Coding with Encoding Uncertainty

Jad Hachem
UCLA, Los Angeles, USA
jadhachem@ucla.edu

I-Hsiang Wang
EPFL, Lausanne, Switzerland
i-hsiang.wang@epfl.ch

Christina Fragouli
EPFL, Lausanne, Switzerland
christina.fragouli@epfl.ch

Suhas Diggavi
UCLA, Los Angeles, USA
suhasdiggavi@ucla.edu

***Abstract*—We study the channel coding problem when errors and uncertainty occur in the encoding process. For simplicity we assume the channel between the encoder and the decoder is perfect. Focusing on linear block codes, we model the encoding uncertainty as erasures on the edges in the factor graph of the encoder generator matrix. We first take a worst-case approach and find the maximum tolerable number of erasures for perfect error correction. Next, we take a probabilistic approach and derive a sufficient condition on the rate of a set of codes, such that decoding error probability vanishes as blocklength tends to infinity. In both scenarios, due to the inherent asymmetry of the problem, we derive the results from first principles, which indicates that robustness to encoding errors requires new properties of codes different from classical properties.**

I. Introduction

In classical channel coding, the goal is to design encoding and decoding algorithms that combat errors and uncertainty introduced by the channel. An implicit yet important assumption is that the encoder and decoder, once designed and implemented, operate in a deterministic and faultless manner. We ask in this paper, what if the encoder itself introduces uncertainty and errors?

Encoder uncertainty can result from several causes. First, defects in a physical device that implements an encoder can make the encoding process faulty. Second, soft errors in processing and storage are becoming more frequent due to the trend in reducing the chip size. Third, as technology scales, variability in transistor design and device degradation also lead to unreliability. Lastly, errors can also happen in distributed encoding across physically separated devices which are connected through noisy channels (as in sensor networks).

The fact that we need to compute from unreliable components has been recognized in the literature, but the focus has mainly been on computing (see, *e.g.*, [1], [2]). Recent work has also looked at the case where decoding might be subject to errors, in particular for message passing algorithms [3], [4]. However, as far as we know, ours is the first work that tries to explore the effect of errors during the encoding process.

As a first step in this direction, we focus in this paper on linear codes, and model unreliability as *erasures* on the *edges* of the factor graph of the generator matrix. As a result, an input message at the source will be mapped to one out of a set of codewords, with a certain probability. Equivalently, we can think of the source as employing one out of a set of encoders, again with a certain probability. The decoder needs to retrieve the message, knowing only the *a priori* probability distribution on the encoders; that is, the goal of the decoder is to recover the original codeword from the output of the defective encoder without knowing the realization of erasures.

Although this is, we believe, the simplest formulation, it is not a simple problem: erasures on the edges of the factor graph lead to bit flips in the codewords, with a nonuniform probability, that highly depends on the mapping from input messages to codewords, that is, on the structure of the generator matrix. Trying to treat these errors as another source of noise, and incorporating them into the noisy channel, results in a channel model where the noise process has memory and depends on the generator matrix and input messages. Moreover, we found that even a small number of edge-erasures can significantly deteriorate the decoding performance. Hence, if such errors occur, it is necessary to design and employ codes that protect against them. We consider edge-erasures as a first step in this paper, but we can look at other error models as well [5].

In this paper, we focus for simplicity on the case where the channel itself does not introduce errors, and make two contributions. First, we take a worst-case approach, in which we find the maximum number of erasures that can occur, regardless of the erasure locations, such that the decoder can always recover the original message. In other words, we characterize the maximum number of erasures that can be tolerated for perfect error correction. To this end, we define a new distance metric between codewords to take the asymmetry of the problem into account. Next, we take a probabilistic approach to analyze a minimum distance decoder that emerges from the worst-case analysis. We assume that each edge is erased with probability p, independently of other edges. We find a sufficient condition on the rate of a code, under which decoding is successful with vanishing error probability as the blocklength tends to infinity. These results lead us to identify code properties that enable robustness to encoding errors.

The rest of this paper is organized as follows. In Section II, we formulate the problem, and summarize the main results in Section III. The worst-case approach and a sketch of its analysis are presented in Section IV, while the probabilistic ones are given in Section V. Because this paper is limited to five pages, Sections IV and V contain only sketches of the analyses and proofs. The detailed proofs can be found in the extended version of the paper [5] available online.

The work of J. Hachem and S. Diggavi was supported in part by NSF award 1136174 and MURI award AFOSR FA9550-09-064. The work of I.-H. Wang was supported by EU project CONECT FP7-ICT-2009-257616. The work of C. Fragouli was supported in part by the ERC Starting Grant Project NOWIRE ERC-2009-StG-240317 and the EU project STAMINA FP7-ICT-2009-265496.

II. Problem Formulation

We denote the generator matrix by $G \in \{0,1\}^{k\times n}$, with rate $R = \frac{k}{n}$. The message we wish to transmit is $m \in \{0,1\}^k$. The set of codewords associated with G is $\mathcal{C}(G) = \{c : c = mG\}$. The word received by the receiver is $r \in \{0,1\}^n$.

Erasures in the generator matrix occur as 1's in G being flipped to 0's. Note that erasures do not always have an effect: for an erasure to affect a certain codeword bit, it must have occured on a bit in G that will be multiplied by a 1 in m, not by a 0. This means that the error in the codeword depends on the message itself, and also on the generator matrix. Moreover, two erasures might cancel each other out, and hence different erasure patterns can give the same r.

We will present two error models, each of which will be used in one of the two approaches of this paper (worst-case and probabilistic).

A. Error Models

1) Worst-case Erasures: In this model, erasures are introduced in G that flip certain 1's in G to 0's. We model this by the addition of what we call an *erasure matrix* $E \in \{0,1\}^{k\times n}$ to G. The received word would then be $r = m(G+E)$ (binary addition) Note that E has to satisfy a certain condition: since the erasures only happen at 1's in G, E can only have 1's at locations where G also has 1's. In other words, $G_{ij} = 0$ implies $E_{ij} = 0$. We denote this condition by $E \in 2^G$ (E can be thought of as being 1 on a subset of the locations where G is 1, hence the power set notation). Moreover, we denote by 2^G_η the subset of such E's that have at most η 1's. Since each 1 denotes an erasure, $E \in 2^G_\eta$ means there are at most η erasures. Effectively these errors alter the generator matrix to $(G+E)$, encoding m as $m(G+E)$ with E unknown to both the encoder and the decoder. The worst-case model allows any erasures with these constraints.

2) Probabilistic Erasures: In the probabilistic erasure model, we assume that each 1 in G has a probability p of being erased, *i.e.*, of becoming a 0, independently of other 1's. We call these errors *erasures*, and we denote their number by η, unless otherwise specified. However, it will be easier to deal with what we will refer to as *bit errors*. These are the bit flips in the codewords themselves. Unless otherwise specified, we denote their number by η'.

Definition 2.1 (Degree): The probability of bit errors depends not only on the erasure probability p, but also on the following notion of 'degrees': given a generator matrix G, we define the *degree* of each column j as $d_j = \sum_{i=1}^k G_{ij}$, *i.e.*, the number of 1's in column j of G, for each $j \in \{1,\ldots,n\}$. We also define the same degree, relative to the message m, as $d_j^{(m)} = \sum_{i=1}^k m_i G_{ij}$. In other words, it is the number of 1's in column G_j that, if flipped, will affect the outcome of the codeword bit j when encoding message m.

For a given message m, we can now compute the probability of a bit error. Bit j will flip if there is an *odd* number of erasures on the $d_j^{(m)}$ bits of column G_j that affect the encoding of m. Since the number of erasures is a binomial random variable with probability p, the probability of a bit flip given a relative degree of $d_j^{(m)} = d$ is:

$$P_d = \sum_{\substack{0\le l\le d\\ l \text{ is odd}}} \binom{d}{l} p^l (1-p)^{d-l} = \frac{1-(1-2p)^d}{2} \quad (1)$$

Note that, for $p \in \left(0, \frac{1}{2}\right)$, P_d increases with d, and approaches $\frac{1}{2}$ from below as d goes to infinity. Therefore, we can simplify things by letting $d^* = \max_j d_j$ (*i.e.*, d^* is the maximum degree of the columns of G) and by using the following bounds: for all $j \in \{1,\ldots,n\}$ and $m \in \{0,1\}^k$,

$$P_{d_j^{(m)}} \le P_{d^*}$$
$$P_{d_j^{(m)}} \ge P_1 = p \quad \text{whenever } d_j^{(m)} \ge 1$$

Finally, note that, given a message m (or, equivalently, its corresponding codeword c), bit errors in each bit of the codeword are independent.

Remark 2.1: One may try to model such an encoder uncertainty as a noisy channel where the channel input is the original codeword and the output is the actual transmitted codeword by the defective encoder. Since the probability of bit errors on each symbol *depends* on the message, that is, the whole input sequence, this channel has memory. More significantly, information bits are not protected by any channel code against such encoding uncertainty.

Due to the asymmetry of the problem, the classical notion of codeword distance is no longer useful. We will next define a new distance metric, which is more relevant to this problem.

B. New Notion of Distance

In classical problems, the (Hamming) distance d_H between codewords plays a critical role in code design and performance. It can be interpreted as the minimum number of bits ($\lceil \frac{d_H}{2} \rceil$) required to 'change' both codewords into the same word. In other words, $\lceil \frac{d_H}{2} \rceil$ is the minimum number of bit flips/errors on one codeword that would cause us to mistake it for the other codeword. For example, if we think of the classical Binary Symmetric Channel, then $\lceil \frac{d_H}{2} \rceil$ is the minimum number of errors that, if occurring on a codeword, would cause us to not distinguish between it and another codeword: $\lceil \frac{d_H}{2} \rceil$ errors could have happened on either codeword and resulted in the same word. This last interpretation will be the most useful to adapt the concept of distance to our problem.

The biggest distinction to be made between our problem and classical problems is that the 'changes' required between codewords are asymmetric. In the classical case, all bits can be flipped or erased, with no distinction. In our case, however, some bits can never flip.

For instance, take the zero codeword. Regardless of what the erasure matrix E is, the zero codeword will always be encoded as $0 \times (G+E) = 0$. Hence, we can never 'change' it into another one. Other codewords, however, can always be encoded as the zero codeword, for instance if $E = G$. So if we are comparing the zero codeword, 0, with another codeword, c, of Hamming weight d_H, then, in order for 0 and c to be 'changed' into the same word, we need only $\lceil \frac{d_H}{2} \rceil$ errors on each in the classical case, but we need d_H bit flips on c (and no bit flips on 0) in our case for the same result.

To capture this phenomenon, we will use a ternary system for the codewords. Notice that, in the computation of a codeword $c = mG$, some zeros result from a sum of only 0's, and others result from a sum of 0's and an even (nonzero) number of 1's. The latter bits flip when an odd number of these 1's are erased. However, since there are no 1's in the sum of the former bits, they will *never* flip.

This leads us to make a distinction between two types of zeros: those that may flip and become 1's, and those that never will. We call the former 'soft zeros' and denote them by '$\bar{0}$', and call the latter 'hard zeros' and denote them by '0'. Hence, all codewords c are in the set $\{0, \bar{0}, 1\}^n$. We will use this important 0-$\bar{0}$ distinction to compute the new distance between any two codewords c and c', denoted by $\eta_0(c, c')$. As a natural extension to the classical distance, we define η_0 as the minimum number of erasures needed to encode both codewords as the same word.

To better describe $\eta_0(c, c')$, we will need to rearrange the bits of c and c' into nine categories, or blocks $\{\mathcal{B}_i, i = 1, \ldots, 9\}$, as seen in Table I.

TABLE I
BLOCKS $\mathcal{B}_i$ GROUPING DIFFERENT PAIRS OF BITS OF c AND c'. $w_i := |\mathcal{B}_i|$.

block ($\mathcal{B}_i$)	1	2	3	4	5	6	7	8	9
bits of c	1	1	1	$\bar{0}$	$\bar{0}$	$\bar{0}$	0	0	0
bits of c'	1	$\bar{0}$	0	1	$\bar{0}$	0	1	$\bar{0}$	0

Each $\mathcal{B}_i$ contains all the bit locations where c and c' are as shown in column i in the table. We denote their number by $w_i(c, c')$, or simply by w_i when there is no ambiguity. So $\mathcal{B}_4$, for example, contains all the bit locations for which c is $\bar{0}$ and c' is 1, and their number is w_4.

Now that we have formulated the problem, we can state the main results of the paper.

III. Main Results

This paper approaches the problem from two different perspectives: a worst-case approach, and a probabilistic approach. The worst-case approach finds the maximum number of erasures that can occur, such that decoding is still perfect, *i.e.,* it is always successful regardless of the erasure locations. The probabilistic approach finds the conditions under which decoding is successful with vanishing error probability (as the blocklength tends to infinity).

A. Main Result: Worst-Case Approach

In the worst case approach, we assume the worst-case erasure error model described in Section II-A1. We use the following decoder:

> Given a received word r, find the *unique* codeword c such that c can be noisily encoded as r for some erasure matrix E. If no such codeword exists, or more than one exist, then declare an error.

For a given G, we wish to find the maximum number $\eta_{\max}$ such that, if no more than $\eta_{\max}$ erasures occur, then the above decoder will *always* decode correctly, *i.e.*, with *zero error probability*. We call that: *perfect decoding*.

Theorem 3.1: For a generator matrix G, the maximum number of erasures allowing perfect decoding is

$$\eta_{\max} = \min_{\substack{c,c' \in \mathcal{C}(G) \\ c \neq c'}} \eta_0(c, c') - 1 \tag{2}$$

where $\eta_0(c, c')$ for any two codewords c and c' is the distance metric described in Section II-B, and equals:

$$\eta_0 = \begin{cases} w_3 & \text{if } w_3 > w_7 + w_2 + w_4 \\ w_7 & \text{if } w_7 > w_3 + w_2 + w_4 \\ \left\lceil \dfrac{w_2 + w_4 + w_3 + w_7}{2} \right\rceil & \text{otherwise} \end{cases}$$

and the w_i's are as defined in Table I. Note that, for small $|w_3 - w_7|$, we have $\eta_0 = \lceil \frac{d_H}{2} \rceil$.

This result, to be proved in Section IV, can be interpreted similarly to that of a classical perfect decoding problem. Indeed, to maximize the number of erasures allowed for perfect decoding, we would want the (new notion of) distances between codewords to be large: we wish to increase the number of bits where two codewords are different (*i.e.,* blocks 2, 3, 4 and 7). However, there is more importance on blocks 3 and 7 than on blocks 2 and 4. This is because mistaking c for c' (resp., c' for c) necessitates that (at least) *all* of the bits of c (resp., c') in block 3 (resp., block 7) flip. Indeed, if r has a 1 in block 3, then we know that the original codeword could not have been c', since the latter is a hard 0 in that same block. On the other hand, if r has a 1 in block 2, then c' is still a possible codeword since it is a soft $\bar{0}$ in that block. Hence, 1-0 bit differences are more favorable than 1-$\bar{0}$ bit differences.

B. Main Result: Probabilistic Approach

In the probabilistic approach, we assume the probabilistic error model described in Section II-A2. We try to find the set of achievable rates R for a given erasure probability p.

Definition 3.1 (Achievable Rate): A rate R is called *achievable* if there exists a set of encoding matrices $\{G_n\}_n$, $G_n \in \{0,1\}^{nR \times n}$, and corresponding decoders, such that the probability of error of these decoders, when encoding messages using G_n, goes to zero as n goes to infinity.

In this paper, we use the following minimum distance (MD) decoder to find an inner bound on the achievable rate:

> Given a word r, find the codeword c that minimizes the distance $\delta(c \to r)$.

The distance δ is defined as $\delta(c \to r) = \sum_{i=1}^{n} \delta(c_i \to r_i)$, and δ for individual bits is explicitly defined in Table II.

Table II can be explained as follows. The distance is infinite if c can never be encoded as r (*i.e.*, if there is a 0 in c that corresponds to a 1 in r). Otherwise, it is, just like with classical distance, the number of bit locations where c and r differ.

Just like $\eta_{\max}$ in Section III-A strongly depends on the w_i's from Table I, so will the conditions on the rate. However, since

TABLE II

THE DISTANCE METRIC δ ON THE BIT LEVEL. EACH CELL REPRESENTS $\delta(c_i \to r_i)$, WHERE c_i IS IN THE TOP ROW AND r_i IS IN THE LEFT COLUMN.

$r_i \backslash c_i$	0	$\bar{0}$	1
0	0	0	1
1	∞	1	0

the w_i's are linear functions of n, and we are taking a set of G_n's with increasing n, we consider their normalized versions: $\alpha_i^{(n)}(c,c') = \frac{w_i(c,c')}{n}$. When there is no ambiguity, we simply denote these values by α_i.

The new distance metric introduced and used in the worst-case approach will also be useful in the probabilistic approach. However, it will be more convenient to define a 'directed' version of it. Specifically, whereas $\eta_0(c,c')$ is the minimum number of erasures required to mistake one of c or c' for the other, we define $\eta_0(c \to c')$ as the minimum number of erasures required to mistake c for c', and *not* the other way around. So the two quantities are related by: $\eta_0(c,c') = \min\{\eta_0(c \to c'), \eta_0(c' \to c)\}$

Lemma 3.1: Given two codewords c and c', the minimum number of erasures required to mistake c for c', $\eta_0(c \to c')$, is:

$$\eta_0 = \begin{cases} \infty & \text{if } w_7 > w_3 + w_2 + w_4 \\ w_3 & \text{if } w_3 > w_7 + w_2 + w_4 \\ \dfrac{w_2 + w_4 + w_3 + w_7}{2} & \text{otherwise} \end{cases}$$

The following theorem gives the sufficient condition on the rate. It will be useful, for the theorem as well as for its proof, presented in Section V and in [5], to introduce the following function ψ: for any number $x \geq 0$, define $\psi(x) = x^x$.

Theorem 3.2: Given a set of encoder matrices $\{G_n\}_n$ of rate R, a sufficient condition for a vanishing decoding error probability, using the MD decoder, is that, for all n:

$$\forall (c,c') \in \mathcal{C}^2(G_n) \text{ s.t. } c \neq c', \quad R < -\log_2 \beta(c \to c') \quad (3)$$

where $\beta(c \to c')$ is a term that depends on the two codewords c and c', with respect to the structure of G_n, specifically on the α_i's for this particular pair (c,c'). Equivalently, the sufficient condition can be rewritten as $R < -\log_2 \beta_{\max}^{(n)}$, $\forall n$, where $\beta_{\max}^{(n)}$ is the maximum β over all pairs (c,c') in $\mathcal{C}^2(G_n)$, with $c \neq c'$. For one such pair (c,c'), let $\alpha_0 = \frac{\eta_0(c \to c')}{n}$, $\gamma = (\alpha_2 + \alpha_4 + \alpha_1 + \alpha_5)$, and $\alpha^* = \alpha_3 + \frac{P_{d^*}}{P_{d^*}+1-p}\gamma$, and take d^* to be the maximum degree of the columns of G. Then:

$$\beta = \begin{cases} 0 & \text{if } \alpha_0 = \infty \\ P_{d^*}^{\alpha_3}(1-p)^{\alpha_6} \times (P_{d^*} + 1 - p)^{\gamma} & \text{if } \alpha^* \geq \alpha_0 \\ P_{d^*}^{\alpha_3}(1-p)^{\alpha_6} \times \tilde{\beta} & \text{otherwise} \end{cases} \quad (4)$$

with

$$\tilde{\beta} = P_{d^*}^{\frac{\alpha_2+\alpha_4+\alpha_7-\alpha_3}{2}}(1-p)^{\alpha_1+\alpha_5+\frac{\alpha_2+\alpha_4-\alpha_7+\alpha_3}{2}} \times \frac{\psi(\alpha_1+\alpha_5+\alpha_2+\alpha_4)}{\psi\left(\frac{\alpha_2+\alpha_4+\alpha_7-\alpha_3}{2}\right) \times \psi\left(\alpha_1+\alpha_5+\frac{\alpha_2+\alpha_4-\alpha_7+\alpha_3}{2}\right)}$$

The $\tilde{\beta}$ term is a constant, $\tilde{\beta} < 1$, that arises when α_0 (equivalently, η_0) is large (but finite).

We now analyze $\beta(c \to c')$ for a *particular* (ordered) pair (c,c'), and give an intuitive explanation of its expression. Having large values for α_3 and α_6 will result in a smaller β, which is favorable. This makes sense, since, for codeword c to be mistaken for codeword c', all of the bits of block 3 have to flip, and all of those of block 6 must stay the same. (This intuitively explains the presence of $P_{d^*}^{\alpha_3}(1-p)^{\alpha_6}$ in the expression for β). Increasing α_3 and α_6 hence reduces the probability that this happens. On the other hand, assuming these two values are fixed, then having a large α_0 is also favorable, as it multiplies the whole expression by $\tilde{\beta} < 1$, and divides it by $(P_{d^*} + 1 - p)^{\gamma} > 1$. This also makes sense, as a large α_0 means a larger number of bits required to mistake c for c'. Finally, an infinite α_0, which is equivalent to 'no errors are possible', results in $\beta = 0$, or $R < \infty$, *i.e.,* regardless of the rate, c will not be mistaken for c'.

However, reducing β for one pair of codewords might increase it for another pair. Therefore, designing a code minimizing the maximum such β, for an overall good performance, is not straightforward, and is subject to further investigation.

C. Discussion: Comparing the two approaches

Both approaches indicate that having a large number of bits in blocks 3 and 7 is favorable (note that block 7 for the ordered pair (c,c') is block 3 for (c',c)). Moreover, they both indicate that having 1-0 differences between codewords is more favorable than having 1-$\bar{0}$ differences. However, the probabilistic approach also indicates that block 6 (and, by symmetry, block 8) should be large. This means that $\bar{0}$-0 differences are also important. It also takes into consideration, in the expression of $\tilde{\beta}$, blocks 1 and 5, where the bits are equal.

So, whereas the worst-case approach only considers which bits are different, the probabilistic approach also looks at which bits are equal. The results in Theorems 3.1 and 3.2 give guidelines for codes that are robust to worst-case and probabilistic encoding errors respectively. This suggests that not all codes designed for transmission errors may be robust to encoder errors. We are currently investigating code designs based on these ideas.

IV. Worst-Case Approach

Recall the error model described in Section II-A1, where the erasures can be thought of as a use of a different codebook than $\mathcal{C}(G)$. Intuitively, perfect decoding can be ensured when these different codebooks do not contradict. To make this idea more rigorous, we introduce the notion of codebook ambiguity, which captures the nature of errors in the worst-case approach.

A. Codebook Ambiguity

Definition 4.1 (Codebook Ambiguity): We say that two generator matrices G_1 and G_2 have ambiguous codebooks, denoted by $\mathcal{A}(G_1, G_2) = 1$, if $\exists m_1, m_2 \in \{0,1\}^k$ such that $m_1 \neq m_2$ and $m_1 G_1 = m_2 G_2$.

Since we will mostly be dealing with matrices of the form $(G+E)$, we write $\mathcal{A}^{(G)}(E_1,E_2) = \mathcal{A}(G+E_1, G+E_2)$.

Now we generalize this definition to all matrices $\{G+E : E \in 2^G_\eta\}$ for some η. We use the following notation:

$$\mathcal{A}^{(G)}(\leq \eta) = \bigvee_{\substack{E_1,E_2\in 2^G_\eta \\ E_1\neq E_2}} \mathcal{A}^{(G)}(E_1,E_2)$$

In other words, $\mathcal{A}^{(G)}(\leq \eta)$ is 1 if and only if there are distinct $E_1, E_2 \in 2^G_\eta$ such that $(G+E_1)$ and $(G+E_2)$ have ambiguous codebooks.

The following lemma relates codebook ambiguity with perfect decoding:

Lemma 4.1: Given an encoder matrix G, its $\eta_{\max}$, *i.e.*, the maximum number of erasures such that perfect decoding is possible, is the largest integer such that $\mathcal{A}^{(G)}(\leq \eta_{\max}) = 0$.

The lemma is useful as it gives us a method of computing $\eta_{\max}$, using codebook ambiguity. It follows directly from the definition of codebook ambiguity. Proof details are left for the extended version of this paper [5].

B. Proof of Theorem 3.1 (sketch)

We present a sketch of the proof, leaving the details for [5].

The proof follows a simple procedure. We seek to find the minimum number of erasures η such that $\mathcal{A}^{(G)}(\leq \eta) = 1$, *i.e.*, such that there exist distinct m_1 and m_2, and distinct E_1 and E_2 with weight at most η, such that $m_1(G+E_1) = m_2(G+E_2)$. To do so, we go over all pairs of codewords (c, c') (equivalently, pairs of messages (m, m')) and compute $\eta_0(c, c')$: the minimum η such that $m(G+E_1) = m'(G+E_2)$ for some E_1, E_2 with weights at most η. Hence, $\mathcal{A}^{(G)}(\leq \eta_0(c, c')) = 1$ for all (c, c'), and $\mathcal{A}^{(G)}(\leq (\min_{c,c'} \eta_0(c, c') - 1)) = 0$. Therefore,

$$\eta_{\max} = \min_{\substack{c,c'\in\mathcal{C} \\ c\neq c'}} \eta_0(c, c') - 1 \tag{5}$$

To compute $\eta_0(c, c')$, we seek to count the minimum possible number of erasures that would confuse c and c'. Specifically, we focus on bit errors on c and c'. Since we only care about the *minimum* number of erasures in G that will cause a decoding error, each erasure has to correspond to exactly one bit error. So if u is the number of bit errors on c, and u' the number of bit errors on c', such that c and c' both become the same word r, then η_0 will be the minimum value of $\max\{u, u'\}$ over all such (u, u') pairs.

As we can see in [5], following these steps will give us the value of η_0 expressed in Theorem 3.1.

V. Probabilistic Approach

Assuming the probabilistic error model of Section II-A2, we use the minimum distance (MD) decoder presented in Section III-B to find an inner bound on the achievable rate. We will give a sketch of the analysis of the error probability of the MD decoder, which is itself the proof of the inner bound on the rate, described in Theorem 3.2. The details of the analysis are given in the extended version of this paper [5].

A. Analysis of the Probability of Error of the MD decoder (sketch)

An error occurs when we encode a codeword c into a word r, and decode r as c'. We call this event a c-to-c' error, denoted by $e(c \to c')$.

The probability of a c-to-c' error, *i.e.*, the probability of decoding c' given that we encoded c, is denoted by $P^{e(c\to c')}$. The average probability of error P_e can be bounded by a function of $P^{e(c\to c')}$ as follows:

$$P_e \leq 2^{nR} \max_{\substack{c,c' \\ c\neq c'}} P^{e(c\to c')} \tag{6}$$

Note that the same bound applies for the maximal error probability (over all codewords c), since we bounded P_e by the maximum error probability given c and c'.

We will show that $P^{e(c\to c')}$ can be bounded by:

$$P^{e(c\to c')} \leq n(n+1)C \times (\beta(c \to c'))^n$$
$$\max P^{e(c\to c')} \leq n(n+1)C \left(\beta^{(n)}_{\max}\right)^n \tag{7}$$

where C is a constant, and $\beta(c \to c')$ depends only on the properties of the pair (c, c'). We also define $\beta^{(n)}_{\max}$ as the maximal β over all pairs (c, c') in $\mathcal{C}^2(G_n)$, with $c \neq c'$. We can thus rewrite the bound on P_e as:

$$P_e \leq n(n+1)C \left(2^R \beta^{(n)}_{\max}\right)^n \tag{8}$$

This allows us to formulate the following sufficient condition for P_e to decay to zero as $n \to \infty$:

$$\forall n: \quad R < -\log_2 \beta(c \to c'), \quad \forall (c, c') \text{ distinct} \tag{9}$$

or, equivalently, $R < -\log_2 \beta^{(n)}_{\max}$ for all n. Indeed, if (9) is true, then $2^R \beta^{(n)}_{\max} < 1$ and $P_e \to 0$ as $n \to \infty$.

The rest of the proof, as shown in [5], consists of finding the expression of $P^{e(c\to c')}$ to deduce the expression of $\beta(c \to c')$. We begin by providing a combinatorial expression for $P^{e(c\to c')}$ by conditioning over the number of bit errors η', and averaging over that. Then, we use Stirling's approximation to provide an upper bound for $P^{e(c\to c')}$. Finally, we use some simple optimization techniques to bound the joint probability of the event "c-to-c' error and η' bit errors" by its maximum over η'. The resulting expression has the form in (7), with the value of β as given in Theorem 3.2.

Following these steps proves Theorem 3.2, giving an inner bound on the achievable rate of our problem.

References

[1] W. H. Pierce, *Failure-tolerant computer design*. Academic Press, 1965.

[2] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, pp. 10–16, November-December 2005.

[3] L. R. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Transactions on Information Theory*, vol. 57, pp. 4427–4444, July 2011.

[4] N. Ruozzi, "Convergent message-passing algorithms in the presence of erasures," *Proceedings of Allerton Conference on Communication, Control, and Computing*, October 2012.

[5] J. Hachem, I.-H. Wang, C. Fragouli, and S. Diggavi, "Coding with encoding uncertainty," *ArXiv Preprint*, May 2013. arXiv:1305.3733 [cs.IT], http://arxiv.org/abs/1305.3733