A Deterministic Approach to Rate-Compatible Fountain Communication

Thomas Courtade and Richard D. Wesel
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, CA 90095, USA
Email: {tacourta, wesel}@ee.ucla.edu

***Abstract*— This paper considers a scenario in which a transmitter wishes to communicate n symbols (Galois field elements) to an arbitrary number of receivers. Each receiver knows some of the original n symbols, and we desire a transmission that allows each receiver to learn the entire n-symbol message from the fewest possible transmitted symbols. Specifically, we assume that receiver i knows k_i of the original n symbols (and their respective indices in the information vector). The value k_i and the values of the indices are unknown to the transmitter. With the proposed rate-compatible transmission scheme, each receiver i can recover the original n symbols after receiving the first $n - k_i$ transmitted symbols, the smallest number of symbols for which this is theoretically possible. The proposed scheme is based on the properties of maximum distance separable codes. A low complexity decoder implementation essentially performs Berlekamp-Massey erasure decoding of an affine shift of a Reed-Solomon code.**

I. Introduction

In this paper, we consider a problem referred to as "rate-compatible fountain communication" in [1]. Our model consists of a single transmitter in possession of a vector of n information symbols, denoted X^n, and N receivers (indexed by $i \in \{1, \ldots, N\}$). Each receiver possesses some random subset of the information symbols. We define k_i to be the number of information symbols that are known to receiver i and assume that receiver i knows the indices of those k_i symbols in the original message vector X^n. Neither the k_i's nor the locations of the known symbols at each receiver are known to the transmitter.

For example, if $n = 3$ and $X^n = (x_1, x_2, x_3)$, then receiver i might possess a vector $Y_i^n = (x_1, ?, x_3)$, where "?" denotes an erased symbol. The goal is for the transmitter to generate a sequence of symbols S^n, so that the receiver can learn the entire sequence of symbols after receiving the fewest possible symbols of S^n, with the ultimate limit being $n-k_i$ symbols. In the example of this paragraph, receiver i would like to learn the erased value of x_2 after receiving one symbol from the transmitter.

This sort of information sharing is a useful endgame to a situation where each peer in a group of nodes knows a subset of the desired n symbols. As soon as one of the nodes learns all n symbols, that node can transmit the rate-compatible broadcast sequence described in this paper to efficiently bring all other peers to complete knowledge of the n symbols.

Another application (similar to the example in [1]) for this model is the following: Suppose a software provider multicasts updates to its subscribers, and that a major software upgrade to be released consists of n smaller updates of equal size. Assume subscriber i has already installed k_i of the updates (for example, through an automated update service), and wants to download as few of the remaining updates as possible (due to resource or cost constraints) to obtain the major upgrade. Also due to resource or cost constraints, the software provider would like each subscriber to receive the minimum possible number of updates while still allowing every subscriber to obtain the major upgrade.

This problem can be analyzed using information theory, and we can achieve optimal performance in a practical manner through an application of maximum distance separable (MDS) codes. This paper is organized as follows. Section II gives a brief information-theoretic analysis of the problem. Practical coding schemes that achieve optimal performance are developed for progressively more general scenarios in Section III. Section IV delivers the conclusions.

II. An Information Theory Perspective

Let $\mathcal{X}$ be a finite alphabet, and let X^n be a vector drawn uniformly at random from $\mathcal{X}^n$. Let $Y_i^n = (y_{i,1}, \ldots, y_{i,n})$ be a randomly "erased" version of X^n as described in the example in Section I. Assume that Y_i^n consists of k_i known symbols (k_i could possibly be increasing with n) and that k_i satisfies $\lim_{n\to\infty}(n - k_i) = \infty$ for each i.

The astute reader will notice that each channel looks like a Slepian-Wolf scenario with (X^n, Y_i^n) being the pair of observations to be recovered, and Y_i^n available noiselessly at the receiver (i.e. a corner point in the achievable region). The key difference between this problem and a traditional Slepian-Wolf problem is that the transmitter must carefully design the Slepian-Wolf compression of X^n so that each receiver can successfully decode after receiving its respective minimum number of symbols. This paper presents a rate-compatible version of Slepian-Wolf coding for this corner point and shows its practical implementation.

Theorem 1: Rate-Compatible Slepian-Wolf. For any $\epsilon > 0$, there exists a coding scheme so that the probability of any receiver being unable to recover X^n from Y_i^n and $(1+\epsilon)(n-k_i)$ transmitted symbols approaches zero as $n \to \infty$.

Proof: Assign each of the information vectors a random label $S^n \in \mathcal{X}^n$ (each label drawn uniformly at random). Let $S^n(X^n)$ denote the label assigned to a particular X^n. This assignment is our random codebook and is made available to the transmitter and receivers. The transmitter, in possession of X^n broadcasts $S^n(X^n)$ to the receivers.

For receiver i, there are $|\mathcal{X}|^{n-k_i}$ different possible X^n's that are compatible with its own observation Y_i^n. Call this set $\mathcal{A}(Y_i^n)$. Upon receiving $(1+\epsilon)(n-k_i)$ of the symbols of $S^n(X^n)$, denoted V_i, receiver i checks for a $\tilde{X}^n \in \mathcal{A}(Y_i^n)$ such that $S^n(\tilde{X}^n)$ begins with V_i. By construction, at least one such $\tilde{X}^n$ exists (i.e. the original information vector X^n). If more than one such exists, an error is declared. Denote this error event E_i. By a union bound argument, it can be shown that:

$$\begin{aligned}\Pr(E_i) &\leq \sum_{\substack{\tilde{X}^n \in \mathcal{A}(Y_i^n) \\ \tilde{X}^n \neq X^n}} \Pr(S^n(\tilde{X}^n) \text{ starts with } V_i) \\ &= \sum_{\substack{\tilde{X}^n \in \mathcal{A}(Y_i^n) \\ \tilde{X}^n \neq X^n}} |\mathcal{X}|^{-(1+\epsilon)(n-k_i)} \\ &\leq |\mathcal{X}|^{-\epsilon(n-k_i)} \to 0 \text{ as } n \to \infty \qquad (1)\end{aligned}$$

Where (1) follows since $\lim_{n\to\infty}(n-k_i) = \infty$ by assumption. Since the number of receivers is finite, another union bound yields: $\Pr(\bigcup_{i=1}^{N} E_i) \to 0$ as $n \to \infty$. ∎

Therefore, almost any random mapping $S^n : \mathcal{X}^n \to \mathcal{X}^n$ provides a sufficiently good coding strategy so that the cutset bound is achieved simultaneously to each receiver.

When framed in the context of rate-compatible Slepian-Wolf coding, the above proof resolves the theoretical issues. Although the argument above gives the desired result, it has two main drawbacks (as is common with most information theory proofs), one is decoding complexity and the other is the fact that we require $n \to \infty$ for good performance. In many real-world applications, n may be quite small. The remainder of this paper demonstrates practical schemes that provide perfect rate-compatible fountain communication even at small blocklengths.

III. Practical Coding Schemes

We limit our attention to linear coding schemes because they are sufficient to our purpose while also providing ease of analysis and implementation. Their practical sufficiency is established by the algorithms presented below. However, linear coding schemes are sufficient for the proof of Theorem 1 by randomly generating a matrix $M \in \mathcal{X}^{n\times n}$ and letting $S^n(X^n) = MX^n$ (assuming X^n is a column vector and performing the matrix-vector multiplication modulo $|\mathcal{X}|$).

Remark 1: At this point in the paper, we shift gears from an information-theoretic analysis to a coding-theoretic analysis and our notation will reflect this. In particular, capital letters (such as A, G, H, etc.) will represent matrices and boldface letters (such as $\mathbf{x}$, $\mathbf{s}$, etc.) will be used to represent vectors. The dimensions of each should be clear from context. The j^{th} element of a vector $\mathbf{x}$ is denoted $\mathbf{x}(j)$. All vectors are column vectors. Linear codes are described by an ordered pair (n,k), or a triple (n,k,d), where n is the blocklength, k is the code dimension, and d is the minimum Hamming distance between two codewords.

A. General Solution When Transmitter Knows k

For pedagogical reasons, consider first the scenario of a single receiver with k known symbols. For this initial scenario, the number of known symbols, k, is also known the the transmitter. However, the transmitter does not know *which* $n-k$ symbols have been erased.

The solution to the problem at hand hinges on the the answer to the following question. Suppose we are constructing an $n \times n$ matrix A whose first k rows are k different natural basis vectors (vectors that are all zero except for a single one). Is it possible to choose the last $n-k$ rows of A so that A always has full rank, regardless of which k natural basis vectors form the first k rows of A? The answer is yes, as shown in the following lemma.

Lemma 1: Let A be an $n\times n$ matrix of Galois field elements for which the first k rows are k different natural basis vectors. If the Galois field is large enough to admit a linear maximum distance separable (MDS) code of length n and dimension $n-k$, then choosing the last $n-k$ rows of A to be a generator matrix for this MDS code guarantees that A always has full rank, regardless of which k natural basis vectors form the first k rows of A.

Proof: We show that A must be full rank by arguing that no non-zero linear combination of the rows can be equal to 0. If the last $n-k$ rows of A form a basis for an MDS code $\mathcal{C}$, then any nonzero linear combination of the last $n-k$ rows has Hamming weight at least $d_{\min}(\mathcal{C}) = n-(n-k)+1 = k+1$. However, since the first k rows are unique natural basis vectors, any linear combination of the first k rows has Hamming weight at most k. Thus, no nonzero linear combinations of the rows of A can be 0. ∎

In light of Lemma 1, we can solve the problem where a transmitter wants to communicate $\mathbf{x}$, to a receiver with a vector $\mathbf{y}$ consisting of k of the n symbols (the other $n-k$ symbols are erased). In this case, we assume that the value of k is known to the transmitter. The coding strategy is defined by G, a generator matrix for an $(n, n-k)$ MDS code known both to the transmitter and receiver. The transmitter sends the $(n-k)$-vector $\mathbf{s} = G\mathbf{x}$, and the receiver is able to recover $\mathbf{x}$ via Gaussian elimination.

B. Reed-Solomon Solution When Transmitter Knows k

The above method is optimal, achieving zero probability of error with zero overhead, for this simplified scenario in which the transmitter knows k. However, it's complexity can be reduced. Without loss of generality, the $(n, n-k)$ MDS code may be assumed to be a (shortened if necessary) Reed-Solomon code. For Reed-Solomon codes, we can replace Gaussian elimination (which has computational complexity

of $O\left((n-k)^3 + k(n-k)\right)$: $O\left(k(n-k)\right)$ operations are required to remove the k known symbols from the system of equations, and $O\left((n-k)^3\right)$ operations are required to solve the remaining system of $n-k$ equations) with a simpler decoder. The resulting approach not only reduces complexity, but lays the foundation for a solution to the general case.

Recall the well known fact that the dual code of an $(n, n-k, k+1)$ Reed-Solomon Code, $\mathcal{C}$, is an $(n, k, n-k+1)$ Reed-Solomon Code. Therefore, we observe that G=H^T, where H is the parity check matrix of the dual $(n, k, n-k+1)$ Reed-Solomon Code, $\mathcal{C}^\perp$. We address the complexity issue by appealing to the efficiency of the erasure-correcting version of the Berlekamp-Massey algorithm for $\mathcal{C}^\perp$.

Theorem 2: In the single-receiver scenario where the receiver has k symbols (k known to the transmitter), the receiver can recover $\mathbf{x}$ from $n-k$ transmitted symbols with decoding complexity $O\left(k^2 + (n-k)\right)$.

Proof: Suppose that the transmitter knows k, which allows the transmitter and receiver to select an appropriately dimensioned H as described above. Assume without loss of generality that H^T is of the form $H^T = [I \mid P]$. (I is the identity matrix). The transmitter sends the $(n-k)$-vector $\mathbf{s} = H^T\mathbf{x}$, which can be interpreted as the syndrome of $\mathbf{x}$ with respect to $\mathcal{C}^\perp$. In fact, by the structure of H, we observe that $\mathbf{x}^T = \left[\mathbf{s}^T\ \mathbf{0}^T\right] + \mathbf{c}_x^T$, where $\mathbf{c}_\mathbf{x}$ is a codeword in $\mathcal{C}^\perp$.

Upon receiving $\mathbf{s}$, the receiver adds its k known symbols to the corresponding symbols of $\left[\mathbf{s}^T\ \mathbf{0}^T\right]$ producing $\hat{\mathbf{c}}_\mathbf{x}$. We define $\hat{\mathbf{c}}_\mathbf{x}$ to be equal to $\mathbf{c}_\mathbf{x}$ with the $n-k$ coordinates erased corresponding to the symbols which are unknown to the receiver. Since $d_{\min}(\mathcal{C}^\perp) = n-k+1$, the receiver can uniquely decode $\hat{\mathbf{c}}_\mathbf{x}$ to $\mathbf{c}_\mathbf{x}$ using the erasure-correcting version of the Berlekamp-Massey algorithm. The decoding process is completed by recovering $\mathbf{x}$ as $\mathbf{x}^T = \left[\mathbf{s}^T\ \mathbf{0}^T\right] + \mathbf{c}_\mathbf{x}^T$. It is well known that the erasure-correcting version of the Berlekamp-Massey algorithm has $O(k^2)$ complexity. The decoding complexity plus the $O(n-k)$ intermediate operations required gives the desired result. ∎

The decoding procedure is summarized in Algorithm 1.

As stated in the proof of Theorem 2, classical erasure decoding of an (n, k) Reed-Solomon code has $O(k^2)$ complexity (e.g. [2]). However, several works such as [3]–[5] have shown that the decoder can be modified for erasure-only decoding resulting in significantly lower complexity (e.g. $O(n\log_2^2(n))$ in [4]). The result of [5] could be particularly useful to the problem at hand because their work demonstrates how to prepare a decoder to efficiently decode for a particular erasure pattern. After the initial setup is complete, the decoding procedure has complexity $O((n-k)k+n)$. This type of decoder is applicable to the software update example given in Section I. In this example, the updates might be files consisting of many symbols. Coding would be done across the corresponding symbols in each of the files. Hence, if a particular file (update) is ! missing, the same erasure pattern would be applicable to each of the decoding operations. Of course, all of these complexities depend on n and k, and therefore certain methods may be more efficient than others given particular values of n and k. For this reason, and to avoid confusion by switching between the many possible decoding techniques, we will restrict our attention to classical erasure decoding of Reed-Solomon codes for the remaining discussion.

Algorithm 1 Decoding procedure for a single receiver with k known symbols

```
for i = 1 to n do {Construct ĉ_x}
  if y(i) ≠ ? and i ≤ n − k then
    ĉ_x(i) = y(i) + s(i)
  else
    ĉ_x(i) = y(i)
  end if
end for

Decode ĉ_x → c_x

for i = 1 to n do {Recover x from c_x}
  if i ≤ n − k then
    x(i) = c_x(i) + s(i)
  else
    x(i) = c_x(i)
  end if
end for
```

C. Reed-Solomon Solution When Transmitter Doesn't Know k

Theorem 2 illustrates how using the Berlekamp-Massey algorithm and the structure of the H^T can reduce decoding complexity. The key to the strategy of Theorem 2 is viewing $\mathbf{x}$ as a codeword in an affine shift of $\mathcal{C}^\perp$. The transmitter communicates this shift via $\mathbf{s}$ and the receiver decodes as if it were decoding to a codeword in $\mathcal{C}^\perp$ using standard techniques.

We can extend this logic to the general case where each receiver has some arbitrarily sized subset of the original symbols and this information is unknown to the transmitter. The cost of this extension manifests itself in the computation of the affine shift at each of the receivers.

In the proof of Theorem 2, the affine shift was directly obtained from $\mathbf{s}$ as $\left[\mathbf{s}^T\ \mathbf{0}^T\right]$. In the general case, each receiver needs to solve a triangular system of $n-k_i$ equations (via back-substitution) in order to recover the vector describing the affine shift.

Theorem 3: In the general case, where receiver i knows k_i of the original symbols and k_i is unknown to the transmitter, receiver i can successfully recover $\mathbf{x}$ with decoding complexity $O\left(k_i^2 + (n-k_i)^2\right)$ after receiving $n-k_i$ transmitted symbols.

Lemma 1 suggests that handling a variety of k values requires a sequence of progressively higher-rate MDS codes. Define an encoding matrix such that for each $j \in \{1, 2, \ldots, n\}$, the first j rows of the encoding matrix form the generator matrix of an (n, j) MDS code. This is necessary so that any receiver missing exactly j symbols can recover $\mathbf{x}$ from the first j transmitted symbols. The existence of such a matrix is guaranteed by the following lemma.

Lemma 2: There exists an upper triangular matrix G such that, for each $j = 1, \ldots, n$, the first j rows of G form the generator matrix of an (n, j) MDS code.

Proof: The proof is by construction. Define the following isomorphism, V, between polynomials and vectors: $V(p_0 + p_1 x + \cdots + p_{n-1}x^{n-1}) = [p_{n-1}, \ldots, p_1, p_0]$. The generator polynomial of a (shortened if necessary) (n, j) Reed-Solomon code has the form $g_{(n,j)}(x) = \prod_{\ell=0}^{n-j-1}(x + \alpha^\ell)$, where α is a primitive element. Clearly $g_{(n,j_1)}(x) = g_{(n,j_2)}(x)t(x)$ for an appropriate polynomial $t(x)$ when $j_1 < j_2$. Therefore, defining G as:

$$G = \begin{bmatrix} V(g_{(n,1)}(x)) \\ V(g_{(n,2)}(x)) \\ \cdots \\ V(g_{(n,n-1)}(x)) \\ V(1) \end{bmatrix} \tag{2}$$

gives a matrix, G, whose first j rows form a generator matrix of an (n, j) MDS code for $j = 1, 2, \ldots, n$. Further, for each j, $\deg(g_{(n,j)}(x)) = n - j$ and $g_{(n,j)}(x)$ is monic. Therefore G is upper triangular with 1's on the diagonal. ■

Lemma 2 provides the essential tool to prove Theorem 3.

Proof: [Proof of Theorem 3] Let G be as defined in (2). The matrix G defines the coding strategy and is thus known to the transmitter and the receivers. The transmitter sequentially transmits the symbols $s_j = g_j^T \mathbf{x}$ for $j = 1, 2, \ldots n$, where g_j^T is the j^{th} row of G. Define H_j^T to be the submatrix formed by the first j rows of G. By construction, $H_j^T = [T_j \mid P_j]$, where T_j is an upper triangular matrix and P_j is some $j \times (n - j)$ matrix. Observe that H_j is the parity check matrix for an $(n, n - j)$ Reed-Solomon Code $\mathcal{C}_{(n,n-j)}$. Therefore, $\mathbf{s}_j^T = [s_1, \ldots, s_j]$ is the syndrome of $\mathbf{x}$ with respect to $\mathcal{C}_{(n,n-j)}$. In particular, $\mathbf{x}^T = \left[t_1, \ldots, t_j, \mathbf{0}^T\right] + \mathbf{c}_\mathbf{x}^T$, where $\{t_i\}_{i=1}^j$ are uniquely determined by $\mathbf{s}_j$ and T_j and $\mathbf{c}_\mathbf{x} \in \mathcal{C}_{(n,n-j)}$.

A receiver, missing exactly j symbols recovers $\mathbf{x}$ from the j symbols of $\mathbf{s}_j$ as follows. First, by back-substitution, $\{t_i\}_{i=1}^j$ is computed from T_j and $\mathbf{s}_j$. Then, similar to the proof of Theorem 2, the receiver forms $\hat{\mathbf{c}}_\mathbf{x}$ by adding its known symbols to the corresponding symbols of $\{t_i\}_{i=1}^j$. The receiver can then decode $\hat{\mathbf{c}}_\mathbf{x}$ to $\mathbf{c}_\mathbf{x}$ using Berlekamp-Massey. Finally, the receiver recovers $\mathbf{x}$ as $\mathbf{x}^T = \left[t_1, \ldots, t_j, \mathbf{0}^T\right] + \mathbf{c}_\mathbf{x}^T$.

The complexity $O\left(k_i^2 + (n - k_i)^2\right)$ follows because $O\left((n - k_i)^2\right)$ operations are required to solve the triangular system of equations, and another $O\left(k_i^2\right)$ operations are required to decode $\hat{\mathbf{c}}_\mathbf{x}$ to $\mathbf{c}_\mathbf{x}$ using classical Reed-Solomon erasure decoding methods. ■

The decoding procedure of Theorem 3 is summarized in Algorithm 2.

Algorithm 2 Decoding procedure for receiver i with k_i known symbols

Compute $\mathbf{t}_{n-k_i} = \{t_j\}_{j=1}^{n-k_i}$ by back-substitution:
$\mathbf{t}_{n-k_i} = T_{n-k_i}^{-1}\mathbf{s}_{n-k_i}$

for $j = 1$ to n **do** {Construct $\hat{\mathbf{c}}_\mathbf{x}$}
  if $\mathbf{y}(j) \neq$? and $j \leq n - k_i$ **then**
    $\hat{\mathbf{c}}_\mathbf{x}(j) = \mathbf{y}(j) + \mathbf{t}(j)$
  else
    $\hat{\mathbf{c}}_\mathbf{x}(j) = \mathbf{y}(j)$
  end if
end for

Decode $\hat{\mathbf{c}}_\mathbf{x} \rightarrow \mathbf{c}_\mathbf{x}$

for $j = 1$ to n **do** {Recover $\mathbf{x}$ from $\mathbf{c}_\mathbf{x}$}
  if $j \leq n - k_i$ **then**
    $\mathbf{x}(j) = \mathbf{c}_\mathbf{x}(j) + \mathbf{t}(j)$
  else
    $\mathbf{x}(j) = \mathbf{c}_\mathbf{x}(j)$
  end if
end for

D. A Further Generalization

One possible extension of Theorem 3 is to ask whether we can develop a practical coding scheme such that a receiver with k_i packets can reconstruct the n symbols of $\mathbf{x}$ from *any* subset of transmitted symbols of size $n - k_i$. Again, the answer is affirmative (note that in Theorem 1, the order of the received symbols is of no importance), but this generalization comes at the price of higher complexity.

One way to accomplish this is to take a $(2n, n)$ systematic Reed-Solomon code and encode the information vector $\mathbf{x}$ into $\mathbf{c}_\mathbf{x}^T = [\mathbf{x}^T \mathbf{s}^T]$, where $\mathbf{s}$ is an n-vector of parity symbols. The transmitter sends $\mathbf{s}$, and the i^{th} receiver, upon receiving $n - k_i$ symbols of $\mathbf{s}$, creates $\hat{\mathbf{c}}_\mathbf{x}$, which consists of n erased symbols and n unerased symbols. At this point, it is possible for the receiver to decode $\hat{\mathbf{c}}_\mathbf{x}$ to $\mathbf{c}_\mathbf{x}$ and recover $\mathbf{x}$ by extracting the first n systematic symbols from $\mathbf{c}_\mathbf{x}$.

A second (yet intimately related) method of accomplishing this is by using an $n \times n$ Cauchy matrix for the encoding matrix at the transmitter. Cauchy matrices have the property that any submatrix is non-singular (and also another Cauchy matrix). Using this method, the receiver can eliminate its known symbols from the set of linear equations and invert the remaining $(n - k_i) \times (n - k_i)$ Cauchy matrix to recover the erased symbols. Cauchy matrices have the property that inversion can be performed with quadratic complexity - hence this method is an improvement over standard Gaussian elimination.

The reason why these two methods are closely related is that the parity check bits of a systematic Reed-Solomon generator matrix form a Cauchy matrix [6, p. 75]. Hence, the two methods are fundamentally equivalent - in the first method, we use the Berlekamp-Massey algorithm to essentially invert the Cauchy matrix, in the latter method the inversion is performed directly.

Although a slightly higher complexity (and field size) is required to allow a receiver to recover $\mathbf{x}$ from *any* sufficiently large subset of transmitted symbols, this method may be the most practical. For example, if the transmitter-receiver

channels are not noiseless, but erasure channels with some small erasure probability (like those channels found in the internet), then this method would be preferred.

IV. Conclusion

In this paper, we studied a rate-compatible fountain communication model consisting of a transmitter which possesses a vector of n information symbols and N receivers, each possessing erased versions of the original information vector. The number or locations of the erasures can be different for each receiver and are unknown to the transmitter. We assumed that the i^{th} receiver has k_i known symbols. We gave a brief argument which proves it is possible for the transmitter to design a sequence of transmitted symbols so that, with overwhelming probability for all i, receiver i can successfully reconstruct the information vector after receiving $(1+\epsilon)(n-k_i)$ of the transmitted symbols (ϵ can be taken to be arbitrarily small).

The main contribution of this work was the development of a theoretically optimal, zero-error rate-compatible transmission scheme that works for finite blocklengths and small field sizes. This scheme can be implemented using existing Reed-Solomon decoders and should therefore be of practical interest for scenarios where the system model is applicable (e.g. content distribution over the internet, etc.).

Acknowledgment

This research was supported by Rockwell Collins through contract #4502769987. The authors would like to thank Shubha Kadambe, Kent Benson, and Rob Frank of Rockwell Collins for proposing the investigation that led to this paper.

References

[1] Z. Wang, J. Luo (2009). Fountain Communication using Concatenated Codes. arXiv:0907.2984v2 [cs.IT].

[2] R. Blahut. Algebraic Codes for Data Transmission. Cambridge University Press, 2003.

[3] J. Blomer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, and D. Zuckerman. An xor-based erasure-resilient coding scheme. Technical report, International Computer Science Institute, Berkeley, California, 1995.

[4] F. Didier (2009). Efficient erasure decoding of Reed-Solomon codes. arXiv:0901.1886v1 [cs.IT].

[5] M. Shrestha, L. Xu. Efficient Erasure Decoding for Generalized Reed Solomon Codes. Technical Report: http://nisl.wayne.edu/Papers/Tech/GRS-Dec.pdf, 2007.

[6] S.B. Wicker and V.K. Bhargava (Editors). Reed-Solomon Codes and Their Applications, IEEE Press, 1994.