MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Information Embedding Codes on Graphs with Iterative Encoding and Decoding

Venkat Chandar, Emin Martinian, Gregory Wornell

TR2006-050 June 2006

Abstract

We show that linear complexity, capacity approaching information embedding codes exist for information embedding problems. Specifically, we introduce the double erasure information embedding channel model, and show that in at least some parameter regimes one can achieve rates arbitrarily close to capacity using suitably defined codes on graphs. Furthermore, we show that both encoding and decoding can be implemented with linear complexity by exploiting belief propagation techniques.

IEEE International Symposium on Information Theory, ISIT 2006

Publication History:–

1. First printing, TR2006-050, May 2006

Information Embedding Codes on Graphs with Iterative Encoding and Decoding

Venkat Chandar
Dept. EECS, MIT
Cambridge, MA
Email: vchandar@mit.edu

Emin Martinian
Mitsubishi Electric Research Labs
Cambridge, MA
Email: emin@alum.mit.edu

Gregory W. Wornell
Dept. EECS, MIT
Cambridge, MA
Email: gww@mit.edu

***Abstract*— We show that linear complexity capacity-approaching information embedding codes exist for information embedding problems. Specifically, we introduce the double-erasure information embedding channel model, and show that in at least some parameter regimes one can achieve rates arbitrarily close to capacity using suitably defined codes on graphs. Furthermore, we show that both encoding and decoding can be implemented with linear complexity by exploiting belief propagation techniques.**

I. Introduction

The information embedding problem of channel coding with transmitter side information arises in a number of applications including coding for a memory with defects, broadcast channels, inter-symbol interference channels, multi-antenna channels, and digital watermarking; see, e.g., [11], [12]. There is a growing interest in understanding the complexity required to approach capacity on such channels, and how to design codes with such complexity.

Low density codes on graphs are compelling candidates for the information embedding problem, which has both channel and source coding aspects. Indeed, low density parity check (LDPC) and low density generator matrix (LDGM) codes have particularly attractive characteristics for channel coding [4] and source coding [2], respectively. This paper develops such a class of codes.

In closely related work, such codes have been used to approach capacity of the noiseless broadcast channel [13], [14], but some difficulties remain. For example, [13] requires logarithmic (as opposed to constant) density in the block length while [14] uses an $\mathcal{O}(n^2)$ algorithm. Furthermore, it is unclear how those approaches fare in the presence of channel noise.

In this work, we consider what may be the simplest information embedding channel model whose source and channel coding aspects are both nontrivial. For this channel, which we refer to as the "double-erasure" channel, we construct a class of capacity-approaching linear complexity codes.

II. Channel Model

Consider a general information embedding problem. A channel state vector $\mathbf{s}$ consisting of n symbols from the alphabet $\mathcal{S}$ is selected according to the probability law $p(\mathbf{s})$. The encoder takes as input $\mathbf{s}$ as well as a k-bit message $\mathbf{m}$, and produces a channel input vector $\mathbf{x}$ consisting of n symbols from the alphabet $\mathcal{X}$. The channel takes $\mathbf{x}$ as input and produces a channel output vector $\mathbf{y}$ consisting of n symbols from the alphabet $\mathcal{Y}$ according to the probabilistic channel law $p(\mathbf{y}|\mathbf{x},\mathbf{s})$. Finally, the decoder receives $\mathbf{y}$ — which we sometimes denote as $\mathbf{y}(\mathbf{x},\mathbf{s})$ to indicate the dependence on the channel input and state — and attempts to determine the message $\mathbf{m}$. The goal is to construct systems operating at rates near capacity with low complexity encoders and decoders, with the probability of decoding error vanishing as $n \to \infty$.

Our "double-erasure" information embedding channel of interest is a variant of the "memory with defects" channel model [11]. Specifically,

$$\mathcal{X} = \{0,1\}, \quad \mathcal{S} = \mathcal{Y} = \{0,1,*\}, \tag{1}$$

the state $\mathbf{s}$ is independent and identically distributed (i.i.d.) with

$$p_S(s) = \begin{cases} (1-e_{\mathrm{s}})/2, & s = 0 \text{ or } s = 1 \\ e_{\mathrm{s}}, & s = *, \end{cases} \tag{2}$$

and the channel law is i.i.d. with

$$p_{Y|S,X}(y|s,x) = \begin{cases} e_{\mathrm{c}}, & y = * \\ 1-e_{\mathrm{c}}, & y = x \text{ and } s = * \\ 1-e_{\mathrm{c}}, & y = s \text{ and } s \neq *. \end{cases} \tag{3}$$

Hence, the channel consists of two parts. The input x and s combine to produce

$$v = \begin{cases} x & s = * \\ s & s \neq *, \end{cases} \tag{4}$$

which is erased (i.e., replaced with $*$) with probability e_{c} to produce y.

III. Capacity

The capacity of the double-erasure channel defined in Section II is as follows.

Claim 1: The capacity of the double-erasure channel is

$$C = e_{\mathrm{s}} - e_{\mathrm{c}} + (1-e_{\mathrm{s}})\, H_{\mathrm{B}}(q) - (1-e_{\mathrm{c}}) H_{\mathrm{B}}(q(1-e_{\mathrm{s}})), \tag{5a}$$

This work was supported in part by NSF under Grant No. CCF-0515109, and by HP through the MIT/HP Alliance.

where q satisfies

$$\frac{1-q}{q} = \left(\frac{1-q(1-e_{\rm s})}{q(1-e_{\rm s})}\right)^{(1-e_{\rm c})} \tag{5b}$$

and $H_{\rm B}(\cdot)$ denotes the binary entropy function.

Proof: To verify (5) it suffices to apply the Gel'fand-Pinsker [7] expression for information embedding capacity

$$C = \max_{p(u|s),p(x|u,s)} I(U;Y) - I(U;S).$$

A particular choice of X and U yields (5). Specifically, let the alphabet for U be $\mathcal{U} = \{0,1\}$, let $p(u=0|s=*) = 1/2$, and let $p(u=0|s=0) = p(u=1|s=1) = 1-2q$. Finally, let $X = U$; the resulting marginal distribution for X is symmetric. Optimizing over the choice of q then gives (5).

To verify that these choices of X and U give capacity, it suffices to verify the optimality conditions in [6]. ■

In the sequel we develop coding schemes that can approach rate $R^- = e_{\rm s} - e_{\rm c}$. When $e_{\rm c} \ll e_{\rm s}$, $C \approx R^-$, so our coding schemes come close to capacity in this regime.

IV. Coding Scheme

A coding scheme consists of a sequence of encoding functions[1] $E_n : \mathcal{S}^n \times \{0,1\}^k \mapsto \mathcal{X}^n$ and decoding functions $D_n : \mathcal{Y}^n \mapsto \{0,1\}^k$ for $n = 1, 2, \ldots$.

Definition 1: A coding scheme is *admissible* for the double-erasure information embedding channel if i) $E_n(\mathbf{s},\mathbf{m})_i = \mathbf{s}_i$ whenever $\mathbf{s}_i \neq *$ for all messages $\mathbf{m}$ (cf. (4)); and ii) for a message $\mathbf{m}$ drawn at random, and any $\epsilon > 0$, there are infinitely many n such that $\Pr[D_n(\mathbf{y}(E_n(\mathbf{s},\mathbf{m}),\mathbf{s})) \neq \mathbf{m}] \leq \epsilon$. The *rate* of an admissible coding scheme is defined as $R = \limsup(k/n)$.

Our encoder, illustrated in Fig. 1, is formed by combining an $(n+\tilde{n},\tilde{k})$ LDGM code $\mathcal{C}_1$ and an $(\tilde{n},k)$ LDPC code $\mathcal{C}_2$.[2] A k-bit message $\mathbf{m}$ is encoded into an n-bit channel input $\mathbf{x}$ as a function of the n-bit channel state $\mathbf{s}$ as follows:

1) Encode $\mathbf{m}$ using $\mathcal{C}_2$, obtaining $\mathbf{w} = \mathbf{G}_2 \cdot \mathbf{m}$, where $\mathbf{G}_2$ is the generator matrix for $\mathcal{C}_2$.
2) Use a modified version of belief propagation (BP) [2], [8], [9] to find a codeword of $\mathcal{C}_1$ denoted $(\tilde{\mathbf{s}},\tilde{\mathbf{w}})$ such that $(\tilde{\mathbf{s}},\tilde{\mathbf{w}})$ matches $(\mathbf{s},\mathbf{w})$ in as many non-$*$ positions as possible. For example, this can be implemented by directly applying the ERASURE-QUANTIZE algorithm of [2]. If ERASURE-QUANTIZE fails, randomly assign values to all of the so-called unreserved variables [2], thereby incurring some small number of errors. Then, solve for the reserved variables as if ERASURE-QUANTIZE did not fail.
3) The channel input is the n-bit vector $\mathbf{x}$ where $x_i = s_i$ if $s_i \neq *$, and $x_i = \tilde{s}_i$ if $s_i = *$.

[1]We use the notation $\mathcal{A}^b$ to denote the b-fold Cartesian product of a set with itself and $\mathbf{c}_i$ to denote the ith component of a vector $\mathbf{c}$.

[2]By LDGM and LDPC codes, we mean codes with a graphical representation having $\mathcal{O}(r)$ edges, where r denote the block length. In particular, this definition allows codes that have unbounded maximum degree, as long as the average degree is bounded by a constant independent of r.

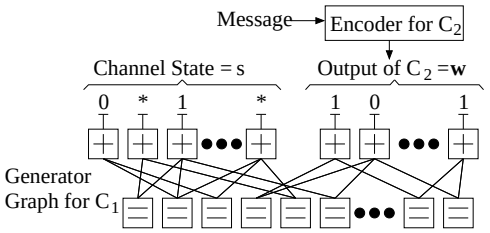

Fig. 1. Encoder structure. The message consists of k bits, which are encoded by code $\mathcal{C}_2$ to produce $\tilde{n}$ outputs labeled $\mathbf{w}$. After concatenating $\mathbf{w}$ onto the n-bit channel state $\mathbf{s}$, the encoder finds a codeword of the main code $\mathcal{C}_1$ that matches $(\mathbf{s},\mathbf{w})$ in as many non-$*$ positions as possible.

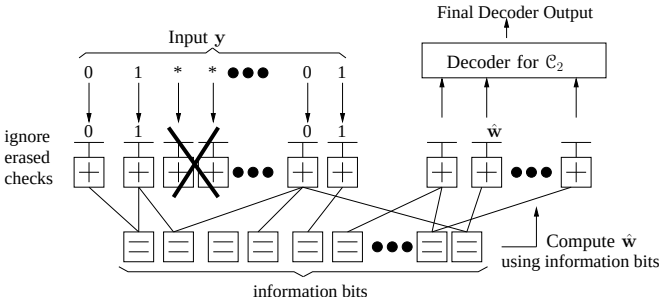

Fig. 2. Decoder Structure. The arrows indicate the flow of information through the decoder from the channel output y to the final decoder output.

To understand the encoding algorithm, it helps to contrast the ideal case where there exists a codeword of $\mathcal{C}_1$ that exactly matches $(\mathbf{s},\mathbf{w})$ in all non-$*$ positions with what actually happens. Usually, there will be at least a few positions of $(\mathbf{s},\mathbf{w})$ that cannot be exactly matched by a codeword of $\mathcal{C}_1$. The encoding algorithm accounts for this in step 3 by changing the positions of $\tilde{\mathbf{s}}$ that do not match the non-$*$ positions of $\mathbf{s}$. The decoder will need to correct these errors in addition to the erasures in the channel output.

Fig. 2 illustrates the decoder for our codes. Decoding a received n-bit channel output $\mathbf{y}$ proceeds as follows:

1) Form the subgraph of $\mathcal{C}_1$ obtained by ignoring the erased positions of the received signal $\mathbf{y}$ and the last $\tilde{n}$ positions of $\mathcal{C}_1$.
2) Use BP on this subgraph as if the vector $\tilde{\mathbf{s}}$ was corrupted by a binary symmetric channel (BSC) to estimate the information bits of $\mathcal{C}_1$. Then, use the information bits to compute an estimate $\hat{\mathbf{w}}$ of $\mathbf{w}$.
3) Decode $\mathcal{C}_2$ to recover the message $\mathbf{m}$ from $\hat{\mathbf{w}}$. [3]

We first discuss the required properties of $\mathcal{C}_1$.

Definition 2: A $\mathcal{C}_1$ code ensemble is *good* if for some choice of $\epsilon_{\rm s},\epsilon_{\rm c},\delta_{\rm s},\delta_{\rm c}$ it is $(\epsilon_{\rm s},\epsilon_{\rm c},\delta_{\rm s},\delta_{\rm c})$-good. The latter is a family of $\mathcal{C}_1$ codes, with a probability distribution over the family, mapping $\tilde{k}$ information bits to $n+\tilde{n}$ code bits, where members $\mathcal{C}_1$ of the ensemble have the following two properties:

1) Erasure Quantization: Let $\mathbf{t} \in \mathcal{S}^{n+\tilde{n}}$ be arbitrary. If the number of $*$ symbols in $\mathbf{t}$ exceeds $n+\tilde{n}-\tilde{k}(1-\epsilon_{\rm s})$,

[3]In practice, one would probably want to use a code $\mathcal{C}_2$ that would allow BP decoding.

then with high probability, there exists a codeword $\mathbf{c} \in \mathcal{C}_1$ such that $|\{i : c_i \neq \mathbf{t}_i \text{ and } \mathbf{t}_i \in \{0,1\} \text{ and } i < n\}| \leq \delta_s n$ and $|\{i : c_i \neq \mathbf{t}_i \text{ and } \mathbf{t}_i \in \{0,1\} \text{ and } i > n-1\}| \leq \delta_s \tilde{n}$.

2) Erasure Correction: Let $\tilde{\mathcal{C}}_1$ denote a punctured version of $\mathcal{C}_1$ that keeps only the first n code bits of every codeword, and let $\tilde{\mathbf{c}} \in \tilde{\mathcal{C}}_1$ correspond to a codeword $\mathbf{c} \in \mathcal{C}_1$. Form $\mathbf{t}$ by changing $\leq n - \tilde{k}(1+\epsilon_c)$ positions of $\tilde{\mathbf{c}}$ to $*$ symbols. With high probability, we can compute a reconstruction $\mathbf{w} = f(\mathbf{t})$ such that $|\{i : \mathbf{c}_i \neq \mathbf{w}_i \text{ and } i > n-1\}| \leq \delta_c \tilde{n}$.

One class of codes $\mathcal{C}_1$ that can meet the conditions of Definition 2 is an LT code [10], to which we restrict our attention for the remainder of the paper. Following the notation from [1], a $(\tilde{k}, \Omega(z))$ LT code is one with $\tilde{k}$ information bits and output degree distribution given by the generator polynomial[4] $\Omega(z)$. For our construction, we use, as in [1], a modified version of the ideal soliton distribution. Specifically, our distribution has generator polynomial

$$\Omega_{\mu,D}(z) = \frac{1}{\mu+1}\left(\mu z + \sum_{i=2}^{D-1} \frac{z^i}{i(i-1)} + \frac{z^D}{D}\right),$$

where we have made the parameters μ and D explicit. We truncate this LT code so that only $n+\tilde{n}$ outputs are produced.

For $\mathcal{C}_2$, we require only that the code 1) be of high-rate, 2) have efficient (linear complexity) encoding and decoding algorithms, and 3) be a good error-correction code. With respect to the latter, we require that the code be capable of correcting a fixed fraction r of errors regardless of the locations of these errors in the received signal. However, we do not require that the code be capacity achieving. As an example, one class of codes $\mathcal{C}_2$ that meets these requirements is that due to Spielman [5].

The parameters of the LT code can be chosen to make it suitable for our application. In particular, let $\epsilon_s > 0$ be arbitrary, and let $\epsilon_c = (2\ln(1/\epsilon_s))^{-1/\epsilon_s}$. In turn, set the $\mathcal{C}_1$ code parameters according to $\mu = \epsilon_s/2 + (\epsilon_s/2)^2$ and $D = \lceil 1/\rho \rceil$, where $\rho = \epsilon_c/(4(1+\epsilon_c))$. Furthermore, let $\delta_s = 10/\Omega'_{\mu,D}(1) < 10/\ln(1/\epsilon_c)$, and let $\delta_c = 2(\tilde{k}/\tilde{n})\rho\Omega'_{\mu,D}(1)$. Then we have the following:

Lemma 1: The ensemble of $(\tilde{k}, \Omega_{\mu,D}(z))$ LT codes truncated to length $n+\tilde{n}$ is an $(\epsilon_s, \epsilon_c, \delta_s, \delta_c)$-good code for $\mathcal{C}_1$.

Finally, this $\mathcal{C}_1$ code, when combined with a suitably parameterized $\mathcal{C}_2$ code, yields an admissible coding scheme for our channel in the sense of Definition 1. Specifically, we have the following as our main result.

Theorem 1: Suppose $\mathcal{C}_1$ is chosen as in Lemma 1, and $\mathcal{C}_2$ is capable of correcting a fraction $r = \delta_s + \delta_c$ of errors. Then for the double-erasure information embedding channel with

$$e_s \geq 1 - \frac{\tilde{k}-\tilde{n}}{n} + \frac{\tilde{k}}{n}\epsilon_s,$$

and

$$e_c \leq 1 - \frac{\tilde{k}}{n}\left(1 + \epsilon_c + \frac{160}{\ln(1/\epsilon_c)}\right),$$

our construction produces an admissible coding scheme with rate $\limsup k/n$.

It follows immediately that our coding scheme is able to achieve rates close to $e_s - e_c$. Specifically, we have:

Corollary 1: For a double-erasure information embedding channel with parameters e_s and e_c, we can choose $k, \tilde{k}, n, \tilde{n}$ to obtain an admissible coding scheme with rate arbitrarily close to $e_s - e_c$.

V. Proofs

In this section, we prove Lemma 1 and Theorem 1.

A. Proof of Lemma 1

To prove Lemma 1, we verify the erasure quantization and erasure correction properties separately.

The erasure quantization property is so named because it essentially requires the code ensemble to be good for the binary quantization problem [2]. To prove that this is true for LT codes, we need the following lemma from [2].

Lemma 2: A linear block code $\mathcal{C}$ can recover from a particular erasure sequence (under ML decoding) if and only if the dual code $\mathcal{C}^\perp$ can quantize the dual sequence, i.e., the sequence where all the erased symbols have been turned into unerased symbols and vice versa. Also, if $\mathcal{C}$ can recover from an erasure sequence using BP decoding, then $\mathcal{C}^\perp$ can quantize the dual sequence using a dualized form of BP.

In Lemma 2, "recover" includes the case where BP decoding can only determine some of the information bits. For a particular erasure sequence, suppose BP decoding can recover l information bits. Then, the dualized form of BP applied to $\mathcal{C}^\perp$ and the dual sequence can quantize the dual sequence such that at least l unerased positions are matched.

Now we prove that $(\tilde{k}, \Omega_{\mu,D}(z))$ LT codes satisfy the erasure quantization property.

Lemma 3: A truncated $(\tilde{k}, \Omega_{\mu,D}(z)))$ LT code with parameters as specified in Section IV matches, with high probability, a fraction $1-\delta_s$ of any subset of $\tilde{k}(1-\epsilon_s)+1$ unerased output symbols.

Proof: From Lemma 2, it follows that to prove Lemma 3 we only need to show that the dual of a truncated $(\tilde{k}, \Omega_{\mu,D}(z))$ LT code is good on the BEC. More precisely, we must show that if all the inputs to the dual code are erased, we can recover all but a δ_s fraction of the erased symbols.

The analysis of the dual code is similar to the proof of [1, Lemma 4]. Let $\omega(z)$ and $\ell(z)$ be the generating functions for the *edge* degree distributions with respect to the variable and check nodes of the dual LT code. [5] From [1], $\omega(z) = \Omega'_{\mu,D}(z)/\Omega'_{\mu,D}(1)$, and $\ell(z) =$

[4]Recall that the probability of a degree i node is specified by the coefficient of z^i in a generator polynomial $\Omega(z)$. Thus the expected degree is given in terms of this polynomial by $\Omega'(1)$.

[5]The dual code can be obtained by replacing all check nodes with variable nodes and vice versa. This follows because the graphical representation of $\mathcal{C}_1$ given in figure 1 shows the generator matrix for $\mathcal{C}_1$. The graphical representation of the dual should use the generator matrix of $\mathcal{C}_1$ as the parity check matrix, and this can be achieved by swapping the variable nodes with the check nodes.

$(1-\Omega'_{\mu,D}(1)(1-z)n(1-e_c))^{n(1-e_c)(1-\epsilon_s)}$. Using the density evolution method, to prove Lemma 3 we must show that $\omega(1-\ell(1-z)) < z, \forall z \in [\delta_s, 1]$; our argument differs from that of [1] only in that [1] proves $\ell(1-\omega(1-z)) < z, \forall z \in [\delta_s, 1]$, i.e., our argument proves that we can interchange ω and ℓ and preserve the inequality.

Now $\omega(1-\ell(1-z)) < z$ reduces to $\Omega'_{\mu,D}(1-\ell(1-z)) < \Omega'_{\mu,D}(1)z$. Using the formula for $\Omega'_{\mu,D}(z)$ given in [1], some algebra shows that $\Omega'_{\mu,D}(1-\ell(1-z)) < \Omega'_{\mu,D}(1)z$ for sufficiently small ϵ_c. In particular, it suffices to choose $\epsilon_c < (\ln(1/\epsilon_s)) + O(1))^{-1/\epsilon_s}$. ∎

Lemma 3 shows that $(\tilde{k}, \Omega_{\mu,D}(z))$ LT codes satisfy the erasure quantization property because the LT code generates every output bit i.i.d. Thus, the unmatched positions are uniformly distributed throughout the $n+\tilde{n}$ output positions, and we can consider the first n positions separately from the last $\tilde{n}$ positions. (In fact, since all the erasures are in the first n positions, the fraction that are incorrect in the first n positions is upper bounded by $\delta_s(1-e_s)$).

The erasure correction property follows from the following lemma, given in [1, Lemma 4].

Lemma 4: With a truncated $(\tilde{k}, \Omega_{\mu,D}(z)))$ LT code with parameters as specified in Section IV one can, with high probability, recover all but a fraction ρ of the $\tilde{k}$ inputs from any subset of $\tilde{k}(1+\epsilon_c/2)+1$ output symbols.

Proof of Lemma 1: Lemma 3 proves the erasure quantization property. To complete the proof of Lemma 1, we need to turn the bound on the number of unrecovered inputs given in Lemma 4 into a bound on the number of unrecovered outputs in the last $\tilde{n}$ positions. With high probability at most $2\tilde{k}\rho$ variable nodes of $\mathcal{C}_1$ are unrecovered (we need the 2 for Lemma 6 to come later). These unrecovered variable nodes induce at most $2\tilde{k}\rho\Omega'_{\mu,D}(1)$ unrecovered check nodes in the last $\tilde{n}$ positions of $\mathcal{C}_1$ with high probability. This is because the number of unrecovered check nodes in the last $\tilde{n}$ positions is upper bounded by $\sum_i \deg(i)$, where $\deg(i)$ is the degree of node i in the subgraph induced by the last $\tilde{n}$ check nodes of $\mathcal{C}_1$, and the sum ranges over all the variables nodes that are in error. Because the check nodes choose their neighbors independently at random, this sum is tightly concentrated around its expected value, which is $\Omega'_{\mu,D}(1)\tilde{n}\rho$. Thus, with high probability we do not see more than $2\tilde{k}\rho\Omega'_{\mu,D}(1)$ unrecovered check nodes in the last $\tilde{n}$ positions.[6] Note that this analysis would hold even if we made errors in the variable nodes instead of just not recovering certain nodes. This is important when we prove Lemma 6 to come. ∎

B. Proof of Theorem 1

We prove, in order, that our construction satisfies both the encoding and decoding properties of an admissible coding scheme.

The former is established by the following Lemma.

Lemma 5: For the choices of $\mathcal{C}_1$, $\mathcal{C}_2$, state, and channel distributions given in Theorem 1, our construction satisfies the encoding property of Definition 1.

Proof: The encoding algorithm given in Section IV guarantees that we satisfy the encoding property, since step 3 of the algorithm ensures that the encoding matches s at all non-$*$ positions. However, we can make a stronger statement than this. Lemma 1 guarantees that a large fraction of the state positions are matched after step 2 on the encoding algorithm. Thus, step 3 only changes a small (δ_s) fraction of unmatched positions to get the final encoding. This will be important when we analyze the decoder. ∎

In the sequel, we refer to the encoding computed after step 2 as the *preliminary encoding*.

Now we prove that our construction satisfies the decoding property. Specifically, we have the following result, whose proof requires us to show that our code can correct the erasures made by the channel, and the errors introduced by the preliminary encoding.

Lemma 6: For the choices of $\mathcal{C}_1$, $\mathcal{C}_2$, state, and channel distributions given in Theorem 1, our construction satisfies the decoding properties of Definition 1.

To prove Lemma 6, we first need the following Lemma, which implies that a truncated version of $\mathcal{C}_1$ can be decoded reliably over BSC(δ_s).

Lemma 7: For the choice of parameters given in Theorem 1, assume that the first n bits of a codeword of $\mathcal{C}_1$ are sent over $\mathrm{BSC}(\delta_s)$, and then over the erasure channel specified in Theorem 1. Then, BP can be used to recover the level 1 variable nodes with high probability, in the sense that at most a fraction ρ of the nodes are not recovered correctly.

In order to prove Lemma 7, we will make use of the following result from [3, Thm. 4.2] relates the performance of a code on the BEC to its performance on any binary input-symmetric channel (BISC). The Bhattacharya parameter of a BISC is defined as $\lambda = E[e^{-L/2}]$, where L is the log-likelihood ratio of the input given the channel output.

Lemma 8: Let $\lambda(\mathcal{C})$ be the Bhattacharya parameter of an arbitrary BISC $\mathcal{C}$. If BP can decode an ensemble of codes over $\mathrm{BEC}(\lambda(\mathcal{C}))$, then BP can also decode reliably over $\mathcal{C}$.

We remark that the proof of Lemma 8 given in [3] actually proves the stronger statement that if the fraction of unrecovered inputs over $\mathrm{BEC}(\lambda(\mathcal{C})) < \delta$, then the fraction of inputs which are recovered incorrectly over $\mathcal{C}$ is also less than δ. [7]

Proof of Lemma 7: We prove that the subgraph of $\mathcal{C}_1$ formed by only considering the positions that were not erased by the erasure channel is good for $\mathrm{BSC}(\delta_s)$. Let $\tilde{\epsilon} > 0$ be a parameter we determine later. Say we receive $\tilde{k}(1+\tilde{\epsilon})$ bits, but a δ_s fraction of these bits are incorrect. From Lemma 4, we know that this subcode of $\mathcal{C}_1$ can recover from erasures provided that $\tilde{k}(1+\epsilon_c/2)$ unerased outputs are available. Thus, we can tolerate an erasure probability of $\tilde{e} = (\tilde{\epsilon} - \epsilon_c)/(1+\tilde{\epsilon})$.

[6] We assume $\tilde{k} \geq \tilde{n}$.

[7] Density evolution typically looks at the values passed along edges of the graph. To turn this into a bound on inputs, it suffices to pretend that each variable nodes has an "extra" edge leaving which is not attached to any other nodes. The value of this edge is updated using the same density evolution equations, and the value on this edge determines the value of the associated variable.

Applying Lemma 8, it follows that if $\delta_{\rm s}$ satisfies $\lambda(\delta_{\rm s}) = 2\sqrt{\delta_{\rm s}(1-\delta_{\rm s})} < \tilde{e}$, then we can correct a $\delta_{\rm s}$ fraction of errors. This inequality is satisfied if we choose $\tilde{\epsilon} = \epsilon_{\rm c} + 160/\ln(1/\epsilon_{\rm c})$. ■

It remains to confirm that $\mathcal{C}_1$ can correct enough of the errors from the preliminary encoding that $\mathcal{C}_2$ can correct those that remain.

Proof of Lemma 6: We first define a new channel $\mathcal{C}_e$, which models the positions whose bits we need to change after the preliminary encoding in order to satisfy the encoding property. Let Γ be the number of unmatched positions after the preliminary encoding, so that $\mathcal{C}_e$ introduces Γ errors to form the final encoding. Because the LT code generates each output symmetrically, and because the state distribution (2) is symmetric, it follows that given $\Gamma = \gamma$, the γ positions flipped by $\mathcal{C}_e$ are equally likely to be any γ positions. Thus, conditioned on the number of errors, $\mathcal{C}_e$ has the same distribution as a BSC.

Lemma 1 guarantees that $\Gamma \leq \delta_{\rm s} n$ with high probability. Let $\overline{\gamma}$ be the expected value of Γ (our proof of Lemma 3 shows $\overline{\gamma} \leq \delta_{\rm s}$, but we can calculate $\overline{\gamma}$ to any desired accuracy using density evolution). Then, a standard martingale argument [4] shows that there exists a constant β such that for any $\epsilon > 0$,

$$\Pr[|\Gamma - \overline{\gamma}| > \epsilon] < e^{-\beta\epsilon^2 n}. \tag{6}$$

[8]

Let $D(\cdot\|\cdot)$ denote the Kullback-Leibler distance between two Bernoulli random variables. For any particular value γ and sufficiently large n, we know that the probability $P_{\gamma,\epsilon}$ that a realization $\mathrm{BSC}(\gamma)$ of the BSC makes, for some $\epsilon > 0$, a fraction $\gamma + \epsilon$ errors in n transmissions is lower bounded by

$$P_{\gamma,\epsilon} \geq \frac{e^{-nD(\gamma+\epsilon\|\gamma)}}{\sqrt{2\pi n}}, \tag{7}$$

and a similar statement is true for $\gamma - \epsilon$.

We say that BP decoding of (a truncated version) of $\mathcal{C}_1$ fails if the fraction of level 1 variable nodes that are not recovered correctly is greater than 2ρ. Using martingale arguments, one can show that Lemma 7 implies

$$\Pr[\text{BP decoding fails for } \mathrm{BSC}(\delta_{\rm s})] \leq e^{-\alpha\rho^2 n}, \tag{8}$$

for a suitable constant $\alpha > 0$. Choosing ϵ such that $\sigma = D(\overline{\gamma} \pm \epsilon||\overline{\gamma}) - \alpha\rho^2 < 0$, then combining (6), (7), (8), and the fact that conditioned on the number of errors, the distribution of a BSC and $\mathcal{C}_e$ is the same, we obtain

$$\Pr[\text{BP decoding fails for } \mathcal{C}_e] \leq e^{-\sigma n}\sqrt{2\pi n} + e^{-\beta\epsilon^2 n}.$$

Thus, the probability that BP decoding of $\mathcal{C}_1$ fails is exponentially small even when the errors are introduced by $\mathcal{C}_e$.

To complete the proof, we need to correct the small fraction of check nodes which are not recovered properly after decoding $\mathcal{C}_1$. There are two sources of error for the last $\tilde{n}$ check nodes: errors caused because our definition of failure allows for a 2ρ fraction of errors in the variable nodes, and errors caused because during encoding we may not be able to match a $\delta_{\rm s}$ fraction of the check nodes. In total, with high probability the two sources of error induce at most a fraction $\delta_{\rm s} + \delta_{\rm c}$ of errors in the last $\tilde{n}$ check nodes, which can be corrected given the choice of $\mathcal{C}_2$. ■

[8] In deriving equation 6, it is important that the encoder uses the algorithm specified in Section IV. Specifically, the unreserved variables need to be assigned randomly. This allows us to conclude that about half of the unreserved checks are not satisfied. Then, we can multiply the density evolution value for the number of unreserved checks by .5 to get $\overline{\gamma}$, and the concentration result in equation 6 follows easily.

VI. Concluding Remarks

We have described a coding scheme for the double-erasure information embedding channel with linear-time encoding and decoding algorithms. The key ingredient in the construction is a code that is good for both BEQ and the BEC. In our construction, because $\epsilon_{\rm c}$ is so small compared to $\epsilon_{\rm s}$, the complexity of encoding and decoding scales with $(1/\epsilon_{\rm s})\ln\ln(1/\epsilon_{\rm s}))$. Thus, our choice of $\Omega_{\mu,D}(x)$ allows us to prove the asymptotic result, but the dependence on $\epsilon_{\rm s}$ makes it difficult to get close to $e_{\rm s} - e_{\rm c}$. We believe that by using techniques similar to those employed in [1] we can find choices for $\Omega(z)$ with a lower value of $\Omega'(1)$, which still perform well for BEQ and BEC. This is the subject of future work.

References

[1] A. Shokrollahi, "Raptor Codes." Available at http://www.inference.phy.cam.ac.uk/mackay/dfountain/RaptorPaper.pdf.

[2] E. Martinian and J. S. Yedidia, "Iterative Quantization Using Codes on Graphs," Allerton Conference on Communications, Control, and Computing, October 2003.

[3] A. Khandekar, *Graph-based Codes and Iterative Decoding.* PhD thesis, California Institute of Technology, 2002.

[4] T. J. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," IEEE Trans. Inform. Theory., vol. 47, pp. 599-618, February 2001.

[5] D. Spielman, "Linear-time encodable and decodable error-correcting codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1723-1731, November 1996.

[6] F. Dupuis, W. Yu, F.M.J. Willems, "Blahut-Arimoto Algorithms for Computing Channel Capacity and Rate-Distortion With Side Information," *IEEE International Symposium on Information Theory (ISIT)*, 2004

[7] S.I. Gel'fand and M.S. Pinsker. Coding for Channel with Random Parameters. *Problems of Control and Information Theory*, vol. 9, no. I, pp. 19-31, 1980.

[8] S. M. Aji and R. J. McEliece, "The Generalized Distributive Law," *IEEE Trans. Inform. Theory*, vol. 46, pp. 325-343, March 2000.

[9] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, Factor Graphs and the Sum-Product Algorithm, *IEEE Trans. Inform. Theory*, vol. 47, pp. 498-519, February 2001.

[10] M. Luby, "LT-codes," in *Proceedings of the 43rd Annual IEEE Symposium on the Foundations of Computer Science (STOC)*, 2002, pp. 271-280.

[11] A. El Gamal and C. Heegard, "On the Capacity of Computer Memory with Defects," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 731-739, September 1983.

[12] U. Erez, S. Shamai, and R. Zamir, "Capacity and lattice strategies for canceling known interference," *IEEE Trans. Inform. Theory*, vol. 51, no. 11, pp. 3820-3833, November 2005.

[13] T. P. Coleman, E. Martinian, M. Effros, and M. Medard, "Interference management via capacity-achieving codes for the deterministic broadcast channel" *Proc. IEEE Information Theory Workshop*, pp. 23-27, September 2005.

[14] W. Yu and M. Aleksic, "Coding for the Blackwell channel: a survey propagation approach," *Proc. IEEE International Symposium on Information Theory*, pp. 1583-1587, September 2005.