Priority Encoding Transmission

Andres Albanese* Johannes Blömer† Jeff Edmonds‡
Michael Luby§ Madhu Sudan¶

Abstract

We introduce a new method, called Priority Encoding Transmission, for sending messages over lossy packet-based networks. When a message is to be transmitted, the user specifies a priority value for each part of the message. Based on the priorities, the system encodes the message into packets for transmission and sends them to (possibly multiple) receivers. The priority value of each part of the message determines the fraction of encoding packets sufficient to recover that part. Thus, even if some of the encoding packets are lost enroute, each receiver is still able to recover the parts of the message for which a sufficient fraction of the encoding packets are received.

*International Computer Science Institute, Berkeley, California. Research supported in part by National Science Foundation operating grant NCR-941610

†Computer Science Department, Swiss Federal Institute of Technology, Zurich, Switzerland. Research done while a postdoc at the International Computer Science Institute, Berkeley, California. Partially supported by NSF operating grant CCR-9304722, ESPRIT BR Grant EC-US 030 and by Deutsche Forschungsgemeinschaft (DFG), Habilitationsstipendium Bl 314/4-1.

‡Department of Computer Science, York University, North York, Ontario, Canada. Research done while a postdoc at the International Computer Science Institute, Berkeley, California. Supported by an NSF postdoctoral fellowship and by a Canadian NSERC postdoctoral fellowship.

§International Computer Science Institute, Berkeley, California and Computer Science Division, University of California at Berkeley. Research supported in part by National Science Foundation operating grants CCR-9304722 and NCR-9416101, United States-Israel Binational Science Foundation grant No. 92-00226, and ESPRIT BR Grant EC-US 030.

¶IBM Thomas J. Watson Research Center, Yorktown Heights, New York. Participated in research while visiting the International Computer Science Institute, Berkeley, California.

For any set of priorities for a message, we define a natural quantity called the *girth* of the priorities. We develop systems for implementing any given set of priorities such that the total length of the encoding packets is equal to the girth. On the other hand, we give an information-theoretic lower bound that shows that for any set of priorities the total length of the encoding packets must be at least the girth. Thus, the system we introduce is optimal in terms of the total encoding length.

This work has immediate applications to multi-media and high speed networks applications, especially in those with bursty sources and multiple receivers with heterogeneous capabilities. Implementations of the system show promise of being practical.

1 Introduction

In many multi-media applications, long messages are to be transmitted in real-time across multiple network links. A message is not sent as one unit, but broken into packets that are sent through the medium. Bit corruption may occur in packets due to transmission, but these can be handled on a link-by-link basis using error correcting techniques. Thus, we can assume that packets are indivisible units that arrive intact if they arrive at all. Once the packets are sent, some of the packets may arrive promptly, but arbitrary subsets of packets may be lost or delayed beyond the point of usefulness due to global conditions in the network such as congestion, buffer overflows and other causes. We hereafter call media with this property *lossy* media. At some point in time, the receiver cannot wait for packets any longer and must recover as much of the original message as possible from the packets received.

It seems highly plausible that packet loss as described will be an ordinary phenomena for reasonably priced networks that connect millions of users spread around the world simultaneously running a multitude of high bandwidth real-time applications. Furthermore, packet losses will not be spread uniformly over the network, but may vary between different sites and may fluctuate over time. Thus, it could be argued that, analogous to noise being the nemesis of analog communication, and error being the nemesis of digital communication, loss will be the nemesis of packet-based wide-area real-time communication.

This paper proposes a general and flexible method to cope with packet loss, which we call Priority Encoding Transmission (PET). The user partitions the message into segments and assigns each segment a priority value. Based on their priority values, the segments are encoded into a set of packets. The priority value of a segment specifies the fraction of packets sufficient to decode it. The system guarantees that a segment can be decoded from any subset of packets as long as the fraction of packets in the subset is at least equal to the segment priority value.

In the networking community encoding systems which allow recovery of the message from only a subset of packets of the encoding have been proposed, for example a system based on Reed-Solomon-codes was suggested by [16, McAuley] and empirically evaluated by [8, Biersack]. A similar encoding system has been proposed by [16, Rabin]. He uses essentially the same coding techniques that are used in this paper. However, these systems allow only one priority level for the entire message.

[17, Shacham] also suggests methods for sending prioritized messages over networks. The basic idea is to partition the message into different priority levels and then use a different channel to send each level. Then, each receiver attaches to as many of these channels as possible, in order of their priority, up to the channel capacity between the sender and that particular receiver. However, this method requires computation of channel capacities from the sender to each receiver, which may be impractical for large networks with capacities that vary quickly because of congestion. Furthermore, this work does not handle packet losses.

Section 2 describes potential applications of the PET system to transmit multicast video images over heterogeneous lossy networks. Section 3 gives the formal requirements of both deterministic and probabilistic PET systems. A PET system is described in Section 4. We also review erasure-resilient codes in this section, as these are one of the main building blocks of our constructions. Section 5 gives an information-theoretic lower bound proof on the total length of the encoding packets produced by a PET system. Section 6 describes a (weak) lower bound on the packet length for any PET system.

Little effort is made to make the systems efficient. [9, Blömer et al.] describes an efficient implementation of the main building block of any PET system. A preliminary version of this paper appeared in [1].

2 Video Multicasting

Information from the sender must be received by all users participating in the multicast session. Present applications use protocols that retransmit missing information when communicating with multiple receivers. Consequently, the information rate is determined by the worst case receiver. Thus, there are difficulties when these protocols operate using lossy networks.

Priority Encoding Transmission is especially suited to implementing multicast protocols on lossy networks. For example, consider a straightforward video conferencing multicast protocol using either JPEG or MPEG. It turns out that the quality of the displayed video degrades rapidly for both JPEG and MPEG as a function of the number of packets lost in transmission, and this degradation is much more dramatic for MPEG than the less highly compressed JPEG. Both JPEG and MPEG apply a discrete cosine transform to the video image to produce what is hereafter called a message [21, Wallace], [14, Le Gall]. Besides allowing a highly compressed representation of the image, this message has a nice property. Consider ordering the information in the message so that the lowest frequency coefficients come first followed by successively higher frequency coefficients. The nice property is that the quality of the image that can be reconstructed from a prefix of this ordered message improves gracefully as a function of the length of the prefix. Thus, the information at the beginning of the message is more important than that at the end. A PET system can be used to protect the different parts of the message from losses according to their importance.

A simpler way to protect MPEG using a PET system is by prioritizing over the different types of frames used in MPEG. An MPEG stream consists of a sequence of so-called I-,P-, and B-frames. Each I-frame can be displayed independently of the other frames. A P-frame needs information of the previous I-frame to be displayed correctly. Each B-frame refers to the previous and the following I- or P-frame. This defines a natural priority order for the different types of frames, I-frames are the most important frames, then come the P-frames. The B-frames are the least important frames. A PET system can be used to protect I-frames more against losses than P-frames, which in turn are protected more than B-frames. This approach has been taken in [13]. The results are promising.

3 Requirements of a PET system

We assume throughout this paper that there is a basic word size w which is long enough to implement all of the encoding schemes we describe. For all the schemes, $w \geq \log(e)$, where e is the total number of words in the encoding, is more than sufficient. In practice, a normal computer word of length 32 is more than enough to support all reasonable length encodings, i.e., encodings of up to over four billion words. In the sequel, all information is implicitly specified in units of words of length w unless otherwise stated.

3.1 Deterministic PET systems

In this system the encoding and decoding is done deterministically. A guarantee is given that once a certain fraction of the encoding is received the decoding of certain pieces of the message is always successful.

Definition 3.1 (PET system) *A PET system with message length m, packet size ℓ, n packets, and encoding length $e = n\ell$ consists of the following:*

(i) An encoding function *E that maps a message M of length m onto an encoding $E(M)$ of total length e consisting of n packets of ℓ words each, i.e. $e = n\ell$.*

(ii) A decoding function *D that maps sets of at most n packets onto m words.*

(iii) A priority function *ρ that maps $\{1, \ldots, m\}$ to the interval $(0, 1]$.*

The guarantee of the system is that, for all messages of length m and for all $i \in \{1, \ldots, m\}$, D is able to decode the i^{th} message word from any ρ_i fraction of the n encoding packets.

Convention: *Throughout this paper we assume that each packet has a unique identifier written in its header. The number of bits necessary to represent this identifier is not considered as part of the packet size.*

This convention is justified because packets usually contain such an identifier in the header information anyway, and in any case the identifier is typically very small compared to the rest of the packet. The identifier is used in the decoding process to identify which portions of the encoding have been received.

Throughout this paper we assume without loss of generality that the priority function is monotonically increasing, i.e., $\rho_1 \leq \rho_2 \leq \cdots \leq \rho_m$. Thus, ρ_i can also be thought of as the fraction of encoding packets needed to recover the first i words of the message.

In our implementations of PET systems ([2], [13]) the user specifies the message length m, the packet length ℓ, and the priority function ρ, and the system computes the number of encoding packets n and implements the encoding and decoding procedures that achieve the guarantees as specified in Definition 3.1.

An important measure of a priority function is the following.

Definition 3.2 (Girth of a priority function/PET system) *Let ρ be a function mapping $\{1, \ldots, m\}$ to the interval $(0, 1]$. The* girth *of ρ is*

$$\mathrm{girth}_\rho = \sum_{i \in \{1,\ldots,m\}} 1/\rho_i.$$

The girth of a PET system is the girth of its priority function.

In a PET system with priority function ρ, each ρ_i fraction of the encoding must determine the i-th message word M_i. Intuitively, this implies that each ρ_i fraction of the encoding must contain at least one word of information about M_i, and thus the entire coding must contain at least $1/\rho_i$ words of information about M_i. Therefore, intuitively the encoding contains $\mathrm{girth}_\rho = \sum_{i \in \{1,\ldots,m\}} 1/\rho_i$ words in total about the message. Hence, it is reasonable to expect that such a system is possible only if the total length of the encoding is at least girth_ρ. The following theorem shows that this intuition is correct.

Theorem 3.3 *For any priority function ρ, if there is a PET system with priority function ρ then the total encoding length is at least* girth_ρ.

A formal proof of this theorem is given in Section 5.

It will also be shown (Theorem 4.3) that, for a given priority function ρ, a PET system with a priority function ρ' that closely approximates ρ can be constructed with total encoding length $\mathrm{girth}_{\rho'}$.

3.2 Probabilistic PET systems

In the model described in this section the encoding and decoding is done via randomized algorithms. Unlike in the previous model the decoding guarantee is only with high probability. As mentioned at the end of Section 4.3, probabilistic PET systems based on probabilistic erasure-resilient codes (see the end of Section 4.1) admit faster encoding and decoding algorithms than deterministic PET systems.

Definition 3.4 (probabilistic PET system) *A probabilistic PET system with message length m, packet size ℓ, n packets, encoding length $e = n\ell$, failure probability $p > 0$ and using r random bits consists of the following:*

(i) A family of encoding functions E^R, $R \in \{0,1\}^r$, that map a message M of length m onto an encoding consisting of n packets of ℓ words each, i.e. $e = n\ell$ words.

(ii) A family of decoding functions D^R, $R \in \{0,1\}^r$, that map sets of at most n packets onto m words.

(iii) A priority function ρ that maps $\{1,\ldots,m\}$ to the interval $(0,1]$.

The guarantee of the system is that, for all messages M of length m, for all $i \in \{1,\ldots,m\}$, and for any ρ_i fraction of the n encoding packets, if the function E^R was used for the encoding then with probability at least $1-p$ the function D^R decodes the i^{th} word of the message from this subset. The probability is with respect to the uniform distribution on the random string $R \in \{0,1\}^r$.

In the probabilistic model it is assumed that a *common* random string R is used for the encoding and the decoding. Once the string R has been selected the encoding and decoding is deterministic. We stress that the failure probability is not over a particular distribution over the messages. For any fixed value of R an encoding/decoding pair E^R, D^R succeeds or fails on certain subsets of packets, independent of the message. This is a reasonable definition if in practice the set of packets that are lost is independent of their contents, but can depend on their identifiers in an arbitrary way.

The priority function ρ has a similar meaning as in the deterministic model, except that even if more than a fraction ρ_i of the encoding packets are received there may still be a chance (at most p) that the decoding function fails to decode the i-th message word.

In Section 5 we show that Theorem 3.3 can be generalized to probabilistic PET systems in the following way.

Theorem 3.5 *For any priority function ρ, if there is a probabilistic PET system with priority function ρ that achieves a failure probability p then the total encoding length is at least* $(1-p)\cdot \mathrm{girth}_\rho$.

4 A PET System

We describe a general method that takes any given message length m, packet size ℓ, and priority function ρ and produces a PET system with a new priority function ρ' that closely approximates ρ, such that the total length of the encoding packets is $\mathrm{girth}_{\rho'}$.

The method to produce a PET system works by first partitioning the message into blocks based on the priority function ρ, and then using the partition to implement a PET system based on erasure-resilient codes.

In the first subsection we describe erasure-resilient codes. In the second subsection, we assume we have the partitioned message and show how to implement a PET system based on erasure-resilient codes. Finally, we describe an algorithm that accepts the description of an arbitrary priority function ρ and produces a partitioned message.

4.1 Erasure-Resilient Codes

An erasure-resilient code is specified by a triple $\langle m, n, d\rangle$. There is both an encoding algorithm and a decoding algorithm. The encoding algorithm encodes an m-word message M into an n-word encoding $E(M)$ and has the property that the encodings of two different messages differ in at least d words.

Note that by the definition of d, any message M is uniquely distinguished from any other message by any $n-d+1$ words of its encoding $E(M)$. The decoding algorithms we consider are able to uniquely and efficiently recover M from any $n-d+1$ words of $E(M)$. It is impossible to always be able to recover a message of length m from less than m words of the encoding, and thus it is always the case that $m \leq n - d + 1$. Furthermore, the larger the value of d the better the recovery properties of the decoding. In the best case, when $d = n - m + 1$, the code is called in the literature a *maximum distance separable (MDS) code* (see for example [15]). In this case the message can be recovered from any portion of the encoding (in units of words) equal to the length of the message. In this paper, all codes are MDS unless otherwise specified.

The decoding algorithm needs the indices of the words of $E(M)$ it receives to help in the decoding process. When erasure-resilient resilient codes are used to implement a PET system, this requirement is satisfied because of the convention mentioned previously that each packet contains a unique index.

One implementation of erasure-resilient codes is the following. The message M is viewed as describing the m coefficients of a univariate polynomial of degree $m - 1$ over GF$[2^w]$. Call this polynomial G. The j^{th} word of the code consists of the value of the polynomial G evaluated at the field element $j \in$ GF$[2^w]$. Since G is of degree $m - 1$, any m words (together with the indices of the words) uniquely determine G. The message M, i.e., the coefficients of G, can be recovered from any m words by interpolation. Since we need to evaluate the polynomial at n different points this method requires $2^w \geq n$.

Using standard evaluation and interpolation algorithms, for this erasure-resilient code the encoding and decoding both require a quadratic number of field operations. Using the Discrete Fourier Transform, this can be reduced to $\mathcal{O}(n \log m)$ field operations for the encoding and $\mathcal{O}(m \log^2 m)$ field operations for the decoding. The practical value of these methods is doubtful.

A practically efficient erasure-resilient code has been described in [9, Blömer et al.]. It is a variant of Reed-Solomon-codes that is based on so-called Cauchy matrices (see the references in [9]). This implementation takes quadratic time for both the encoding and decoding. However, it is efficient enough to support existing real-time video applications implemented on current workstations (see [13]). This code is also *systematic*, i.e., the unencoded message is part of the encoding. This has the advantage that the decoding time depends only on how much of the unencoded message is missing, and in particular the decoding is trivial if none of the unencoded message part of the encoding is missing.

A different family of codes, called $(1+\epsilon)$-MDS codes, that have slightly weaker erasure-resilient properties than the MDS-code described above have been described and constructed in [4, Alon et al.] and [5, Alon, Luby]. For these codes, the requirement is that the message can be recovered from any $(1+\epsilon)m$ words of the encoding. Here ϵ is an adjustable parameter that is used to establish a tradeoff between the erasure-resilient properties of the code and the efficiency of the encoding and decoding procedures. The codes are based on expander graphs and, for constant ϵ, admit linear time encoding and decoding. At present their practical value is doubtful. However, probabilistic codes based on the ideas in [4],[5], on the codes using Cauchy matrices and on ideas from [18, Spielman] show promise of being more efficient than deterministic codes in practice. As mentioned at the end of Section 4.3, these codes can be used directly to implement probabilistic PET systems that have faster encoding and decoding algorithms.

The erasure-resilient codes based on Cauchy matrices require that the word size w satisfy

$$w - 1 \geq \max\{\log(m/w), \log(n - m/w)\}. \tag{1}$$

Theorem 6.1 found in Section 6 proves an almost matching lower bound for the word size of any erasure-resilient code.

4.2 Block Systems

The input parameters for a PET system are a message length m, a packet length ℓ, and a priority function ρ. The first step in constructing a PET system is to compute the total number of encoding packets n and to partition the message into ℓ blocks. This first step is described in the next subsection. In this subsection, we show how to implement a PET system given this information.

An *ℓ-partition of m* consists of a sequence of positive integers $\langle m_1, \ldots, m_\ell \rangle$ such that $\sum_{j\in\{1\ldots,\ell\}} m_j = m$. Let M be a message of length m, and let $B_1, \ldots, B_\ell$ be the ℓ blocks of M with respective lengths $m_1, \ldots, m_\ell$. We now describe how to implement a PET system based on an ℓ-partition of m and on the total number n of encoding packets. The PET system puts information about block B_j into the j^{th} word of each packet.

Lemma 4.1 *Given n and an ℓ-partition $\langle m_1, \ldots, m_\ell \rangle$ of m, there is a PET system consisting of n encoding packets containing ℓ words each such that the priority value for all words of the message in block j is m_j/n.*

Proof of Lemma 4.1: Let $B_1, \ldots, B_\ell$ be the blocks of M, and thus the length of B_j is m_j. The idea is to use a separate erasure-resilient code for each of the ℓ blocks of the message. The j^{th} erasure-resilient code is used to encode B_j into a code E_j consisting of n words. The entire encoding consists of n packets of size ℓ each, where the k^{th} packet consists of the concatenation, for $j \in \{1, \ldots, \ell\}$, of the k^{th} word from the code E_j. The decoding works in the obvious way.

Since we use an erasure-resilient code for each block, all words in the same block have the same priority value. Any m_j words of the code E_j suffice to recover block B_j. Since there is one such word in each packet, it follows that a fraction m_j/n of the n packets are sufficient to recover B_j. Thus, the priority value of all words in block B_j is as claimed. □

In the system described above, each packet needs to contain an identifier. Although this is part of the packet, we did not include it in the packet size because of the convention stated in Section 3.

We give two examples of block systems.

Example 1: This is an example where the fraction of the packets needed to recover a message word is linear in its index. For a given message length m, let the packet length be $\ell = \log(m)$. For all $j \in \{1, \ldots, \ell\}$, let B_j be the next $m_j = 2^j$ consecutive words of the message, and let the number of packets be $n = 2m$. Note that all words in B_j can be recovered from a fraction $2^{j-1}/m$ of the packets. Also, the total encoding length is $2m\log(m)$.

Example 2: Suppose the message length is 800, the packet length is 10, and the 10-partition of the message is $\langle 60, 60, 75, 75, 75, 75, 95, 95, 95, 95 \rangle$, and the number of packets is 100. Note that the first two blocks can be recovered from any 60% of the packets, the next four blocks from any 75% of the packets, and the remaining four blocks from any 95% of the packets. The total encoding length in this example is 1000 words, and thus the total message length is an 80% fraction of the total encoding length.

4.3 Partitioning a Message

We assume that the priority function ρ for a message of length m specifies d different priority levels, where d is smaller than the packet length ℓ. This is not a big assumption in practice, as IP packets for transferring data at reasonably high rates are typically between 500 and 1500 bytes long (which is between 125 and 375 words assuming 4 bytes per word), and usually 10 priority levels is more than sufficient. Let $\langle \alpha_1, \ldots, \alpha_d \rangle$ be a d-partition of m, and let $\rho = \langle \rho_1, \ldots, \rho_d \rangle$ be the corresponding priority values of the blocks, i.e., all words in block i of the partition have priority value ρ_i.

Our goal is to produce an ℓ-partition of m and the number of encoding packets n that can be directly used to implement a PET system based on Lemma 4.1. The basic idea is to refine the original d-partition in a simple way, although we must take care of some technical details due to round-off errors.

Refinement Procedure:

(1) Compute $g = \text{girth}_\rho = \sum_{i \in \{1,\ldots,d\}} \alpha_i/\rho_i$.

(2) Compute $n = \left\lceil \frac{g}{\ell - d} \right\rceil$.

(3) For all $i \in \{1, \ldots, d\}$, compute $\beta_i = \lceil \rho_i n \rceil$.

(4) For all $i \in \{1, \ldots, d\}$, subpartition α_i into at most $\lceil \alpha_i/\beta_i \rceil$ pieces of length at most β_i each.

Lemma 4.2 *On input m, ℓ, a d-partition $\langle \alpha_1, \ldots, \alpha_d \rangle$ of m, and corresponding priority values $\langle \rho_1, \ldots, \rho_d \rangle$, the* **Refinement Procedure** *produces a refined partition and n with the following properties:*

(i) *The refined partition has at most ℓ parts.*

(ii) *The value of n satisfies $n \leq \frac{g}{\ell - d} + 1$.*

(iii) *Each part in the refinement of the i^{th} part of the d-partition has length at most $\rho_i n + 1$.*

Proof of Lemma 4.2: To prove (i), note that the number of parts in the refined partition is at most $d + \sum_{i \in \{1,\ldots,d\}} \alpha_i/\beta_i$. Because $\frac{\alpha_i}{\beta_i} \leq \frac{\alpha_i}{\rho_i n} \leq \frac{\alpha_i}{\rho_i} \cdot \frac{\ell - d}{g}$, and by definition of g, it follows that $\sum_{i\in\{1,\ldots,d\}} \alpha_i/\beta_i \leq \ell - d$, and thus the total number of parts is at most $d + (\ell - d) = \ell$. The proofs of parts (ii) and (iii) follow directly from the definitions. $\square$

Theorem 4.3 *On input message length m, packet length ℓ, a d-partition $\langle \alpha_1, \ldots, \alpha_d \rangle$ of m, and corresponding priority values $\langle \rho_1, \ldots, \rho_d \rangle$, there is an efficient procedure that produces a PET system with priority function ρ' and n encoding packets with the following properties:*

(i) *The total encoding length is $n\ell \leq \frac{\mathrm{girth}_\rho}{1-d/\ell} + \ell$.*

(ii) *All words of the message in the i^{th} block of the d-partition have priority value $\rho'_i \leq \rho_i + \ell/m$.*

Proof of Theorem 4.3: The proof follows by a direct combination of Lemma 4.2 and Lemma 4.1. The only detail missing in the proof of part (ii) is that since $n\ell \geq g \geq m$, $n \geq m/\ell$, and thus $1/n \leq \ell/m$. $\square$

Example: Suppose the packet length is 250, and a message of total length $100K$ is partitioned into five priority levels described by the five partition

$$\langle 10K, 10K, 20K, 30K, 30K \rangle$$

with associated priority values

$$\langle .50, .60, .65, .80, .95 \rangle.$$

The girth of the priorities g computed in step (1) of the **Refinement Procedure** is $136.5K$, and thus the total number of packets n computed in step (2) is 558. In step (3), the computed lengths of the pieces are

$$\langle 279, 335, 363, 447, 531 \rangle,$$

and the number of pieces of each is at most

$$\langle 36, 30, 56, 68, 57 \rangle,$$

respectively, for a total of 247 pieces (recall that 250 is the target value). The total length of the encoding is $139.5K$, which is only 2% more than the girth of the original priorities. The priority values for the resulting PET system are

$$\langle .500, .600, .651, .801, .952 \rangle,$$

i.e., extremely close to the specified priorities. Note that if there were only one priority level with the same amount of overall redundancy then it would be possible to recover the message from any $.72 = 100/139.5$ fraction of the encoding, i.e., a fraction that is somewhere in the middle of the five priority values.

The refinement procedure implemented in [2] is based on the refinement procedure described above, except that it doesn't necessarily produce a refinement of the d-partition. It alleviates the effect of the round-offs by moving through the d-partition from the beginning to end, refining the partition as described above, except that the last block of the ℓ-partition associated with a particular part of the d-partition may be padded out with some words of the subsequent block of the d-partition. It also adjusts the number of packets downwards until all the words of the packet are used (in the example above, three words of the packet were left unused, and thus the total encoding length is slightly more than the girth of the new priorities).

A probabilistic PET system with theoretically more efficient encoding and decoding times can be constructed similar to the deterministic scheme described above, where the theoretically more efficient $(1+\epsilon)$-MDS probabilistic erasure-resilient codes of [4], [5] are used in place of deterministic MDS erasure-resilient codes.

5 Lower Bound on the Encoding Length

This section proves Theorem 3.3, i.e., for any priority function ρ, any PET system with priority function ρ has total encoding length at least girth_ρ. Using similar methods, we also prove Theorem 3.5, i.e., for any priority function ρ, any probabilistic PET system with priority function ρ and failure probability p has total encoding length at least $\text{girth}_\rho \cdot (1-p)$.

The proofs we give here are alternatives of our original proofs of the same results. The alternate proofs were found by Noga Alon and independently by Stephan Boucheron. They follow the same basic outline as the original proofs, but they are more elegant than the originals because they use entropy measures instead of geometric measures of information.

Theorem 5.1 *Let $\tau_1, \ldots, \tau_m$ be m finite alphabets and let $\sigma_1, \ldots, \sigma_n$ be n finite alphabets. Suppose we have a (deterministic) scheme that encodes each possible vector $M = \langle M_1, \ldots, M_m \rangle$, where $M_i \in \tau_i$, by a vector $E = \langle E_1, \ldots, E_n \rangle$, where $E_j \in \sigma_j$. Suppose that $0 < \rho_1 \le \rho_2 \le \ldots \le \rho_m \le 1$, and the value of M_i can be correctly recovered from the values of* any *set of at least $\rho_i n$ of the coordinates of E. Then,*

$$\sum_{i=1}^{m} \frac{\log |\tau_i|}{\rho_i} \le \sum_{i=1}^{n} \log |\sigma_i|.$$

Here, and in what follows, all the logarithms are in base 2. In our applications to PET systems, M is the message and E is its encoding. In this application, $\tau_1 = \cdots = \tau_m = \{0,1\}^w$ are the possible encodings of message words, $\sigma_1 = \cdots = \sigma_n = \{0,1\}^{\ell w}$ are the possible encodings of packets, and $\rho = \langle \rho_1, \ldots, \rho_m \rangle$ are the priority values of the message words.

This theorem immediately implies Theorem 3.3 in even the more general case where each message word and each packet is allowed to have a different number of symbols. We prove Theorem 5.1 after first introducing some ideas used in the proof.

5.1 Preliminaries for the Lower Bound

For any random variable Y with density function Pr,

$$H(Y) = \mathrm{Exp}[-\log(\Pr[Y])]$$

denotes the binary entropy of Y. Let $X_1, \ldots, X_n$ be random variables taking values in $\sigma_1, \ldots, \sigma_n$, respectively and let $X = \langle X_1, \ldots, X_n \rangle$. For a subset I of $\{1, \ldots, n\}$, let X_I denote the random variable $\langle X_i \rangle_{i \in I}$. With these notations, the following proposition is proved in [10] for the case $\sigma_i = \{0,1\}$ for all i. The general case, mentioned in [3], can be proved analogously.

Proposition 5.2 *Let $X = \langle X_1, \ldots, X_n \rangle$ be as above. If $\mathcal{I}$ is a family of subsets of $\{1, \ldots, n\}$ and each $i \in \{1, \ldots, n\}$ belongs to at least r members of $\mathcal{I}$ then*

$$r \cdot H(X) \le \sum_{I \in \mathcal{I}} H(X_I).$$

For $1 \le q \le n$, define

$$H_q(X) = \frac{1}{\binom{n-1}{q-1}} \sum_{\substack{Q \subseteq \{1,\ldots,n\} \\ |Q| = q}} H(X_Q).$$

The following lemma is due to Han [11]. For the sake of completeness we present a short proof.

Lemma 5.3 *For any random variable $X = \langle X_1, \ldots, X_n \rangle$,*

$$H_1(X) \ge H_2(X) \ge \ldots \ge H_n(X) = H(X).$$

Proof of Lemma 5.3: By Proposition 5.2, for any q, $1 < q \le n$,

$$(n-q+1)\binom{n-1}{q-2} H_{q-1}(X) = \sum_{\substack{Q \subseteq \{1,\ldots,n\} \\ |Q| = q}} \sum_{\substack{Q' \subset Q \\ |Q'| = q-1}} H(X_{Q'})$$

$$\ge \sum_{\substack{Q \subseteq \{1,\ldots,n\} \\ |Q| = q}} (q-1) H(X_Q) = (q-1)\binom{n-1}{q-1} H_q(X).$$

$\square$

5.2 Deterministic Lower Bound Proof

Proof of Theorem 5.1: Let $M = \langle M_1, \ldots, M_m \rangle$ attain each value in $\tau_1 \times \ldots \times \tau_m$ with equal probability. Put $\rho_0 = 1/n$ and, for each $i \in \{1, \ldots, n\}$, define $h_i = \log|\tau_i| = H(M_i)$. Let $J = \{1, \ldots, j\}$, and thus $M_J = \langle M_1, \ldots, M_j \rangle$. We prove, by induction on $j \le m$, that

$$H_1(E) \ge H_{\rho_j n}(E|M_J) + \sum_{i=1}^{j} \frac{h_i}{\rho_i}. \tag{2}$$

For $j = 0$ there is nothing to prove. Assuming the above holds for j, $(0 \le j < m)$, we prove it for $j+1$. Define $q = \rho_{j+1} n$ and fix $Q \subseteq \{1, \ldots, n\}$, $|Q| = q$. Then,

$$H(E_Q|M_J, M_{j+1}) \quad = \quad H(E_Q, M_{j+1}|M_J) - H(M_{j+1}|M_J) \tag{3}$$

But, because M_{j+1} is independent of M_J,

$$H(M_{j+1}|M_J) = H(M_{j+1}) = h_{j+1}. \tag{4}$$

Furthermore, since the value of E_Q determines that of M_{j+1},

$$H(M_{j+1}|E_Q, M_J) = 0, \tag{5}$$

and thus,

$$\begin{aligned} H(E_Q, M_{j+1}|M_J) &= H(E_Q|M_J) + H(M_{j+1}|E_Q, M_J) \\ &= H(E_Q|M_J). \end{aligned} \tag{6}$$

From Equation (3), Equation (4), and Equation (6), it follows that

$$H(E_Q|M_J, M_{j+1}) \quad = \quad H(E_Q|M_J) - h_{j+1}. \tag{7}$$

By summing over all possible subsets Q and dividing by $\binom{n-1}{q-1}$ we conclude that

$$H_q(E|M_J, M_{j+1}) = H_q(E|M_J) - \frac{n}{q}h_{j+1} = H_q(E|M_J) - \frac{h_{j+1}}{\rho_{j+1}}.$$

By Lemma 5.3, since $q = \rho_{j+1}n \geq \rho_j n$,

$$H_q(E|M_J) \leq H_{\rho_j n}(E|M_J),$$

and this, together with the induction hypothesis, implies the assertion of (2) for $j+1$, completing the proof of the induction step. Since

$$H_1(E) = \sum_{i=1}^{n} H(E_i) \leq \sum_{i=1}^{n} \log|\sigma_i|,$$

the assertion of the theorem follows by taking $j = m$ in (2). ◻

5.3 Probabilistic Lower Bound Proof

Theorem 5.4 *Let $\tau_1, \ldots, \tau_k$ be m finite alphabets and let $\sigma_1, \ldots, \sigma_n$ be n finite alphabets. Let R be a random string that takes on all possible values in a set S with equal probability. Suppose we have a scheme that uses R to* *encode each possible vector $M = \langle M_1, \ldots, M_m \rangle$, where $M_i \in \tau_i$, by a vector $E = \langle E_1, \ldots, E_n \rangle$, where $E_j \in \sigma_j$. Suppose that $0 < \rho_1 \leq \rho_2 \leq \ldots \leq \rho_m \leq 1$, and, for* any *set of at least $\rho_i n$ of the coordinates of E, the value of M_i can be correctly recovered from the values of these coordinates and from the value of R with probability at least $1 - p$ with respect to R. Then,*

$$(1-p) \cdot \sum_{i=1}^{m} \frac{\log |\tau_i|}{\rho_i} \leq \sum_{i=1}^{n} \log |\sigma_i|.$$

Proof of Theorem 5.4: The proof is analogous to the proof of Theorem 5.1, except that all quantities are conditioned on R, e.g., $H(E)$ becomes $H(E|R)$. The only significant difference is that because a particular set of packets indexed by Q determines M_{j+1} with probability at least $1 - p$ (as opposed to always determining M_{j+1} before), Equation (5) becomes

$$H(M_{j+1}|E_Q, M_J, R) \leq p,$$

and Equation (7) becomes

$$H(E_Q|M_J, M_{j+1}, R) \quad \leq \quad H(E_Q|M_J, R) - (1-p)h_{j+1}.$$

The remainder of the proof is identical, except that the loss in the $1 - p$ factor remains throughout the rest. $\square$

This theorem immediately implies Theorem 3.5 in even the more general case where each message word and each packet is allowed to have a different number of symbols.

6 Lower Bound on the Size of a Packet

In this section a lower bound for the word size of erasure-resilient codes. These bounds imply weak bounds on the packet size of a PET system.

Theorem 6.1 *Let $\mathcal{C}$ be a $\langle n, m, d \rangle$ erasure-resilient code with word length w. Then,*

$$2^w + 1 \geq \frac{n - m + 2}{n - m - d + 2}.$$

Equivalently, if any k words of the encoding determine the message, then

$$2^w + 1 \geq \frac{n - m + 2}{k - m + 1}.$$

Proof of Theorem 6.1: Since $k = n - d + 1$, we only need to prove the second assertion of the theorem. View each word as an element of the set $Q = \{0, \ldots, 2^w - 1\}$ with 2^w elements, and view each message as an element of the set $T = Q^m$. For a message $M \in T$, denote by $E(M) = \langle E_1(M), \ldots, E_n(M)\rangle$ its encoding. Here each E_i is a function from T to Q. For each encoding $E(M)$ consider its suffix $\langle E_{n-m+3}(M), \ldots, E_n(M)\rangle$ consisting of the last $m-2$ words. Since there are at most $2^{w(m-2)}$ such suffixes, and since there are 2^{wm} possible messages, there must be a set $S \subset T$ of 2^{2w} messages all of which have the same encoding suffix of length $m-2$.

For a fixed $i \in \{1, \ldots, n-m+2\}$ and for any $k \in Q$, let n_k be the number of messages $M \in S$ such that $E_i(M) = k$. The number of pairs of messages $\langle M, M'\rangle \in S^2$ such that $E_i(M) = E_i(M')$ is

$$\sum_{k=0}^{2^w-1} \binom{n_k}{2}.$$

This sum is minimized if $n_k = 2^w$ for all k in which case its value is $2^w\binom{2^w}{2}$. Hence for each $i \in \{1, \ldots, n-m+2\}$ there are at least $2^w\binom{2^w}{2}$ pairs of messages $(M, M') \in S^2$ such that $E_i(M) = E_i(M')$.

Any two messages $M, M' \in S$ must be distinguishable from the last $m-2$ words and any other $k-m+2$ words of their encodings $E(M), E(M')$.

If $\frac{n-m+2}{k-m+1} > 2^w + 1$ then

$$\binom{2^{2w}}{2} < \frac{n-m+2}{k-m+1} 2^w \binom{2^w}{2}.$$

By the previous calculations this implies that there are two different messages $M, M' \in S$ whose encodings agree in at least $k-m+2$ of the first $n-m+2$ words. Since the encodings already agree in the last $m-2$ words, the encodings of these two messages agree in at least k words. This contradicts the assumption of the theorem and shows $2^w + 1 \geq \frac{n-m+2}{k-m+1}$. ◻

There are two interesting special cases for this theorem. If the code is an MDS erasure-resilient code then $k = m$. Hence the theorem yields $w \geq \log(n - m + 1)$.

Secondly, assume that the system has encoding length cm for some constant c and that $k = (1 + \epsilon)m$. Efficiently encodable and decodable codes with these parameters and with one priority level have been constructed in [4],[5]. For these codes the theorem gives a lower bound on the word size of approximately $\log((c-1)/\epsilon)$. The bound is asymptotically tight. The algebraic-geometric codes achieve these bounds (see for example [19]).

Notice that if k is $cm, c > 1$ then the theorem can only yield constant lower bounds. In particular, for a PET system with two priority levels of equal length, where the first level has priority 50% and the second one has priority strictly larger than 50%, the theorem only provides a constant lower bound on the packet size.

7 Acknowledgment

We thank Richard Karp for help in the original proofs of Section 5, and both Noga Alon and Stephan Boucheron for pointing out the alternative proofs to Theorem 3.3 and Theorem 3.5.

References

[1] A. Albanese, J. Blömer, J. Edmonds, M. Luby, M. Sudan, *Priority Encoding Transmission*, Proc. 35^{th} Symposium on Foundations of Computer Science (FOCS), 1994, pp. 604-613.

[2] A. Albanese, M. Kalfane, B. Lamparter, M. Luby, *Application Programmer Interface for PET*, internal ICSI document.

[3] N. Alon, *Probabilistic methods in extremal finite set theory*, in: Extremal Problems for Finite Sets, (G. O. H. Katona et al. Eds.), Bolyai Society Mathematical Studies, Visegrád, Hungary, 1991, 25-43.

[4] N. Alon, J. Edmonds, M. Luby, *Linear time erasure codes with nearly optimal recovery*, in Proc. 36^{th} Symposium on Foundations of Computer Science, 1995, pp. 512-519.

[5] N. Alon, M. Luby, *Linear time erasure codes with nearly optimal recovery*, submitted to this special issue.

[6] N. Alon, J. H. Spencer, *The probabilistic method*, John Wiley & Sons, Inc., New York, 1992.

[7] E. R. Berlekamp, *Algebraic Coding Theory*, McGrawhill, New York, 1968.

[8] E. W. Biersack, *Performance evaluation of forward error correction in an ATM enviroment*, Journal of Selected Areas in Communication, 11(2)(1993), pp. 631-640.

[9] J. Blömer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, D. Zuckerman, *An XOR-based erasure-resilient coding scheme*, Technical Report TR-95-048, International Computer Science Institute, Berkeley, 1995.

[10] F. R. K. Chung, P. Frankl, R. L. Graham and J. B. Shearer, *Some intersection theorems for ordered sets and graphs*, J. Combinatorial Theory, Ser. A 43 (1986), 23-37.

[11] T. S. Han, *Nonnegative entropy measures of multivariate symmetric correlations*, Infor. Contr. 36 (1978), 133-156.

[12] G. H. Hardy, J. E. Littlewood, G. Pólya, *Inequalities,* Cambridge University Press, 1934.

[13] B. Lamparter, A. Albanese, M. Kalfane, M. Luby, *PET - Priority Encoding Transmission: A New, Robust and Efficient Video Broadcast Technology,* video tape and paper appear in ACM Multimedia, 1995.

[14] D. Le Gall, *MPEG: A Video Compression Standard for Multimedia Applications*, CACM, Vol. 34, No. 4, April 1991, pp. 47-58.

[15] F. J. MacWilliams, N. J. A. Sloane, *The theory of error-correcting codes*, North-Holland, 1977.

[16] A. J. McAuley, *Reliable broadband communication using a burst erasure correcting code*, in Proceedings SIGCOMM'90, Philadelphia, 1990.

[16] M. O. Rabin, *Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance*, J. ACM, Vol. 36, No. 2, April 1989, pp. 335-348.

[17] N. Shacham, *Multicast Routing of Hierarchical Data,* Proceedings of ICC'92, Chicago, 1992.

[18] D. Spielman, *Linear time encodable and decodable error-correcting codes*, in Proc. 27^{th} Symposium on Theory of Computing (STOC), 1995, pp. 388-397.

[19] M. A. Tsfasman, S. G. Vladut, *Algebraic-geometric codes*, Kluwer Academic Publishers, 1991.

[20] J. H. van Lint, *Introduction to coding theory*, Springer Verlag, 1982.

[21] G. K. Wallace, *The JPEG Still Picture Compression Standard*, CACM, Vol. 34, No. 4, April 1991, pp. 30-44.