential erasures, then the *Protect All* technique already provides the best quality in the sense that all of the coded packets will be recovered and the distortion will be zero. However, when the available FEC is not sufficient to protect all of the packets then the situation is unclear – and this is the operational situation which motivated this work. Since we consider the case where the media is pre-encoded and stored for later delivery, the sender can not re-encode the media at a different bit rate, and the sender's options are limited to those mentioned above. We examine this situation next.

3. PROPOSED SOLUTION

To simplify the analysis in this section, we neglect the fact that the numbers of packets are integer values, instead assuming they can take on real values. The case of integer number of packets is examined in the subsequent section. Relatedly, we assume we have distortion density $D(x)$ as a staircase function of the continuous-domain packet number x, obtained from the discrete analog. With these simplifications, we can state that:

> The best distortion performance at a PLR of p is achieved by overwriting the least distorting d^* data packets with FEC and protecting the remaining data packets, where $d^* = \max\{pn - (n-k), 0\}$.

We next show this is true.

Given a particular way of coding k data packets, some of the data packets are discarded/overwritten, some are left unprotected, and some receive error protection. At a PLR of p, the expected distortion due to transmitting packet i is, by case:

- Discarded: $D(i)$;
- Unprotected: $pD(i)$;
- Protected: 0 if $p \leq p_{th}$ and $pD(i)$ if $p > p_{th}$, where p_{th} is the threshold where FEC fails for packet i, determined by code parameters.

These characteristics are illustrated in Figure 1. The overall expected distortion is the aggregate sum of the distortion performances of each packet. The plot of the overall distortion vs. PLR is piecewise linear and exhibits discontinuities at any PLR that is at a threshold of FEC decoding failure.

We identify a best-case lower bound for the distortion vs. PLR of any coding strategy: when packet unavailability occurs in check packets first, then in the data packets in the order of increasing distortion. This gives the lowest possible distortion at any PLR. In Figure

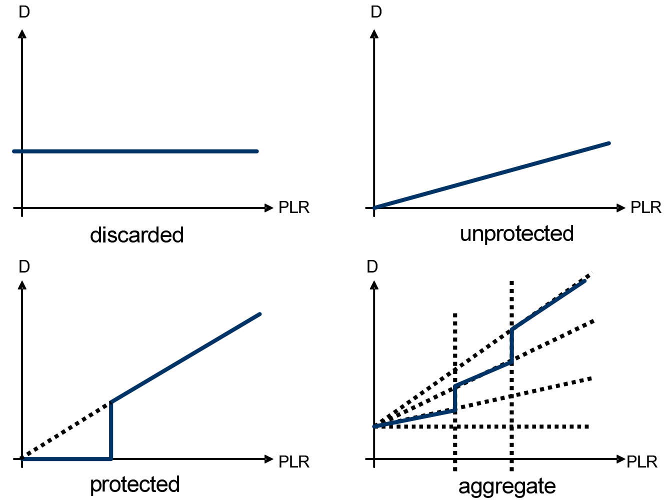

Fig. 1. Effects as a function of PLR for a subset of packets when (1) discarding the subset, (2) leaving subset unprotected, (3) protecting to a single level of protection; and (4) aggregate result for discarding a subset, leaving another subset unprotected, and providing two levels of protection.

2, we plot the distortion vs. PLR performances of several example schemes that try to transmit $k = 285$ data packets across a link with capacity of $n = 300$ packets, providing room for 15 FEC packets initially. These example results are using H.264 coded video packets as described in Section 5. The three schemes are (1) *Protect All* where all 285 video packets are protected with a $(300, 285)$ code, (2) *Protect Subset* where the 190 most important video packets are protected with a $(205, 190)$ code and the remaining video packets are unprotected, and (3) *Discard & Protect* where the 15 least important video packets are discarded and replaced by extra checks in a $(300, 270)$ code to provide additional protection to the remaining more important video packets. The performances of these are plotted against the best-case lower bound. The expected distortion is normalized so that if all of the packets are lost the total distortion would be 1. Note that for *Protect All* the distortion is zero for PLR less than 5%, and rises linearly for larger PLRs. As shown via these examples, if the available FEC is sufficient to overcome the PLR then *Protect All* is the best solution. However, if the PLR is greater than what the available room for FEC can handle, then *Protect Subset* and *Discard & Protect* provide significant gains in performance.

No scheme performs better than all others over all PLRs. However, we can identify the best scheme at each PLR and show that the statement at the beginning of this section is true.

Claim 1: Suppose we rank order the k data packets by their associated distortion-on-loss in the order of rising distortion (packet 1 least, packet k most). Then the best scheme at any PLR p can be put in the form that: packets numbered $1, ..., d$ are overwritten, packets $d+1, ...m$ are unprotected, and packets $m+1, ..., k$ are protected. See notation in Figure 3. Therefore, we take $D(x)$ to be monotonic non-decreasing without loss of generality.

Claim 2: The number of check packets available for FEC purposes is $r = \max\{n - k + d, 0\}$.

Claim 3: Given r check packets, the best scheme at PLR p uses a single $(s+r, s)$ code to protect s data packets, where s is set equal to $\frac{1-p}{p} r$ so that erasures in the protected packets are correctable.

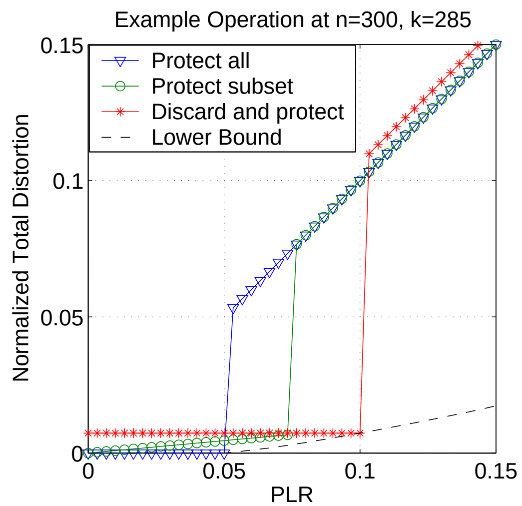

Fig. 2. Examples of operation versus PLR.

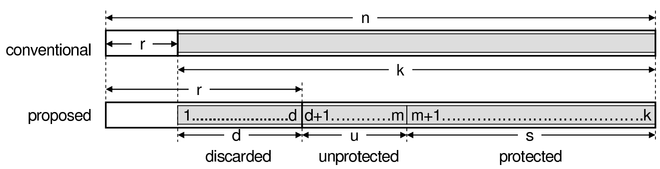

Fig. 3. Notation illustrating conventional (n, k) code (top) and proposed technique for discard/unprotect/protect when packet index is sorted by distortion (bottom). Pre-encoded data packets are shaded.

Therefore, we are left only to pick the optimal value for d, which then determines s and u, given n, k, and p. The expected total distortion of an optimal scheme in terms of d is:

$$D_{total} = \int_{x=0}^{d} D(x)dx + p \int_{x=d}^{m} D(x)dx$$

where m is determined by $m = k - \frac{1-p}{p}(n - k + d)$. We find the d that minimizes D_{total} by taking a derivative and setting it to zero:

$$\begin{aligned} 0 = \frac{\partial D_{total}}{\partial d} &= D(d) + p[\frac{\partial m}{\partial d} D(m) - D(d)] \\ &= D(d) + p[-\frac{1-p}{p} D(m) - D(d)] \\ &= (1-p)[D(d) - D(m)] \end{aligned}$$

By the monotonic non-decreasing property of $D(x)$, $\frac{\partial D_{total}}{\partial d} \leq 0$ as long as $m \geq d$, so the total distortion is decreased as d increases incrementally (and m decreases), until the point $d = m$. We take this last value to be d^*, i.e. $d^* = k - \frac{1-p}{p}(n - k + d^*)$, which is equivalent to the statement at the beginning of this section.

Also note that this formulation is valid for the case $n < k$ without modification; this more unusual case will also be examined in the experimental results.

4. DISCRETIZED ALGORITHM

In practice, we need a scheme that takes into account the discrete nature of packets, so we modify our statement from Section 3 for the case of integer numbers of packets.

The best distortion performance at a PLR of p is achieved by overwriting the least distorting d^* data packets with FEC and protecting the most distorting s^* data packets where s^* is either $\left\lfloor \frac{1-p}{p}(n - k + d^*) \right\rfloor$ or $k - d^*$. The remaining data packets, if any, are left unprotected. The values d^* and s^* are determined by $D(x)$.

We note that we only need to modify one claim from the previous section, along with its consequences, to account for the discrete case:

Claim 3* (discrete): Given r check packets, the best scheme at PLR p uses a single $(s + r, s)$ code to protect s data packets, where s is set equal to $\left\lfloor \frac{1-p}{p} r \right\rfloor$ so that erasures in the protected packets are correctable.

For $n \geq k$, the following algorithm finds the optimal values d^* and s^* by searching at most $\lceil pn - (n - k) \rceil + 1$ values of d and finding the one that gives minimum distortion D^*_{total}. A slight modification makes this algorithm work for $n < k$ as well.

$D^*_{total} \Leftarrow \infty$
$d \Leftarrow 0$
while $d + \left\lfloor \frac{1-p}{p}(n + d - k) \right\rfloor < k$ **do**
  $D_{total} \Leftarrow \sum_{i=1}^{d} D(i) + p \sum_{i=d+1}^{k - \left\lfloor \frac{1-p}{p}(n+d-k) \right\rfloor} D(i)$
  if $D_{total} < D^*_{total}$ **then**
    $D^*_{total} \Leftarrow D_{total}$
    $d^* \Leftarrow d$
    $s^* \Leftarrow \left\lfloor \frac{1-p}{p}(n + d - k) \right\rfloor$
  end if
  $d \Leftarrow d + 1$
end while
$D_{total} \Leftarrow \sum_{i=1}^{d} D(i)$
if $D_{total} < D^*_{total}$ **then**
  $D^*_{total} \Leftarrow D_{total}$
  $d^* \Leftarrow d$
  $s^* \Leftarrow k - d^*$
end if

5. EXPERIMENTAL RESULTS

To examine the potential benefits of the proposed approach, we consider the case with H.264/MPEG-4 Part 10 Advanced Video Coding (AVC) coded video. Specifically, we consider when 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, which have a homogeneous coding dependency structure and therefore do not suggest a natural prioritization of frames (besides for the earlier P-frames being more important than the later ones), in contrast to scalably coded video or video coded with I, P, and B frames. This homogeneous coding structure would appear to be a nice match for FEC designed to protect all the packets. 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 – something that can be beneficially exploited [8]. Note that the proposed approach extends trivially to include I and B frames, or other coding structures which produce coded data of different importance.

Four standard test sequences in QCIF format were examined, *Carphone, Foreman, Mother & Daughter,* and *Salesman*. Each is coded at a constant quantization level for an average PSNR of about

36 dB, at 30 fps, and has at least 350 frames, in the same manner as in [8]. The first frame of each sequence is intra-coded, followed by all P-frames. Every 4 frames a slice is intra updated to improve error-resilience by reducing error propagation corresponding to an intra update period of $N = 4 \times 9 = 36$ frames. Every P-frame fits within a single 1500 byte packet, hence in these experiments the loss of one packet corresponds to the loss of one P-frame. Every lost frame is replaced by the last correctly received frame, and distortion is measured after decoder error concealment. We assume that the initial I-frame is always correctly received to simplify the analysis. Similar behavior was observed for each video sequence, and because of the limited space only results for *Carphone* are shown.

Note that for simplicity, the experiments assume that the number of packets lost is exactly given by the PLR times the number of transmitted packets, which produces performance curves with sharp cutoffs that are easy to interpret. When the packet loss PDF has a more conventional distribution (e.g., binomial PDF) then the problem is solved by minimizing the expected distortion over the distribution.

The optimal performance achievable by each technique at each PLR for 285 video packets and 300 transmittable packets is shown in Figure 4. Specifically, the result at each PLR corresponds to the optimal result possible at that PLR by that technique. For example, while *Protect All* does not vary the processing for different PLRs (it operates as an $(300, 285)$ code for all PLRs), *Protect Subset* and *Discard & Protect* do vary their operation as a function of PLR in order to optimize their performance. *Protect Subset* varies the size of the subset protected so those packets (the most important packets) will not suffer losses at the given PLR. At each PLR, *Discard & Protect* operates as described in Section 4 to minimize the expected distortion. Clearly, the proposed *Discard & Protect* can provide significant benefits over the conventional *Protect All* and *Protect Subset*.

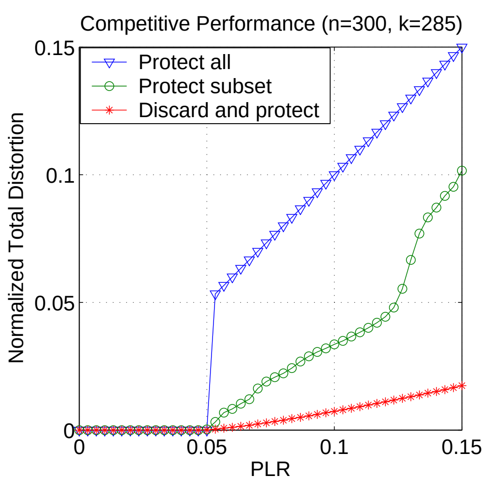

Fig. 4. Optimal performance at each PLR for each scheme, given knowledge of the PLR.

Figure 5 examines the unusual cases when the number of transmittable packets is equal to ($n = k$) and when it is less than ($n < k$) the number of original coded packets. Note that in both cases there is no room for FEC packets, so *Protect All* and *Protect Subset* both protect nothing. In the $n < k$ case, not only is there no room for FEC, but some 15 packets must be discarded. *Protect All* is assumed to discard 15 packets at random, whereas *Protect Subset*, which operates with packet distortion information, is assumed to discard the 15 least important packets. *Protect All* and *Protect Subset* do not vary their operation with PLR here. On the other hand, *Discard & Protect* at each PLR optimally discards the appropriate number of least important packets (beyond 15 when necessary), thereby creating FEC capabilities where there was none, to provide optimal protection. These unusual cases ($n = k$ and $n < k$) demonstrate the flexibility and benefits of the *Discard & Protect* technique.

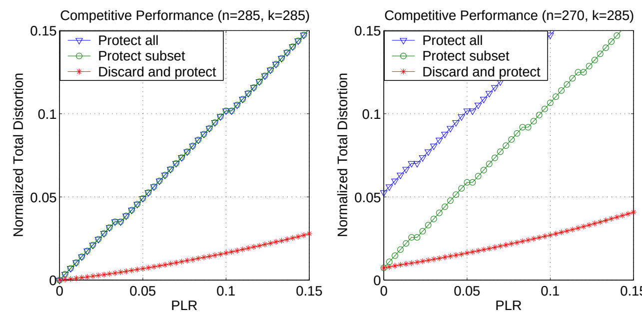

Fig. 5. Optimal performance at each PLR for each scheme, for the unusual cases of $n = k$ (left) and $n < k$ (right).

6. SUMMARY

This paper examined the problem of improving the quality of FEC protected media sent over a packet erase channel. 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 explicitly discards the least important packets to make additional room for FEC to protect the most important packets. Given information on the relative importance of the packets, we determine the optimal number and selection of packets to discard as well as to protect, and the optimal level of protection, to minimize the expected distortion at the decoder. This highly flexible approach is also applicable in cases when FEC is not normally considered, e.g., the number of transmittable packets is equal to ($n = k$) or less than ($n < k$) the number of original packets. Experimental results with H.264 coded video demonstrate that significant performance improvements can be achieved as compared to conventional FEC approaches.

7. REFERENCES

[1] S. Rane, A. Aaron, and B. Girod, "Systematic lossy forward error protection for error-resilient digital video broadcasting - a Wyner-Ziv coding approach," *IEEE ICIP*, October 2004.

[2] R. Puri and K. Ramchandran, "PRISM: a new robust video coding architecture based on distributed compression principles," *Proc. of Allerton Conference on Communication, Control, and Computing*, Oct. 2002.

[3] A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-Ziv coding of video: an error-resilient compression framework," *IEEE Transactions on Multimedia*, April 2004.

[4] B. Girod and N. Färber, "Feedback-based error control for mobile video transmission," *Proceedings of the IEEE*, October 1999.

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

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

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

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