

Optimal Erasure Protection Assignment for Scalable Compressed Data with Small Channel Packets and Short Channel Codewords

Johnson Thie

*School of Electrical Engineering & Telecommunications, The University of New South Wales, Sydney, NSW 2052, Australia
Email: j.thie@ee.unsw.edu.au*

David Taubman

*School of Electrical Engineering & Telecommunications, The University of New South Wales, Sydney, NSW 2052, Australia
Email: d.taubman@unsw.edu.au*

Received 24 December 2002; Revised 7 July 2003

We are concerned with the efficient transmission of scalable compressed data over lossy communication channels. Recent works have proposed several strategies for assigning optimal code redundancies to elements in a scalable data stream under the assumption that all elements are encoded onto a common group of network packets. When the size of the data to be encoded becomes large in comparison to the size of the network packets, such schemes require very long channel codes with high computational complexity. In networks with high loss, small packets are generally more desirable than long packets. This paper proposes a robust strategy for optimally assigning elements of the scalable data to clusters of packets, subject to constraints on packet size and code complexity. Given a packet cluster arrangement, the scheme then assigns optimal code redundancies to the source elements subject to a constraint on transmission length. Experimental results show that the proposed strategy can outperform previously proposed code redundancy assignment policies subject to the above-mentioned constraints, particularly at high channel loss rates.

Keywords and phrases: unequal error protection, scalable compression, priority encoding transmission, image transmission.

1. INTRODUCTION

In this paper, we are concerned with reliable transmission of scalable data over lossy communication channels. For the last decade, scalable compression techniques have been widely explored. These include image compression schemes, such as the embedded zerotree wavelet (EZW) [1] and set partitioning in hierarchical trees (SPIHT) [2] algorithms and, most recently, the JPEG2000 [3] image compression standard. Scalable video compression has also been an active area of research, which has recently led to MPEG-4 fine granularity scalability (FGS) [4]. An important property of a scalable data stream is that a portion of the data stream can be discarded or corrupted by a lossy communication channel without compromising the usefulness of the more important portions. A scalable data stream is generally made up of several elements with various dependencies such that the loss of a single element might render some or all of the subsequent elements useless but not the preceding elements.

For the present work, we focus our attention on “erasure” channels. An erasure channel is one whose data, prior to transmission, is partitioned into a sequence of symbols,

each of which either arrives at the destination without error, or is entirely lost. The erasure channel is a good model for modern packet networks, such as Internet protocol (IP) and its adoption, general packet radio services (GPRS), into the wireless realm. The important elements are the network's packets, each of which either arrives at the destination or is lost due to congestion or corruption. Whenever there is at least one bit error in an arriving packet, the packet is considered lost and so discarded. A key property of the erasure channel is that the receiver knows which packets have been lost.

In the context of erasure channels, Albanese et al. [5] pioneered an unequal error protection scheme known as priority encoding transmission (PET). The PET scheme works with a family of channel codes, all of which have the same codeword length N , but different source lengths, $1 \leq k \leq N$. We consider only “perfect codes” which have the key property that the receipt of any k out of the N symbols in a codeword is sufficient to recover the k source symbols. The amount of redundancy $R_{N,k} = N/k$ determines the strength of the code, where smaller values of k correspond to stronger codes.

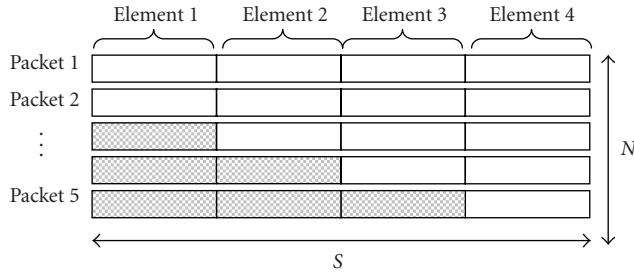


FIGURE 1: An example of PET arrangement of source elements into packets. Four elements are arranged into $N = 5$ packets with size S bytes. Elements \mathcal{E}_1 to \mathcal{E}_4 are assigned $k = \{2, 3, 4, 5\}$, respectively. The white areas correspond to the elements' content while the shaded areas contain parity information.

We define a scalable data source to consist of groups of symbols, each of which is referred to as the “source element” \mathcal{E}_q having L_q symbols. Although in our experiment, each symbol corresponds to one byte, the source symbol is not restricted to a particular unit. Given a scalable data source consisting of source elements $\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_Q$ having uncoded lengths L_1, L_2, \dots, L_Q and channel code redundancies $R_{N,k_1} \geq R_{N,k_2} \geq \dots \geq R_{N,k_Q}$, the PET scheme packages the encoded elements into N network packets, where source symbols from each element \mathcal{E}_q occupy k_q packets. This arrangement guarantees that the receipt of any k packets is sufficient to recover all elements \mathcal{E}_q with $k_q \leq k$. The total encoded transmission length is $\sum_q L_q R_{N,k_q}$, which must be arranged into N packets, each having a packet size of S bytes. Figure 1 shows an example of arranging $Q = 4$ elements into $N = 5$ packets. Consider element \mathcal{E}_2 , which is assigned a $(5, 3)$ code. Since $k_2 = 3$, three out of the five packets contain the source element's L_2 symbols. The remaining $N - k_2 = 2$ packets contain parity information. Hence, receiving any three packets guarantees recovery of element \mathcal{E}_2 and also \mathcal{E}_1 .

Given the PET scheme and a scalable data source, several strategies have been proposed to find the optimal channel code allocation for each source element under the condition that the total encoded transmission length is no greater than a specified maximum transmission length $L_{\max} = NS$ [6, 7, 8, 9, 10, 11, 12]. The optimization objective is an expected utility U which must be an additive function of the source elements that are correctly received. That is,

$$U = U_0 + \sum_{q=1}^Q U_q P_{N,k_q}, \quad (1)$$

where U_0 is the amount of utility at the receiver when no source element is received and P_{N,k_q} is the probability of recovering element \mathcal{E}_q , which is assigned an (N, k_q) code. This probability equals the probability of receiving at least k_q out of N packets for $k_q > 0$. If a source element is not transmitted, we assign the otherwise meaningless value of $k_q = 0$ for which $R_{N,k_q} = 0$ and $P_{N,k_q} = 0$. As an example, for a scalable compressed image, $-U$ might represent the mean

square error (MSE) of the reconstructed image, while U_q is the amount of reduction in MSE when element \mathcal{E}_q is recovered correctly. In the event of losing all source elements, the reconstructed image is “blank” so $-U_0$ corresponds to the largest MSE and is equal to the variance of the original image. The term U_0 is included only for completeness; it plays no role in the intuitive or computational aspects of the optimization problem.

Unfortunately, these optimization strategies rely upon the PET encoding scheme. This requires all of the encoded source elements to be distributed across the same N packets. Given a small packet size and large amount of data, the encoder must use a family of perfect codes with large values of N . For instance, transmitting a 1 MB source using ATM cells with a packet size of 48 bytes requires $N = 21,000$. This imposes a huge computational burden on both the encoder and the decoder.

In this paper, we propose a strategy for optimally assigning code redundancies to source elements under two constraints. One constraint is transmission length, which limits the amount of encoded data being transmitted through the channel. The second constraint is the length of the channel codewords. The impact of this constraint depends on the channel packet size and the amount of data to be transmitted. In Sections 2 and 3, we explore the nature of scalable data and the erasure channel model. We coin the term “cluster of packets” (COP) to refer to a collection of network packets whose elements are jointly protected according to the PET arrangement illustrated in Figure 1. Section 4 reviews the code redundancy assignment strategy under the condition that all elements are arranged into a single COP; accordingly, we identify this as the “UniCOP assignment” strategy.

In Section 5, we outline the proposed strategy for assigning source elements into several COPs, each of which is made up of at most N channel packets, where N is the length of the channel codewords. Whereas packets are encoded jointly within any given COP, separate COPs are encoded independently. The need for multiple COPs arises when the maximum transmission length is larger than the specified COP size, NS . We use the term “MultiCOP assignment” when referring to this strategy. Given arrangement of source elements into COPs together with a maximum transmission length, we find the optimal code redundancy $R_{N,k}$ for each source element so as to maximize the expected utility U . Section 6 provides experimental results in the context of JPEG2000 data streams.

2. SCALABLE DATA

Scalable data is composed of nested elements. The compression of these elements generally imposes dependencies among the elements. This means that certain elements cannot be correctly decoded without first successfully decoding certain earlier elements. Figure 2 provides an example of dependency structure in a scalable source. Each “column” of elements $\mathcal{E}_{1,y}, \mathcal{E}_{2,y}, \dots, \mathcal{E}_{X,y}$ has a simple chain of dependencies, which is expressed as $\mathcal{E}_{1,y} < \mathcal{E}_{2,y} < \dots < \mathcal{E}_{X,y}$. This means that the element $\mathcal{E}_{1,y}$ must be recovered before the

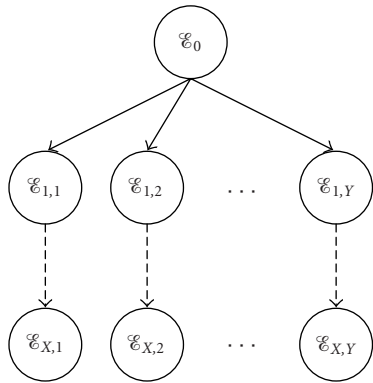


FIGURE 2: Example of dependency structure of scalable sources.

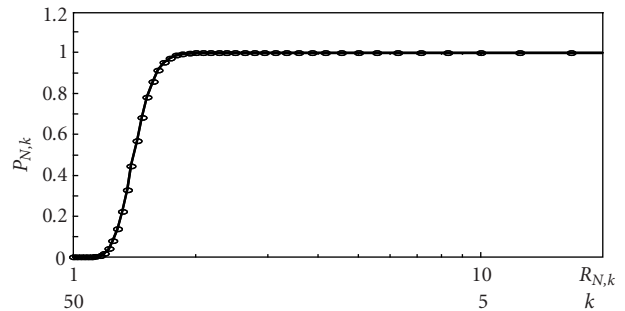
information in element $\mathcal{E}_{2,y}$ can be used and so forth. Since each column depends on element \mathcal{E}_0 , this element must be recovered prior to the attempt to recover the first element of every column. There is, however, no dependency between the columns, that is, $\mathcal{E}_{x,y} \not\prec \mathcal{E}_{\bar{x},\bar{y}}$ and $\mathcal{E}_{x,y} \not\prec \mathcal{E}_{\bar{x},\bar{y}}$, $y \neq \bar{y}$. Hence, the elements from one column can be recovered without having to recover any elements belonging to other columns.

An image compressed with JPEG2000 serves as a good example, since it can have a combination of dependent and independent elements. Dependencies exist between successive “quality layers” within the JPEG2000 data stream, where an element which contributes to a higher quality layer cannot be decoded without first decoding elements from lower quality layers. JPEG2000 also contains elements which exhibit no such dependencies. In particular, subbands from different levels in the discrete wavelet transform (DWT) are coded and represented independently within the data stream. Similarly, separate colour channels within a colour image are also coded and represented independently within the data stream.

Elements of the JPEG2000 compressed data stream form a tree structure, as depicted in Figure 2. The data stream header becomes the “root” element. The “branches” correspond to independently coded precincts, each of which is decomposed into a set of elements with linear dependencies.

3. CHANNEL MODEL

The channel model we use is that of an erasure channel, having two important properties. One property is that packets are either received without any error or discarded due to corruption or congestion. Secondly, the receiver knows exactly which packets have been lost. We assume that the channel packet loss process is i.i.d., meaning that every packet has the same loss probability p and the loss of one packet does not influence the likelihood of losing other packets. To compare the effect of different packet sizes, it is useful to express the probability p in terms of a bit error probability or bit error rate (BER) ϵ . To this end, we will assume that packet loss arises from random bit errors in an underlying binary symmetric channel. The probability of losing any packet with size S bytes is then $p = 1 - (1 - \epsilon)^{8S}$. The probability of receiving


 FIGURE 3: Example of $P_{N,k}$ versus $R_{N,k}$ characteristic with $N = 50$ and $p = 0.3$.

at least k out of N packets with no error is then

$$P_{N,k} = \sum_{i=k}^N \binom{N}{i} (1-p)^i p^{N-i}. \quad (2)$$

Figure 3 shows an example of the relationship between $P_{N,k}$ and $R_{N,k}$ for the case $p = 0.3$. Evidently, $P_{N,k}$ is monotonically increasing with $R_{N,k}$. Significantly, however, the curve is not convex.

It is convenient to parametrize $P_{N,k}$ and $R_{N,k}$ by a single parameter

$$r = \begin{cases} N+1-k, & k > 0, \\ 0, & k = 0, \end{cases} \quad (3)$$

and to assume N implicitly for simpler notation so that

$$P(r) = P_{N,N+1-r}, \quad R(r) = R_{N,N+1-r} \quad (4)$$

for $r = 1, \dots, N$. It is also convenient to define

$$P(0) = R(0) = 0. \quad (5)$$

The parameter r is more intuitive than k since r increases in the same direction as $P(r)$ and $R(r)$. The special case $r = 0$ means that the relevant element is not transmitted at all.

4. UNICOP ASSIGNMENT

We review the problem of assigning an optimal set of channel codes to the elements of a scalable data source, subject to the assumption that all source elements will be packed into the same set of N channel packets, where N is the codeword length. The number of packets N and packet size S are fixed. This is the problem addressed in [6, 7, 8, 9, 10, 11, 12], which we identified earlier as the UniCOP assignment problem. Puri and Ramchandran [6] provided an optimization technique based on the method of Lagrange multipliers to find the channel code allocation. Mohr et al. [7] proposed a local search algorithm and later a faster algorithm [8] which is essentially a Lagrangian optimization. Stankovic et al. [11]

also presented a local search approach based on a fast iterative algorithm, which is faster than [8]. All these schemes assume that the source has convex utility-length characteristic. Stockhammer and Buchner [9] presented a dynamic programming approach which finds an optimal solution for convex utility-length characteristics. However, for general utility-length characteristics, the scheme is close to optimal. Dumitrescu et al. [10] proposed an approach based on a global search, which finds a globally optimal solution for both convex and nonconvex utility-length characteristics with similar computation complexity. However, for convex sources, the complexity is lower since it need not take into account the constraint from the PET framework that the amount of channel code redundancy must be nonincreasing. The UniCOP assignment strategy we discuss below is based on a Lagrangian optimization similar to [6]. However, this scheme not only works for sources with convex utility-length characteristic but also applies to general utility-length characteristics. Unlike [10], the complexity in both cases is about the same and the proposed scheme does not need to explicitly include the PET constraint since the solution will always satisfy that constraint. Most significantly, the UniCOP assignment strategy presented here serves as a stepping stone to the “MultiCOP assignment” in Section 5, where the behaviour with nonconvex sources will become important.

Suppose that the data source contains Q elements and each source element \mathcal{E}_q has a fixed number of source symbols L_q . We assume that the data source has a simple chain of dependencies $\mathcal{E}_1 < \mathcal{E}_2 < \dots < \mathcal{E}_Q$. This dependency will in fact impose a constraint that the code redundancy of the source elements must be nonincreasing, $R_{N,k_1} \geq R_{N,k_2} \geq \dots \geq R_{N,k_Q}$, equivalently, $r_1 \geq r_2 \geq \dots \geq r_Q$, such that the recovery of the element \mathcal{E}_q guarantees the recovery of the elements \mathcal{E}_1 to \mathcal{E}_{q-1} . Generally, the utility-length characteristic of the data source can be either convex or nonconvex. To impart intuition, we begin by considering the former case in which the source utility-length characteristic is convex, as illustrated in Figure 4. That is,

$$\frac{U_1}{L_1} \geq \frac{U_2}{L_2} \geq \dots \geq \frac{U_Q}{L_Q}. \quad (6)$$

We will later need to consider nonconvex utility-length characteristics when extending the protection assignment algorithm to multiple COPs even if the original source’s utility-length characteristic was convex. Nevertheless, we will defer the generalization to nonconvex sources for the moment until Section 4.2 so as to provide a more accessible introduction to ideas.

4.1. Convex sources

To develop the algorithm for optimizing the overall utility U , we temporarily ignore the constraint $r_1 \geq \dots \geq r_Q$, which arises from the dependence between source elements. We will show later that the solution we obtain will always satisfy this constraint by virtue of the source convexity. Our optimization problem is to optimize the utility function given in (1),

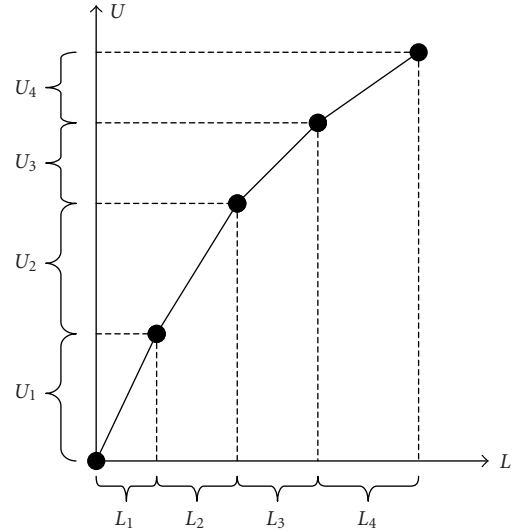


FIGURE 4: Example of convex utility-length characteristic for a scalable source consisting of four elements with a simple chain of dependencies.

subject to the overall transmission length constraint

$$L = \sum_{q=1}^Q L_q R(r_q) \leq L_{\max}. \quad (7)$$

This constrained optimization problem may be converted to a family of unconstrained optimization problems parametrized by a quantity $\lambda > 0$. Specifically, let $U^{(\lambda)}$ and $L^{(\lambda)}$ denote the expected utility and transmission length associated with the set $\{r_q^{(\lambda)}\}_{1 \leq q \leq Q}$, which maximize the functional

$$\begin{aligned} J^{(\lambda)} &= U^{(\lambda)} - \lambda L^{(\lambda)} \\ &= \sum_{q=1}^Q U_q P(r_q^{(\lambda)}) - \lambda L_q R(r_q^{(\lambda)}). \end{aligned} \quad (8)$$

We omit the term U_0 since it only introduces an offset to the optimization expression and hence does not impact its solution. Evidently, it is impossible to increase U beyond $U^{(\lambda)}$ without also increasing L beyond $L^{(\lambda)}$. Thus if we can find λ such that $L^{(\lambda)} = L_{\max}$, the set $\{r_q^{(\lambda)}\}$ will form an optimal solution to our constrained problem. In practice, the discrete nature of the problem may prevent us from finding a value λ such that $L^{(\lambda)}$ is exactly equal to L_{\max} , but if the source elements are small enough, we should be justified in ignoring this small source of suboptimality and selecting the smallest value of λ such that $L^{(\lambda)} \leq L_{\max}$. The unconstrained optimization problem decomposes into a collection of Q separate maximization problems. In particular, we seek $r_q^{(\lambda)}$ which maximizes

$$J_q^{(\lambda)} = U_q P(r_q^{(\lambda)}) - \lambda L_q R(r_q^{(\lambda)}) \quad (9)$$

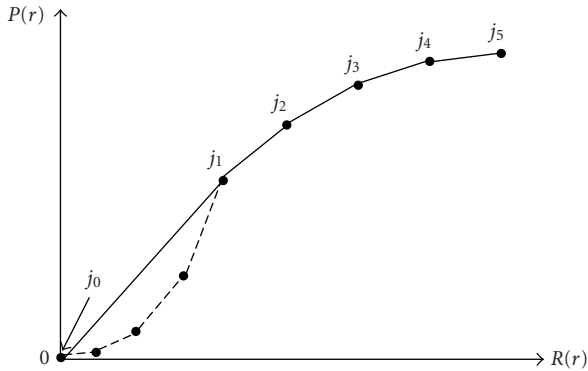


FIGURE 5: Elements of a convex hull set are the vertices $\{j_0, j_1, \dots, j_5\}$ which lie on the convex hull of the $P(r)$ versus $R(r)$ characteristic.

for each $q = 1, 2, \dots, Q$. Equivalently, $r_q^{(\lambda)}$ is the value of r that maximizes the expression

$$P(r) - \lambda_q R(r), \quad (10)$$

where $\lambda_q = \lambda L_q / U_q$. This optimization problem arises in other contexts, such as the optimal truncation of embedded compressed bitstreams [13, Section 8.2]. It is known that the solution $r_q^{(\lambda)}$ must be a member of the set \mathcal{H}_C which describes the vertices of the convex hull of the $P(r)$ versus $R(r)$ characteristic [13, Section 8.2], as illustrated in Figure 5. Then, if $0 = j_0 < j_1 < \dots < j_I = N$ is an enumeration of the elements in \mathcal{H}_C , and

$$S_C(i) = \begin{cases} \frac{P(j_i) - P(j_{i-1})}{R(j_i) - R(j_{i-1})}, & i > 0, \\ \infty, & i = 0, \end{cases} \quad (11)$$

are the ‘‘slope’’ values on the convex hull, then $S_C(0) \geq S_C(1) \geq \dots \geq S_C(I)$. The solution to our optimization problem is obtained by finding the maximum value of $j_i \in \mathcal{H}_C$, which satisfies

$$P(j_i) - \lambda_q R(j_i) \geq P(j_{i-1}) - \lambda_q R(j_{i-1}). \quad (12)$$

Specifically,

$$\begin{aligned} r_q^{(\lambda)} &= \max \left\{ j_i \in \mathcal{H}_C \mid \frac{P(j_i) - P(j_{i-1})}{R(j_i) - R(j_{i-1})} \geq \lambda_q \right\} \\ &= \max \{ j_i \in \mathcal{H}_C \mid S_C(i) \geq \lambda_q \}. \end{aligned} \quad (13)$$

Given λ , the complexity of finding a set of optimal solutions $\{r_q^{(\lambda)}\}$ is $\mathcal{O}(IQ)$. Our algorithm first finds the largest λ such that $L^{(\lambda)} < L_{\max}$ and then employs a bisection search to find λ^{opt} , where $L^{(\lambda^{\text{opt}})} \approx L_{\max}$. The number of iteration required to search for λ^{opt} is bounded by the computation precision, and the bisection search algorithm typically requires a small number of iterations to find λ^{opt} . In our experiments, the number of iterations is typically fewer than 15, which is usually much smaller than I or Q . It is also worth noting that the

number of iterations required to find λ^{opt} is independent of other parameters, such as the number of source elements Q , the packet size S , and the codeword length N .

All that remains now is to show that this solution will always satisfy the necessary constraint $r_1 \geq r_2 \geq \dots \geq r_Q$. To this end, observe that our source convexity assumption implies that $L_q / U_q \leq L_{q+1} / U_{q+1}$ so that

$$\{j_i \in \mathcal{H}_C \mid S_C(i) \geq \lambda_q\} \supseteq \{j_i \in \mathcal{H}_C \mid S_C(i) \geq \lambda_{q+1}\}. \quad (14)$$

It follows that

$$\begin{aligned} r_q^{(\lambda)} &= \max \{j_i \in \mathcal{H}_C \mid S_C(i) \geq \lambda_q\} \\ &\geq \max \{j_i \in \mathcal{H}_C \mid S_C(i) \geq \lambda_{q+1}\} = r_{q+1}^{(\lambda)}, \quad \forall q. \end{aligned} \quad (15)$$

4.2. Nonconvex sources

In the previous section, we restricted our attention to convex source utility-length characteristics, but did not impose any prior assumption on the convexity of the $P(r)$ versus $R(r)$ channel coding characteristic. As already seen in Figure 3, the $P(r)$ versus $R(r)$ characteristic is not generally convex. We found that the optimal solution is always drawn from the convex hull set \mathcal{H}_C and that the optimization problem amounts to a trivial element-wise optimization problem in which $r_q^{(\lambda)}$ is assigned to the largest element $j_i \in \mathcal{H}_C$ whose slope $S_C(i)$ is no smaller than $\lambda L_q / U_q$.

In this section, we abandon our assumption on source convexity. We begin by showing that in this case, the optimal solution involves only those protection strengths r which belong to the convex hull \mathcal{H}_C of the channel code’s performance characteristic. We then show that the optimal protection assignment depends only on the convex hull of the source utility-length characteristic and that it may be found using the comparatively trivial methods previously described.

4.2.1. Sufficiency of the channel coding convex hull \mathcal{H}_C

Lemma 1. *Suppose that $\{r_q^{(\lambda)}\}_{1 \leq q \leq Q}$ is the collection of channel code indices which maximizes $J^{(\lambda)}$ subject to the ordering constraint $r_1^{(\lambda)} \geq r_2^{(\lambda)} \geq \dots \geq r_Q^{(\lambda)}$. Then $r_q^{(\lambda)} \in \mathcal{H}_C$ for all q . More precisely, whenever there is $r_q \notin \mathcal{H}_C$ yielding $\bar{J}(\lambda)$, there is always another $r_q \in \mathcal{H}_C$, which yield $J^{(\lambda)} \geq \bar{J}(\lambda)$.*

Proof. As before, let $0 = j_0 < j_1 < \dots < j_I$ be an enumeration of the elements in \mathcal{H}_C . For each $j_i \in \mathcal{H}_C$, define $\mathcal{F}_i = \{r_q^{(\lambda)} \mid j_i < r_q^{(\lambda)} < j_{i+1}\}$. For convenience, we define $j_{I+1} = \infty$ so that the last of these sets \mathcal{F}_I is well defined. The objective of the proof is to show that all of these sets \mathcal{F}_i must be empty. To this end, suppose that some \mathcal{F}_i is nonempty and let $\bar{r}_1 < \bar{r}_2 < \dots < \bar{r}_Z$ be an enumeration of its elements. For each $\bar{r}_z \in \mathcal{F}_i$, let \bar{U}_z and \bar{L}_z be the combined utilities and lengths of all source elements which were assigned $r_q^{(\lambda)} = \bar{r}_z$. That is,

$$\bar{U}_z = \sum_{q \ni r_q^{(\lambda)} = \bar{r}_z} U_q, \quad \bar{L}_z = \sum_{q \ni r_q^{(\lambda)} = \bar{r}_z} L_q. \quad (16)$$

For each $z < Z$, we could assign the alternate value of \bar{r}_{z+1} to all of the source elements with $r_q^{(\lambda)} = \bar{r}_z$ without violating the ordering constraint on $\bar{r}_q^{(\lambda)}$. This adjustment would result in a net increase in $J^{(\lambda)}$ of

$$\bar{U}_z [P(\bar{r}_{z+1}) - P(\bar{r}_z)] - \lambda \bar{L}_z [R(\bar{r}_{z+1}) - R(\bar{r}_z)]. \quad (17)$$

By hypothesis, we already have the optimal solution, so this alternative must be unfavourable, meaning that

$$\frac{P(\bar{r}_{z+1}) - P(\bar{r}_z)}{R(\bar{r}_{z+1}) - R(\bar{r}_z)} \leq \lambda \frac{\bar{L}_z}{\bar{U}_z}. \quad (18)$$

Similarly, for any $z \leq Z$, we could assign the alternate value of \bar{r}_{z-1} to the same source elements (where we identify \bar{r}_0 with j_i for completeness) again without violating our ordering constraint. The fact that the present solution is optimal means that

$$\frac{P(\bar{r}_z) - P(\bar{r}_{z-1})}{R(\bar{r}_z) - R(\bar{r}_{z-1})} \geq \lambda \frac{\bar{L}_z}{\bar{U}_z}. \quad (19)$$

Proceeding by induction, we must have monotonically decreasing slopes

$$\begin{aligned} \frac{P(\bar{r}_1) - P(j_i)}{R(\bar{r}_1) - R(j_i)} \\ \geq \frac{P(\bar{r}_2) - P(\bar{r}_1)}{R(\bar{r}_2) - R(\bar{r}_1)} \geq \dots \geq \frac{P(\bar{r}_Z) - P(\bar{r}_{Z-1})}{R(\bar{r}_Z) - R(\bar{r}_{Z-1})}. \end{aligned} \quad (20)$$

It is convenient, for the moment, to ignore the pathological case $i = I$. Now since $\bar{r}_z \notin \mathcal{H}_C$, we must have

$$\begin{aligned} \frac{P(j_{i+1}) - P(j_i)}{R(j_{i+1}) - R(j_i)} \\ \geq \frac{P(\bar{r}_1) - P(j_i)}{R(\bar{r}_1) - R(j_i)} \geq \dots \geq \frac{P(\bar{r}_Z) - P(\bar{r}_{Z-1})}{R(\bar{r}_Z) - R(\bar{r}_{Z-1})}, \end{aligned} \quad (21)$$

as illustrated in Figure 6. So, for any given $z \geq 1$, we must have

$$\frac{P(j_{i+1}) - P(\bar{r}_z)}{R(j_{i+1}) - R(\bar{r}_z)} \geq \frac{P(\bar{r}_z) - P(\bar{r}_{z-1})}{R(\bar{r}_z) - R(\bar{r}_{z-1})} \geq \lambda \frac{\bar{L}_z}{\bar{U}_z}, \quad (22)$$

meaning that all of the source elements which are currently assigned $r_q^{(\lambda)} = \bar{r}_z$ could be assigned $r_q^{(\lambda)} = j_{i+1}$ instead without decreasing the contribution of these source elements to $J^{(\lambda)}$. Doing this for all z simultaneously would not violate the ordering constraint, meaning that there is another solution, which is at least as good as the one claimed to be optimal, in which \mathcal{F}_i is empty.

For the case $i = I$, the fact that $\bar{r}_1 \notin \mathcal{H}_C$ and that there are no larger values of r which belong to the convex hull means that $(P(\bar{r}_1) - P(j_i))/(R(\bar{r}_1) - R(j_i)) \leq 0$ and hence $(P(\bar{r}_z) - P(\bar{r}_{z-1}))/R(\bar{r}_z) - R(\bar{r}_{z-1})) \leq 0$ for each z . But this contradicts (19) since $\lambda(\bar{L}_z/\bar{U}_z)$ is strictly positive. Therefore, \mathcal{F}_I is also empty. \square

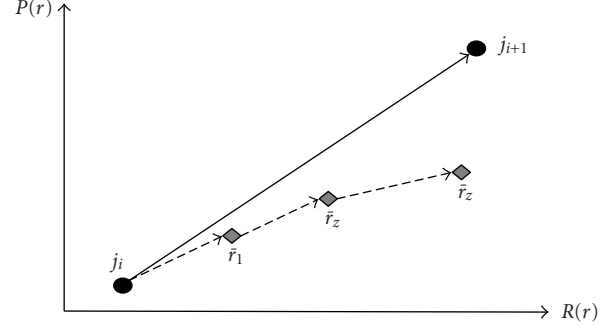


FIGURE 6: The parameters $\bar{r}_1, \dots, \bar{r}_Z$ between j_i and j_{i+1} are not part of convex hull points and have decreasing slopes.

4.2.2. Sufficiency of the source convex hull \mathcal{H}_S

In the previous section, we showed that we may restrict our attention to channel codes belonging to the convex hull set, that is, $r \in \mathcal{H}_C$, regardless of the source convexity. In this section, we show that we may also restrict our attention to the convex hull of the source utility-length characteristic.

Since the solution to our optimization problem satisfies $r_1^{(\lambda)} \geq r_2^{(\lambda)} \geq \dots \geq r_Q^{(\lambda)}$, it may equivalently be described in terms of a collection of thresholds $1 \leq t_i^{(\lambda)} \leq Q$ which we define according to

$$t_i^{(\lambda)} = \max \{q \mid r_q^{(\lambda)} \geq j_i\}, \quad (23)$$

where $0 = j_0 < j_1 < \dots < j_I = N$ is our enumeration of \mathcal{H}_C . For example, consider a source with $Q = 6$ elements and a channel code convex hull \mathcal{H}_C with $j_i \in \{0, 1, 2, \dots, 6\}$. Suppose that these elements are assigned

$$(r_1, r_2, \dots, r_6) = (5, 3, 2, 1, 1, 0). \quad (24)$$

Then, elements that are assigned at least $j_0 = 0$ correspond to all the six r 's and so $t_0 = 6$. Similarly, elements that are assigned at least $j_1 = 1$ correspond to the first five r 's and so, $t_1 = 5$. Performing the same computation as above for the remaining j_i produces

$$(t_0, t_1, \dots, t_6) = (6, 5, 3, 2, 1, 1, 0). \quad (25)$$

Evidently, the thresholds are ordered according to $Q = t_0^{(\lambda)} \geq t_1^{(\lambda)} \geq \dots \geq t_I^{(\lambda)}$. The $r_q^{(\lambda)}$ values may be recovered from this threshold description according to

$$r_q^{(\lambda)} = \max \{j_i \in \mathcal{H}_C \mid t_i^{(\lambda)} \geq q\}. \quad (26)$$

Using the same example above, given the channel code convex hull points $\{0, 1, 2, \dots, 6\}$ and a set of thresholds (25), possible threshold values for \mathcal{E}_1 are (t_0, t_1, \dots, t_5) and so, $r_1 = 5$. Similarly, possible threshold values for \mathcal{E}_2 are (t_0, \dots, t_3) and so, $r_2 = 3$. Performing the same computation as above for the remaining elements will produce the original code (24). Now, the unconstrained optimization problem from

(8) may be expressed as

$$\begin{aligned}
J^{(\lambda)} &= \sum_{q=1}^{t_1^{(\lambda)}} U_q P(j_1) - \lambda L_q R(j_1) \\
&\quad + \sum_{q=1}^{t_2^{(\lambda)}} U_q [P(j_2) - P(j_1)] - \lambda L_q [R(j_2) - R(j_1)] + \dots \\
&= \sum_{i=1}^I \underbrace{\left[\sum_{q=1}^{t_i^{(\lambda)}} U_q \dot{P}_i - \lambda L_q \dot{R}_i \right]}_{O_i^{(\lambda)}}, \tag{27}
\end{aligned}$$

where

$$\dot{P}_i \triangleq P(j_i) - P(j_{i-1}), \quad \dot{R}_i \triangleq R(j_i) - R(j_{i-1}). \tag{28}$$

If we temporarily ignore the constraint that the thresholds must be properly ordered according to $t_1^{(\lambda)} \geq t_2^{(\lambda)} \geq \dots \geq t_I^{(\lambda)}$, we may maximize $J^{(\lambda)}$ by maximizing each of the terms $O_i^{(\lambda)}$ separately. We will find that we are justified in doing this since the solution will always satisfy the threshold ordering constraint. Maximizing $O_i^{(\lambda)}$ is equivalent to finding $t_i^{(\lambda)}$, which maximize

$$\sum_{q=1}^{t_i^{(\lambda)}} U_q - \lambda_i L_q, \tag{29}$$

where $\lambda_i = \lambda \dot{R}_i / \dot{P}_i$. The same problem arises in connection with optimal truncation of embedded source codes¹ [13, Section 8.2]. It is known that the solutions $t_i^{(\lambda)}$ must be drawn from the convex hull set \mathcal{H}_S . Similar to \mathcal{H}_C , \mathcal{H}_S contains vertices lying on the convex hull curve of the utility-length characteristic. Let $0 = h_0 < h_1 < \dots < h_H = Q$ be an enumeration of the elements of \mathcal{H}_S and let

$$S_S(n) = \begin{cases} \frac{\sum_{q=h_{n-1}+1}^{h_n} U_q}{\sum_{q=h_{n-1}+1}^{h_n} L_q}, & n > 0, \\ \infty, & n = 0, \end{cases} \tag{30}$$

be the monotonically decreasing slopes associated with \mathcal{H}_S . Then

$$\begin{aligned}
t_i^{(\lambda)} &= \max \left\{ h_n \in \mathcal{H}_S \mid \sum_{q=h_{n-1}+1}^{h_n} U_q - \lambda_i L_q \geq 0 \right\} \\
&= \max \left\{ h_n \in \mathcal{H}_S \mid \frac{\sum_{q=h_{n-1}+1}^{h_n} U_q}{\sum_{q=h_{n-1}+1}^{h_n} L_q} \geq \lambda_i \right\} \\
&= \max \{ h_n \in \mathcal{H}_S \mid S_S(n) \geq \lambda_i \} \\
&= \max \{ h_n \in \mathcal{H}_S \mid S_S(n) \geq \lambda \dot{R}_i / \dot{P}_i \}. \tag{31}
\end{aligned}$$

¹In fact, this is the same problem as in Section 4.1 except that $P(r)$ and $R(r)$ are replaced with $\sum_{q=1}^r U_q$ and $\sum_{q=1}^r L_q$.

Finally, observe that

$$\frac{\dot{R}_i}{\dot{P}_i} = \frac{R(j_i) - R(j_{i-1})}{P(j_i) - P(j_{i-1})} = \frac{1}{S_C(i)}. \tag{32}$$

Monotonicity of the channel coding slopes $S_C(i)$ implies that $S_C(i) \geq S_C(i+1)$ and hence $\lambda/S_C(i) \leq \lambda/S_C(i+1)$. Then,

$$\begin{aligned}
&\{h_n \in \mathcal{H}_S \mid S_S(n) \geq \lambda/S_C(i)\} \\
&\supseteq \{h_n \in \mathcal{H}_S \mid S_S(n) \geq \lambda/S_C(i+1)\}. \tag{33}
\end{aligned}$$

It follows that

$$\begin{aligned}
t_i^{(\lambda)} &= \max \{ h_n \in \mathcal{H}_S \mid S_S(n) \geq \lambda/S_C(i) \} \\
&\geq \max \{ h_n \in \mathcal{H}_S \mid S_S(n) \geq \lambda/S_C(i+1) \} = t_{i+1}^{(\lambda)}. \tag{34}
\end{aligned}$$

Therefore, the required ordering property $t_1^{(\lambda)} \geq t_2^{(\lambda)} \geq \dots \geq t_I^{(\lambda)}$ is satisfied.

In summary, for each $j_i \in \mathcal{H}_C$, we find the threshold $t_i^{(\lambda)}$ from

$$t_i^{(\lambda)} = \max \{ h_n \in \mathcal{H}_S \mid S_S(n) \geq \lambda/S_C(i) \} \tag{35}$$

and then assign (26). The solution is guaranteed to be at least as good as any other channel code assignment, in the sense of maximizing $J^{(\lambda)}$ subject to $r_1^{(\lambda)} \geq r_2^{(\lambda)} \geq \dots \geq r_Q^{(\lambda)}$, regardless of the convexity of the source or channel codes. The computational complexity is now $\mathcal{O}(IH)$ for each λ . Similar to the convex sources case, we employ the bisection search algorithm to find λ^{opt} .

5. MULTICOP ASSIGNMENT

In the UniCOP assignment strategy, we assume that either the packet size S or the codeword length N can be set sufficiently large so that the data source can always fit into N packets. Specifically, the UniCOP assignment holds under the following condition:

$$\sum_{q=1}^Q L_q R(r_q) \leq NS. \tag{36}$$

Recall from Figure 1 that NS is the COP size.

The choice of the packet size depends on the type of channel that the data is transmitted through. Some channels might have low BERs allowing the use of large packet sizes with a reasonably high probability of receiving error-free packets. However, wireless channels typically require small packets due to their much higher BER. Packaging a large amount of source data into small packets requires a large number of packets and hence long codewords. This is undesirable since it imposes a computational burden on both the channel encoder and, especially, the channel decoder.

If the entire collection of protected source elements cannot fit into a set of N packets of length S , more than one COP must be employed. When elements are arranged into COPs, we no longer have any guarantee that a source element

with a stronger code can be recovered whenever a source element with a weaker code is recovered. The code redundancy assignment strategy described in Section 4 relies upon this property in order to ensure that element dependencies are satisfied, allowing us to use (1) for the expected utility.

5.1. Code redundancy optimization

Consider a collection of C COPs $\{\mathcal{C}_1, \dots, \mathcal{C}_C\}$ characterized by $\{(s_1, f_1), \dots, (s_C, f_C)\}$, where s_c and f_c represent the indices of the first and the last source elements residing in the COP \mathcal{C}_c . We assume that the source elements have a simple chain of dependencies $\mathcal{E}_1 < \mathcal{E}_2 < \dots < \mathcal{E}_Q$ such that prior to recovering an element \mathcal{E}_q , all preceding elements $\mathcal{E}_1, \dots, \mathcal{E}_{q-1}$ must be recovered first. Within each COP \mathcal{C}_i , we can still constrain the code redundancies to satisfy

$$r_{s_i} \geq r_{s_i+1} \geq \dots \geq r_{f_i} \quad (37)$$

and guarantee that no element in COP \mathcal{C}_i will be recovered unless all of its dependencies within the same COP are also recovered. The probability $P(r_{f_i})$ of recovering the last element \mathcal{E}_{f_i} thus denotes the probability that all elements in COP \mathcal{C}_i are recovered successfully. Therefore, any element \mathcal{E}_q in COP \mathcal{C}_c , which is correctly recovered from the channel, will be usable if and only if the last element of each earlier COP is recovered. This changes the expected utility in (1) to

$$U = U_0 + \sum_{c=1}^C \sum_{q=s_c}^{f_c} \left(U_q P(r_q) \prod_{i=1}^{c-1} P(r_{f_i}) \right). \quad (38)$$

Our objective is to maximize this expression for U subject to the same total length constraint L_{\max} , as given in (7), and subject also to the constraint that

$$r_{s_c} \geq r_{s_c+1} \geq \dots \geq r_{f_c} \quad (39)$$

for each COP \mathcal{C}_c . Similar to the UniCOP assignment strategy, this constrained optimization problem can be converted into a set of unconstrained optimization problems parametrized by λ . Specifically, we search for the smallest λ such that $L^{(\lambda)} \leq L_{\max}$, where $L^{(\lambda)}$ is the overall transmission length associated with the set $\{r_q^{(\lambda)}\}_{1 \leq q \leq Q}$, which maximizes

$$\begin{aligned} J^{(\lambda)} &= U^{(\lambda)} - \lambda L^{(\lambda)} \\ &= \sum_{c=1}^C \sum_{q=s_c}^{f_c} U_q P(r_q^{(\lambda)}) \prod_{i=1}^{c-1} P(r_{f_i}^{(\lambda)}) - \lambda L_q R(r_q^{(\lambda)}) \end{aligned} \quad (40)$$

subject to the constraint $r_{s_c} \geq r_{s_c+1} \geq \dots \geq r_{f_c}$ for all c . This new functional turns out to be more difficult to optimize than that in (8) since the product terms in $U^{(\lambda)}$ couple the impact of code redundancy assignments for different elements. In fact, the optimization objective is generally multimodal exhibiting multiple local optima.

Nevertheless, it is possible to devise a simple optimization strategy, which rapidly converges to a local optimum, with good results in practice. Specifically, given an initial set of $\{r_q\}_{1 \leq q \leq Q}$ and considering only one COP, \mathcal{C}_c , at a time,

we can find a set of code redundancies $\{r_{s_c}, \dots, r_{f_c}\}$ which maximizes $J^{(\lambda)}$ subject to all other r_q 's being held constant. The solution is sensitive to the initial $\{r_q\}$ set since the optimization problem is multimodal. However, as we shall see shortly in Section 5.2, since we build multiple COPs out of one COP, it is reasonable to set the initial values of $\{r_q\}$ equal to those obtained from the UniCOP assignment of Section 4. The UniCOP assignment works under the assumption that all encoded source elements can fit into one COP. This algorithm is guaranteed to converge as we cycle through each COP in turn, since the code redundancies for each COP either increase $J^{(\lambda)}$ or leave it unchanged, and the optimization objective is clearly bounded above by $\sum_q U_q$. The optimal solution for each COP is found by employing the scheme developed in Section 4. Our optimization objective for each COP \mathcal{C}_c is to maximize a quantity

$$\begin{aligned} J_c^{(\lambda)} &= \sum_{q=s_c}^{f_c} U_q \left[\prod_{i=1}^{c-1} P(r_{f_i}^{(\lambda)}) \right] P(r_q^{(\lambda)}) \\ &\quad - \lambda L_q R(r_q^{(\lambda)}) + P(r_{f_c}^{(\lambda)}) \Gamma_c \end{aligned} \quad (41)$$

while keeping code redundancies in other COPs constant. The last element \mathcal{E}_{f_c} in COP \mathcal{C}_c is unique since its recovery probability appears in the utility term of succeeding elements $\mathcal{E}_{f_c+1}, \dots, \mathcal{E}_Q$ which reside in COPs $\mathcal{C}_{c+1}, \dots, \mathcal{C}_C$. This effect is captured by the term

$$\Gamma_c = \sum_{m=c+1}^C \sum_{n=s_m}^{f_m} U_n P(r_n^{(\lambda)}) \prod_{i=1, i \neq c}^{m-1} P(r_{f_i}^{(\lambda)}); \quad (42)$$

Γ_c can be considered as an additional contribution to the effective utility of \mathcal{E}_{f_c} .

Evidently, Γ_c is nonnegative, so it will always increase the effective utility of the last element in any COP \mathcal{C}_c , $c < C$. Even if the original source elements have a convex utility-length characteristic such that

$$\frac{U_{s_c}}{L_{s_c}} \geq \frac{U_{s_c+1}}{L_{s_c+1}} \geq \dots \geq \frac{U_{f_c}}{L_{f_c}}, \quad (43)$$

the optimization of $J_c^{(\lambda)}$ subject to $r_{s_c} \geq r_{s_c+1} \geq \dots \geq r_{f_c}$ involves the effective utilities

$$U'_q = \begin{cases} U_q \prod_{i=1}^{c-1} P(r_{f_i}^{(\lambda)}), & q = s_c, \dots, f_c - 1, \\ U_q \prod_{i=1}^{c-1} P(r_{f_i}^{(\lambda)}) + \Gamma_c, & q = f_c. \end{cases} \quad (44)$$

Apart from the last element $q = f_c$, U'_q is a scaled version of U_q involving the same scaling factor $\prod_{i=1}^{c-1} P(r_{f_i}^{(\lambda)})$ for each q . However, the last element \mathcal{E}_{f_c} has an additional utility Γ_c which can destroy the convexity of the source effective utility-length characteristic. This phenomenon forms the principle motivation for the development in Section 4 of code redundancy assignment strategy, which is free from the assumption of convexity on the source or channel code characteristic.

In summary, the code redundancy assignment strategy for multiple COPs involves cycling through the COPs one at a time, holding the code redundancy for all COPs constant, and finding the values of $r_q^{(\lambda)}$, $s_c \leq q \leq f_c$, which maximize

$$J_c^{(\lambda)} = \sum_{q=s_c}^{f_c} U'_q P(r_q^{(\lambda)}) - \lambda L_q R(r_q^{(\lambda)}) \quad (45)$$

subject to the constraint $r_{s_c} \geq \dots \geq r_{f_c}$. Maximization of $J_c^{(\lambda)}$ subject to $r_{s_c} \geq \dots \geq r_{f_c}$, is achieved by using the strategy developed in Section 4, replacing each element's utility U_q with its current effective utility U'_q . Specifically, for each COP \mathcal{C}_c , we find a set of $\{t_i^{(\lambda)}\}$ which must be drawn from the convex hull set $\mathcal{H}_S^{(c)}$ of the source effective utility-length characteristic. Since U'_{f_c} is affected by $\{r_{f_c+1}, \dots, r_Q\}$, elements in $\mathcal{H}_S^{(c)}$ may vary depending on these code redundancies and thus must be recomputed at each iteration of the algorithm. Then,

$$t_i^{(\lambda)} = \max \{h_n \in \mathcal{H}_S^{(c)} \mid S_S(n) \geq \lambda / S_C(i)\}, \quad (46)$$

where

$$S_S(n) = \begin{cases} \frac{\sum_{q=h_{n-1}+1}^{h_n} U'_q}{\sum_{q=h_{n-1}+1}^{h_n} L_q}, & n > 0 \\ \infty, & n = 0. \end{cases} \quad (47)$$

The solution $r_q^{(\lambda)}$ may be recovered from $t_i^{(\lambda)}$ using (26). As in the UniCOP case, we find the smallest value of λ such that the resulting solution satisfies $L^{(\lambda)} \leq L_{\max}$. Similar to the UniCOP assignment of nonconvex sources, for each COP \mathcal{C}_c , the computation complexity is $\mathcal{O}(IH_c)$, where H_c is the number of elements in $\mathcal{H}_S^{(c)}$. Hence, in each iteration, it requires $\mathcal{O}(IH)$ computations, where $H = \sum_{c=1}^C H_c$. For some $\lambda > 0$, it typically requires fewer than 10 iterations for the solution to converge.

5.2. COP allocation algorithm

We are still left with the problem of determining the best allocation of elements to COPs subject to the constraint that the encoded source elements in any given COP should be no larger than NS . When L_{\max} is larger than NS , the need to use multiple COPs is inevitable. The proposed algorithm starts by allocating all source elements to a single COP \mathcal{C}_1 . Code redundancies are found by applying the UniCOP assignment strategy of Section 4. COP \mathcal{C}_1 is then split into two parts, the first of which contains as many elements as possible (f_1 as large as possible) while still having an encoded length $L_{\mathcal{C}_1}$ no larger than NS . At this point, the number of COPs is $C = 2$ and \mathcal{C}_2 does not generally satisfy $L_{\mathcal{C}_2} \leq NS$.

The algorithm proceeds in an iterative sequence of steps. At the start of the t th step, there are C_t COPs, all but the last of which have encoded lengths no larger than NS . In this step, we first apply the MultiCOP code redundancy assignment algorithm of Section 5.1 to find a new set of $\{r_{s_c}, \dots, r_{f_c}\}$ for each COP \mathcal{C}_c maximizing the total expected utility subject to

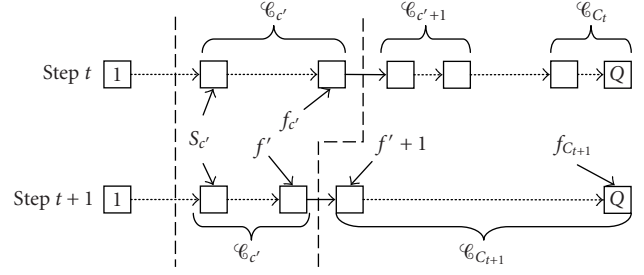


FIGURE 7: Case 1 of the COP allocation algorithm. At step t , $L_{\mathcal{C}_{c'}}$ exceeds NS and hence is truncated. Its trailing elements and the rest of source elements are allocated to one COP, $\mathcal{C}_{c_{t+1}}$.

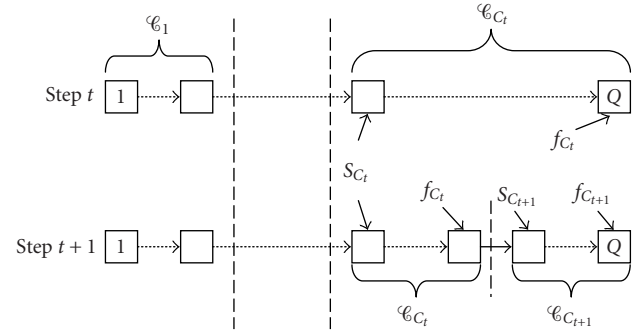


FIGURE 8: Case 2 of the COP allocation algorithm. At step t , the last COP is divided into two, the first of which, \mathcal{C}_{c_t} , satisfies NS .

the overall length constraint L_{\max} . The new code redundancies produced by the MultiCOP assignment algorithm may cause one or more of the initial $C_t - 1$ COPs to violate the encoded length constraint of $L_{\mathcal{C}_c} \leq NS$. In fact, as the algorithm proceeds, the encoded lengths of source elements assigned to all but the last COP tend to increase rather than decrease, as we shall argue later. The step is completed in one of two ways depending on whether or not this happens.

Case 1 ($L_{\mathcal{C}_{c'}} > NS$ for some $c < C_t$). Let $\mathcal{C}_{c'}$ be the first COP for which $L_{\mathcal{C}_{c'}} > NS$. In this case, we find the largest value of $f' \geq s_{c'}$ such that $\sum_{q=s_{c'}}^{f'} L_q R(r_q) \leq NS$. COP $\mathcal{C}_{c'}$ is truncated by setting $f_{c'} = f'$ and all of the remaining source elements $\mathcal{C}_{f'+1}, \mathcal{C}_{f'+2}, \dots, \mathcal{C}_Q$ are allocated to $\mathcal{C}_{c'+1}$. The algorithm proceeds in the next step with only $C_{t+1} = c' + 1 \leq C_t$ COPs, all but the last of which satisfy the length constraint. Figure 7 illustrates this case.

Case 2 ($L_{\mathcal{C}_c} \leq NS$, for all $c < C_t$). In this case, we find the largest value of $f \geq s_{C_t}$, in order to satisfy $\sum_{q=s_{C_t}}^f L_q R(r_q) \leq NS$, setting $f_{C_t} = f$. If $f = Q$, all source elements are already allocated to COPs, satisfying the length constraint, and their code redundancies are already jointly optimized, so we are done. Otherwise, the algorithm proceeds in the next step with $C_{t+1} = C_t + 1$ COPs, where $\mathcal{C}_{C_{t+1}}$ contains all of the remaining source elements $\mathcal{C}_{f+1}, \mathcal{C}_{f+2}, \dots, \mathcal{C}_Q$. Figure 8 demonstrates this case.

To show that the algorithm must complete after a finite number of steps, observe first that the number of COPs must be bounded above by some quantity $M \leq Q$. Next, define an integer-valued functional

$$Z_t = \sum_{c=1}^{C_t} \|\mathcal{C}_c^{(t)}\| Q^{(M-c)}, \quad (48)$$

where $\|\mathcal{C}_c^{(t)}\|$ denotes the number of source elements allocated to COP \mathcal{C}_c at the beginning of step t . This functional has the important property that each step in the allocation algorithm decreases Z_t . Since Z_t is always a positive finite integer, the algorithm must therefore complete in a finite number of steps. To see that each step does indeed decrease Z_t , consider the two cases. If step t falls into Case 1, with $\mathcal{C}_{c'}$ the COP whose contents are reduced, we have

$$\begin{aligned} Z_{t+1} &= \sum_{c=1}^{c'-1} \|\mathcal{C}_c^{(t)}\| Q^{(M-c)} + (f+1 - s_{c'}^{(t)}) Q^{(M-c')} \\ &\quad + (Q-f) Q^{(M-c'-1)} \\ &= \sum_{c=1}^{c'} \|\mathcal{C}_c^{(t)}\| Q^{(M-c)} + (Q-f_{c'}^{(t)}) Q^{(M-c'-1)} \\ &\quad - (f_{c'}^{(t)} - f) (Q^{(M-c')} - Q^{(M-c'-1)}) \\ &< \sum_{c=1}^{C_t} \|\mathcal{C}_c^{(t)}\| Q^{(M-c)} + (Q-2) \cdot Q^{M-c'-1} \\ &\quad - (Q^{(M-c')} - Q^{(M-c'-1)}) \\ &< \sum_{c=1}^{C_t} \|\mathcal{C}_c^{(t)}\| Q^{(M-c)} = Z_t. \end{aligned} \quad (49)$$

If step t falls into Case 2, some of the source elements are moved from \mathcal{C}_{C_t} to \mathcal{C}_{C_t+1} , where their contribution to Z_t is reduced by a factor of Q , so $Z_{t+1} < Z_t$.

The key property of our proposed COP allocation algorithm, which ensures its convergence, is that whenever a step does not split the final COP, it necessarily decreases the number of source elements contained in a previous COP $\mathcal{C}_{c'}$. The algorithm contains no provision for subsequently reconsidering this decision and moving some or all of these elements back into $\mathcal{C}_{c'}$. We claim that there is no need to revisit the decision to move elements out of $\mathcal{C}_{c'}$ for the following reason. Assuming that we do not alter the contents of any previous COPs (otherwise, the algorithm essentially restarts from that earlier COP boundary), by the time the allocation is completed, all source elements following the last element in $\mathcal{C}_{c'}$ should be allocated to COPs \mathcal{C}_c , with indices c at least as large as they were in step t . Considering (44), the effective utilities of these source elements will tend to be reduced relative to the effective utilities of the source elements allocated to \mathcal{C}_1 through $\mathcal{C}_{c'}$. Accordingly, one should expect the source elements allocated to \mathcal{C}_1 through $\mathcal{C}_{c'}$ to receive a larger share of the overall length budget L_{\max} , meaning that their coded lengths should be at least as large as they were in step t . While this is not a rigorous proof of optimality, it provides a strong justification for the proposed allocation algorithm.

In practice, as the algorithm proceeds, we always observe that the code redundancies assigned to source elements in earlier COPs either remain unchanged or else increase.

5.3. Remarks on the effect of packet size

Throughout this paper, we have assumed that $P(r_q)$, the probability of receiving sufficient packets to decode an element \mathcal{E}_q , depends only on the selection of $r_q = N - k_q + 1$, where an (N, k_q) channel code is used. The value of N is fixed, but as discussed in Section 3, the value of $P(r_q)$ also depends upon the actual size of each encoded packet. We have taken this to be S , but our code redundancy assignment and COP allocation algorithms use S only as an upper bound for the packet size. If the maximum NS bytes are not used by any COP, each packet may actually be smaller than S . This, in turn, may alter the values of $P(r_q)$ so that our assigned codes are no longer optimal.

Fortunately, if the individual source elements are sufficiently small, the actual size of each COP should be approximately equal to its maximum value of NS , meaning that the actual packet size should be close to its maximum value of S . It is true that allocating more COPs, each with a smaller packet size, can yield higher expected utilities. However, rather than explicitly accounting for the effect of actual packet sizes within our optimization algorithm, various values for S are considered in an ‘‘outer optimization loop.’’ In particular, for each value of S , we compute the channel coding characteristic described by $P(r_q)$ and $R(r_q)$ and then invoke our COP allocation and code redundancy optimization algorithm. Section 6 presents expected utility results obtained for various values of S . One potential limitation of this strategy is that the packet size S is essentially being forced to take on the same value within each COP. We have not considered the possibility of allowing different packet sizes or even different channel code lengths N for each COP.

6. COMPARATIVE RESULTS

In this section, we compare the total expected utility of a compressed image at the destination whose code redundancies have been determined using the UniCOP and MultiCOP assignment strategies described in Sections 4 and 5. We select a code length $N = 100$, a maximum transmission length $L_{\max} = 1\,000\,000$ bytes, a range of BER ϵ , and packet sizes S . The scalable data source used in these experiments is a 2560×2048 JPEG2000 compressed image, decomposed into 6 resolution levels. The image is grayscale exhibiting only one colour component and we treat the entire image as one tile component. Each resolution level is divided into a collection of precincts with size 128×128 samples, resulting in a total of 429 precincts. Each precinct is further decomposed into 12 quality elements. Overall, there are 5149 elements, treating each quality element and the data stream header as a source element. It is necessary to create a large number of source elements so as to minimize the impact of the discrete nature of our optimization problem, which may otherwise produce suboptimal solutions as discussed in Section 4.1.

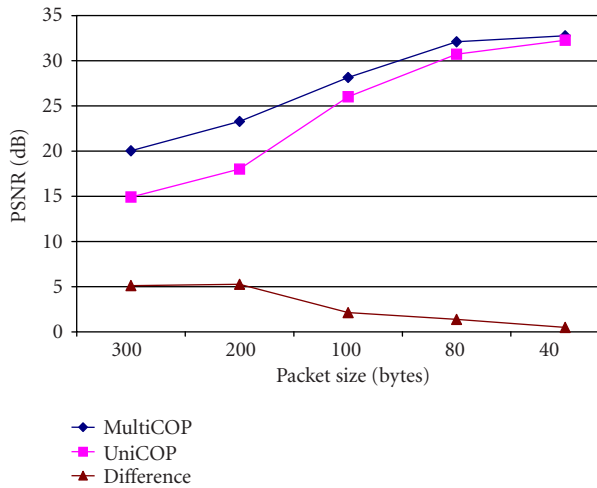


FIGURE 9: Comparative performance between UniCOP and MultiCOP assignment strategies with $\epsilon = 10^{-3}$.

We arrange the source elements into a linear sequence exhibiting a simple chain of dependencies with a convex utility-length characteristic. For simplicity, we assume that in the event, where any part of any element is corrupted, the entire element will be rendered useless along with all subsequent elements which depend upon it.² The UniCOP results were obtained by using the UniCOP assignment under the assumption that all source elements can be arranged into one COP. The encoded elements are then assigned to multiple COPs, whenever this is demanded by the constraint NS . The utility measure used here is a negated MSE, and the total expected utility is conveniently expressed in terms of peak signal-to-noise ratio (PSNR).³ An improvement in the expected utility is equivalent to an increase in PSNR. To get a reasonable approximation of the total expected utility for each value of the packet size parameter S , the number of experiments which we run to find the overall expected utility is adjusted according to the packet loss probability. The MultiCOP results were based on the MultiCOP assignment algorithm, which progressively allocates source elements to COPs and assigns code redundancies to source elements accordingly. Figures 9, 10, and 11 compare the UniCOP results with those obtained by using the MultiCOP assignment strategy.

If all of the coded source elements are able to fit inside a single COP subject to the constraints determined by N and S , the UniCOP assignment will be optimal. Moreover, in this case, the UniCOP and MultiCOP strategies produce identical solutions. Otherwise, source elements must be assigned to multiple COPs, violating some of the assumptions underlying the UniCOP assignment strategy. In particular, the

²In practice, this assumption is excessively conservative since JPEG2000 decoders are able to recover well from some types of error.

³Peak signal-to-noise ratio is defined as $10 \log(P^2/MSE)$, where P is the peak-to-peak signal amplitude. In this case, $P = 255$ since we are working with 8-bit images.

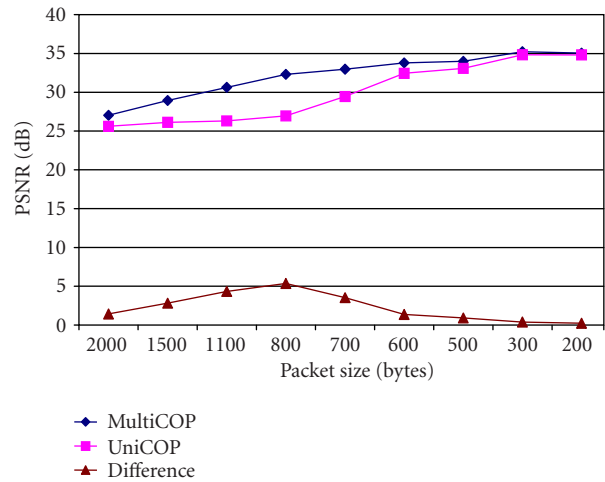


FIGURE 10: Comparative performance between UniCOP and MultiCOP assignment strategies with $\epsilon = 10^{-4}$.

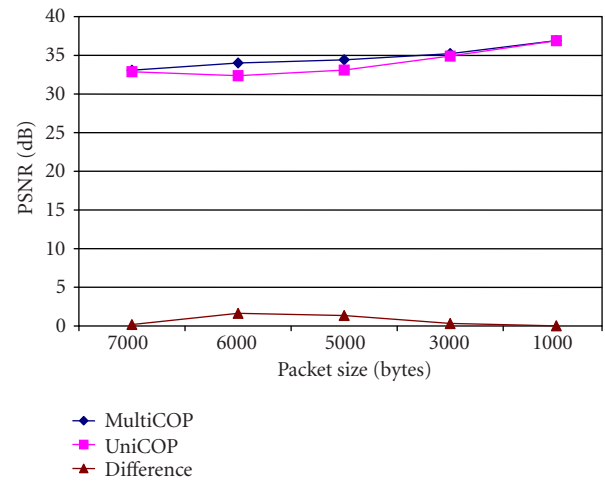


FIGURE 11: Comparative performance between UniCOP and MultiCOP assignment strategies with $\epsilon = 10^{-5}$.

recovery of any element \mathcal{E}_q no longer guarantees the recovery of all the preceding elements $\mathcal{E}_1, \dots, \mathcal{E}_{q-1}$ whose code redundancies are at least equal to that of \mathcal{E}_q . In this case, we would expect to see the MultiCOP assignment strategy providing superior performance. Figures 9, 10, and 11 show that both UniCOP and MultiCOP assignment strategies produce higher PSNR when the packet sizes are small. This is due to the fact that for a given BER, the packet loss probability decreases with decreasing the packet size. Low packet loss probability allows the elements to be assigned with weak codes and hence to be encoded with lower amount of redundancy. Therefore, it is possible to transmit more encoded elements without exceeding the maximum length L_{\max} .

The PSNR values from the MultiCOP assignment strategy are always higher relative to the UniCOP case for a given packet size and BER. The MultiCOP assignment process

described in Section 5 increases the expected utilities of the elements in the earlier COPs relative to those of the elements in the later COPs. This causes the same or a stronger code to be assigned to the elements in the earlier COPs. As a result, the elements in the earlier COPs are not corrupted as easily by packet loss.

The improvement in the PSNR for the MultiCOP assignment also depends on the BER ϵ . At high BER, the difference in PSNR could reach above 5 dB, but at low BER, the difference is at most 2 dB. The main reason for this is that the high BER, which requires the use of small packet sizes, produces a large number of COPs. Therefore, the code redundancies produced by the MultiCOP assignment differ much from those produced by the UniCOP assignment at high error rates. Accordingly, the MultiCOP assignment strategy is particularly very appealing for channels with high error rates, such as wireless channels.

Finally, the results show that for any given BER ϵ , the improvement in PSNR diminishes as the packet size decreases. This is due to the fact that fewer packets are lost since a decrease in packet size reduces packet loss probability. In turn, the likelihood of recovering source elements for both MultiCOP and UniCOP assignment strategies increases, resulting in similar overall expected utility. Of course, applications do not generally have the freedom of selecting packet sizes. The use of small packets increases the amount of the packet overhead, a fact which is not taken into account in the results presented here.

7. CONCLUSIONS

Although PET provides an excellent framework for optimal protection of scalable data sources against erasure, it suffers from a difficulty that all channel codes must span the entire collection of network packets. In many practical applications, the size of the data source is large and packet sizes must be relatively small, leading to the need for long and computationally demanding channel codes. Two solutions to this problem present themselves immediately. Small network packets can be concatenated forming larger packets, thereby reducing the codeword length of the channel codes. Unfortunately, an erasure channel model is required such that the larger packets must be considered lost if any of their constituent packets are lost. Clearly, this solution is unsuitable for channels with significant packet loss probability.

As an alternative, the code redundancy assignment optimized for the PET framework could be used with shorter channel codes representing smaller COPs. When data must be divided up into independently coded COPs with shorter channel codes, the MultiCOP assignment strategy proposed in this paper provides significant improvements in the expected utility (PSNR). Nevertheless, the need to use multiple COPs imposes a penalty of its own. One drawback of the multiple COP assignment strategy is the amount of computation required to determine optimal code redundancies at the transmitter. This is particularly significant when there are large numbers of source elements and/or the COP size

is small. It is reasonable to expect exactly these conditions when transmitting a large compressed image over a wireless network. The development of fast algorithms for finding the MultiCOP assignment remains an active topic of investigation.

Including the codeword length as a parameter in the code redundancy assignment problem allows for flexibility in the choice of channel coding complexity. Since the channel decoder is generally more complex than the channel encoder, selecting short codewords will ease the computational burden at the receiver. This is particularly important for wireless mobile devices which have tight power and hence computation constraints.

REFERENCES

- [1] J. M. Shapiro, "An embedded hierarchical image coder using zerotrees of wavelet coefficients," in *Proc. IEEE Data Compression Conference*, pp. 214–223, Snowbird, Utah, USA, March–April 1993.
- [2] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, 1996.
- [3] ISO/IEC 15444-1, "JPEG2000 image coding system," 2000.
- [4] W. Li, "Overview of fine granular scalability in MPEG-4 video standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 301–317, 2001.
- [5] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1737–1744, 1996.
- [6] R. Puri and K. Ramchandran, "Multiple description source coding using forward error correction codes," in *Proc. 33rd Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 342–346, Pacific Grove, Calif, USA, October 1999.
- [7] A. Mohr, E. Riskin, and R. Ladner, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 819–828, 2000.
- [8] A. Mohr, R. Ladner, and E. Riskin, "Approximately optimal assignment for unequal loss protection," in *Proc. IEEE International Conference on Image Processing*, vol. 1, pp. 367–370, BC, Canada, September 2000.
- [9] T. Stockhammer and C. Buchner, "Progressive texture video streaming for lossy packet networks," in *Proc. 11th International Packet Video Workshop*, Kyongju, Korea, May 2001.
- [10] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams," in *Proc. IEEE Data Compression Conference*, pp. 73–82, Snowbird, Utah, USA, April 2002.
- [11] V. Stankovic, R. Hamzaoui, and Z. Xiong, "Packet loss protection of embedded data with fast local search," in *Proc. IEEE International Conference on Image Processing*, vol. 2, pp. 165–168, Rochester, NY, USA, September 2002.
- [12] J. Thie and D. Taubman, "Optimal protection assignment for scalable compressed images," in *Proc. IEEE International Conference on Image Processing*, vol. 3, pp. 713–716, Rochester, NY, USA, September 2002.
- [13] D. Taubman and M. Marcellin, Eds., *JPEG2000: Image Compression Fundamentals, Standards and Practice*, vol. 642, Kluwer Academic Publishers, Boston, Mass, USA, 2001.

Johnson Thie received his B.E. degree in electrical engineering and his Master of Biomedical Engineering from the University of New South Wales, Australia, in 2000. He is currently a Ph.D. student in the School of Electrical Engineering & Telecommunications at the same university. Mr. Thie spent two summers (1996–1998) at the St George Hospital, Sydney, for industrial training. During the summer of 1998–1999, he worked at the Centre of Telecommunications and Industrial Physics, a division of Commonwealth Scientific & Industrial Research Organization (CSIRO), Australia. His research interests are distribution of compressed media over packet-based networks and general applications of signal/image processing in medicine.



David Taubman received his B.S. degree in computer science and mathematics and the B.E. degree in electrical engineering in 1986 and 1988 from the University of Sydney, and his M.S. and Ph.D. degrees in electrical engineering in 1992 and 1994 from the University of California, Berkeley. From 1994 to 1998, he worked at Hewlett-Packard's Research Laboratories in Palo Alto, California. He joined the University of New South



Wales in 1998 as a Senior Lecturer in the School of Electrical Engineering and Telecommunications. Dr. Taubman was awarded the University Medal from the University of Sydney, as well as the Institution of Engineers, Australia Prize, and the Texas Instruments Prize for digital signal processing, all in 1998. He has received two Best Paper Awards from the IEEE Circuits and Systems Society in 1997 for the 1996 paper, "A common framework for rate and distortion based scaling of highly scalable compressed video," and from the IEEE Signal Processing Society in 2002 for the 2000 paper, "High performance scalable image compression with EBCOT." Dr. Taubman's research interests include highly scalable image and video compression, inverse problems in imaging, perceptual modeling, and multimedia distribution. He currently serves as an Associate Editor for the IEEE Transactions on Image Processing.