Performance of LDPC Decoders with Missing Connections

Linjia Chang, Avhishek Chatterjee, and Lav R. Varshney

Abstract

Due to process variation in nanoscale manufacturing, there may be permanently missing connections in information processing hardware. Due to timing errors in circuits, there may be missed messages in intra-chip communications, equivalent to transiently missing connections. In this work, we investigate the performance of message-passing LDPC decoders in the presence of missing connections. We prove concentration and convergence theorems that validate the use of density evolution performance analysis. Arbitrarily small error probability is not possible with missing connections, but we find suitably defined decoding thresholds for communication systems with binary erasure channels under peeling decoding, as well as binary symmetric channels under Gallager A and B decoding. We see that decoding is robust to missing wires, as decoding thresholds degrade smoothly. Moreover, there is a stochastic facilitation effect in Gallager B decoders with missing connections. We also compare the decoding sensitivity with respect to channel noise and missing wiring.

Index Terms

decoding, density evolution, low-density parity-check (LDPC) codes, stochastic facilitation, wiring

I. INTRODUCTION

Low-density parity-check (LDPC) codes are prevalent due to their performance near the Shannon limit with message-passing decoders that have efficient implementation [1]. With the end of CMOS scaling nearing, there is interest in nanoscale circuit implementations of decoders, but this introduces concerns that process variation in manufacturing may lead to interconnect patterns different than designed [2]–[4], especially under self-assembly [5], [6]. Yield on manufactured chips deemed perfectly operational is small, leading to rather expensive industrial waste [7], but changing the paradigm of circuit functionality from perfection to some small probability $\alpha$ of missing wires may eliminate much wastage. It is of interest to characterize chips with permanently missing connections so that suitable error tolerances may be determined.

Process variation in manufacturing also causes fluctuation in device geometries which might prevent them from meeting timing constraints [8]. Such timing errors lead to missed messages in intra-chip communications, equivalent to transiently missing connections. It is also of interest to characterize decoders with transiently missing connections.

However most fault-tolerant computing research assumes the circuit is constructed correctly and is concerned only with faults in computational elements. Peter Elias noted the following [9], but it remains true today.

J. Von Neumann has analyzed computers whose unreliable elements are majority organs-crude models of a neuron. Shannon and Moore have analyzed combinational circuits whose components are unreliable relays.

Both papers assume that the wiring diagram is correctly drawn and correctly followed in construction, but that computation proper is performed only by unreliable elements.

Such assumptions of fault-free circuit construction need to be reevaluated and performance analysis of computation with such wiring faults needs to be carried out. The only work we are aware of in fault-tolerant computing theory that briefly discusses wiring errors is the monograph of Winograd and Cowan [10, Ch. 9.2].

We had previously extended the method of density evolution to decoders with faults in the computational elements and showed that it is possible to communicate with arbitrarily small error probability with noisy Gaussian belief propagation [11]. Asymptotic characterizations were also determined for Gallager A [11] and Gallager B decoders.

This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

L. Chang, A. Chatterjee, and L. R. Varshney are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: {lchang10, avhishek, varshney}@illinois.edu).
Fig. 1. Tanner graph of a ($3, 6$) regular LDPC code, with a missing wire for a corresponding message-passing decoder highlighted with a dashed line.

with transient noise [12]–[14], and both permanent and transient noise [15]. Noisy decoding [16]–[19], and general noisy belief propagation, not necessarily in decoding [20], [21], have also been studied.

Rather than noise in computational elements, here we analyze the performance of message-passing decoders with missing connections and show that appropriately defined decoding thresholds are robust, in the sense of degrading smoothly. This is true for both transiently and permanently missing connections in message-passing decoding circuits. However in certain settings, missing connections actually improve the performance, resulting in stochastic facilitation (cf. [22]).

The celebrated results of Richardson and Urbanke [23] developed density evolution for analyzing message-passing decoders for LDPC codes that are correctly wired. Here we extend those results, so we can use the density evolution technique to characterize symbol error rate $P_e$, measuring the fraction of incorrectly decoded symbols at the end of message-passing decoding, even when the decoder has missing connections. We also show that the performance of decoders with transiently and with permanently missing connections are asymptotically equivalent in blocklength. Traditionally [23], there are thresholds for channel noise level $\varepsilon$ below which $P_e$ can be driven to 0 with increasing blocklength $n$. Unfortunately with missing connections in message-passing decoders, $P_e$ cannot be driven to 0 in general. Thus, following [11], we let $\eta$ be an upper bound to the final error probability that can be achieved by decoders with missing connections after many iterations $\ell$ and give thresholds to $\varepsilon$, below which $\lim_{\varepsilon \to \infty} P_e(\ell) \leq \eta$ under density evolution.

Sec. II elaborates the background of this work, including models of code, channel, and LDPC decoders with both transiently and permanently missing connections. Sec. III develops tools including concentration and convergence theorems that provide validity to density evolution analysis of symbol error rate. Secs. IV, V, and VI analyze the peeling decoder on the binary erasure channel (BEC) and the Gallager A and Gallager B decoders on binary symmetric channel (BSC) using density evolution, characterizing $P_e$ with missing connections. Sec. VII concludes the paper by outlining directions for further investigation.

II. BACKGROUND

In this section we describe the problem of message-passing LDPC decoders with missing connections. We define the code and channels considered in this work and construct fault-free and missing-wire decoder models for characterization in further sections.

A. Ensemble of LDPC Codes and Channel

We are concerned with the standard LDPC code ensemble $G^n$, both regular and irregular. First consider $G^n(d_v, d_c)$-regular LDPC codes of length $n$, which can be defined by a bipartite Tanner graph with $n$ variable nodes of degree $d_v$ in one set, and $nd_v/d_c$ check nodes of degree $d_c$ in the other set. For irregular codes $G^n(\lambda, \rho)$, the degree distribution of variable and check nodes are denoted by functions $\lambda(x) = \sum_{d=2}^{\infty} \lambda_d x^{d-1}$ and $\rho(x) = \sum_{d=2}^{\infty} \rho_d x^{d-1}$, where $\lambda_d$ and $\rho_d$ specify the fraction of edges in the graph that are connected to nodes with degree $d$. The variable nodes hold the codeword messages, and the check nodes enforce the constraints among variable nodes according to the code design. We consider this binary linear code ensemble as defined over the alphabet $\{\pm 1\}$. Although
results in this section are general, for convenience, let us think of the communication channel as either BSC with output alphabet \{±1\} or BEC with output alphabet \{±1, ?\}.

### B. Fault-Free Message-Passing Decoder

The decoder operates by passing messages iteratively over the edges in the Tanner graph of the code. The implementation of such message-passing decoders in hardware follows the construction of the same Tanner graph too. We define a variable-to-check node message \( u_{v \rightarrow c} \) and a check-to-variable node message \( u_{c \rightarrow v} \). Message \( u_{v' \rightarrow c'} \) from variable node \( v' \) to check node \( c' \) is often computed based on all incoming \( u_{c \rightarrow v'} \) messages, where \( c \in N(v') \) is a neighboring node of \( v' \) and \( c \neq c' \). For peeling, Gallager A, and Gallager B decoders, message \( v_{c' \rightarrow v'} \) from check node \( c' \) to variable node \( v' \) is the product of all incoming \( u_{v \rightarrow c'} \) messages, where \( v \in N(c') \) is a neighboring node of \( c' \) and \( v \neq v' \).

### C. Missing Connections

For notational convenience, let us restrict attention to decoders with messages in \{±1, ?\}, but again concentration and convergence results demonstrated in Sec. III are general.

Now we introduce missing connections between check nodes and variable nodes, as depicted in Fig. 1. Connections may go missing either permanently due to manufacturing defects or transiently due to failures of intra-chip communication. For a given decoder circuit, permanent failure is modeled by removing each connection between variable and check nodes with probability \( \alpha \) independently from others, before decoding starts. These connections are never active once removed. On the contrary, with transiently missing wires, each connection is removed independently from others with probability \( \alpha \) at each decoding iteration.

Our conversations with circuit designers suggest that when an interconnect is broken in LDPC decoders, the measured voltage at this open-ended wire is neither low nor high in circuit signals; it is some undefined floating value that may vary within a range. So, whenever there is a missing connection between a variable node and a check node, we assume that the decoder has knowledge of the missing connection, and an erasure symbol “?” is exchanged instead. Though motivated by different concerns, there are similarities between our model of missing connections and other decoder models [24, Example 4.86], [25].

### III. PERFORMANCE ANALYSIS TOOLS

We now present mathematical tools to simplify the performance analysis of LDPC decoders with missing connections. In particular, we establish symmetry conditions for binary codes for easy analysis, and concentration and convergence results that endow the density evolution method with significance. Such results can not only be applied to LDPC codes with binary alphabet, but also non-binary codes.

#### A. Restriction to All-One Codeword

Under certain symmetry conditions of the code, the communication channel, and the message-passing decoder, the probability of error is independent of the transmitted codeword. These conditions are:

- **C1. Code Symmetry:** Code is a binary linear code.
- **C2. Channel Symmetry:** Channel is a binary memoryless symmetric channel [24, Def. 4.3 and 4.8].
- **C3. Check Node Symmetry:** If incoming messages of a check node are multiplied by \( \{b_i \in \{±1\}\} \), then the computed message is multiplied by \( \prod_i b_i \).
- **C4. Variable Node Symmetry:** If the sign of each incoming message is flipped, the sign of the computed message is also flipped.

**Proposition 1:** Under conditions C1–C4, in the presence of transiently or permanently missing connections, the probability of error of a message passing decoder is independent of the transmitted codeword.

**Proof:** First consider mapping the erasure message “?” sent when a connection is missing, to 0; thus the check-to-variable and variable-to-check messages are the messages computed at check node and variable node, respectively, multiplied by either 1 (connection exists) or 0 (missing connections). It is easy to see that messages passed between check and variable nodes satisfy the respective symmetry conditions [24, Def. 4.82]. Hence, the result follows by invoking [24, Lem. 4.92].

In the sequel, we restrict the analysis of all models to the all-one codeword.
B. Concentration around Ensemble Average

We now show that the performance of LDPC codes decoded with missing-connection decoders stays close to the expected performance of the code ensemble for both transiently and permanently missing wires. The approach follows [23] and is based on constructing an exposure Martingale, obtaining bounded difference constants, and using Azuma’s inequality.

Fix the number of decoding iterations at some finite $\ell$ and let $Z$ be the number of incorrect values held among all $d_v n$ variable nodes at the end of $\ell$th iteration for a specific choice of code, channel noise, and a decoder with missing wires. Let $E[Z]$ denote the expectation of $Z$. Note that the theorem below holds for decoders with both transiently and permanently missing connections.

Theorem 1 (Concentration around Expected Value): There exists a positive constant $\beta = \beta(d_v, d_c, \ell)$ such that for any $\varepsilon > 0$,

$$\Pr[|Z - E[Z]| > nd_v \varepsilon / 2] \leq 2e^{-\beta \varepsilon^2 n}.$$

Proof: The full proof of this theorem can be found in Appendices B and C for both permanently and transiently missing connections, respectively. Recall the Doob’s Martingale construction from [23], and the bounded difference constants for exposing channel noise realizations and the realized code connections, together with Azuma’s inequality. The main difference between the Martingale construction and [23] is in the bounded differences due to the additional randomness from missing connections.

For permanently missing connections, one can think of the final connection graph being sampled from an ensemble of irregular random graphs with binomial degree distribution with average degrees $(1 - \alpha)d_c$ and $(1 - \alpha)d_v$, bounded by maximum degrees $d_c$ and $d_v$. Hence, the result follows from the result for correctly-wired irregular codes [23].

For transiently missing connections, the Martingale is constructed differently. Here instead of edges, for $\ell$ iterations, we sequentially expose the realization of edges at different iterations. Similar to [11] for transient noise in computational elements, the Martingale difference is bounded using the maximum number of edges over which a message can propagate in $\ell$ iterations, by unwrapping a computation tree.

Note $\beta$ will be larger for transient than permanent miswiring. The theorem extends directly to irregular LDPC codes.

C. Cycle-Free Case

We now show that the average performance of an LDPC code ensemble converges to an associated cycle-free tree structure, unwrapping a computation tree as in [23].

For an edge whose connected neighborhood with depth $2\ell$ is cycle-free, let $q$ denote the expected number of incorrect values held along this edge at the end of $\ell$th decoding iteration. The expectation is taken over the choice of code, the messages received from the channel, and the realization of the decoder with missing wires. The theorems hold for both transiently and permanently missing wires.

Theorem 2 (Convergence to Cycle-Free Case): There exists a positive constant $\gamma = \gamma(d_v, d_c, \ell)$ such that for any $\varepsilon > 0$ and $n > \frac{2\varepsilon^2}{\gamma}$,

$$|E[Z] - nd_v q| < nd_v \varepsilon / 2.$$

Proof: The proof is identical to [23, Thm. 2].

The basic idea of the proof is to show the probability of repeats in the computation tree goes to zero with increasing graph girth. Note that if a fault-free code ensemble converges to an associated cycle-free tree structure, introducing missing wires in this cycle-free tree structure does not change its cycle-free property.

Theorem 3 (Concentration around Cycle-Free Case): There exists positive constants $\beta = \beta(d_v, d_c, \ell)$ and $\gamma = \gamma(d_v, d_c, \ell)$ such that for any $\varepsilon > 0$ and $n > \frac{2\varepsilon^2}{\gamma}$,

$$\Pr[|Z - nd_v q| > nd_v \varepsilon] \leq 2e^{-\beta n \varepsilon^2}.$$

Proof: Follows directly from Thms. [1] and [2].

This tree-ensemble concentration result holds for all message-passing decoders with missing connections. In the sequel, we consider the special cases of peeling, Gallager A, and Gallager B decoders.
D. Density Evolution

With the concentration around the cycle-free case, it is clear that the symbol error rate \( P_e \) of message-passing decoders with missing connections can be characterized with the density evolution technique. Let \( \bar{P}_e^{(\ell)}(g, \varepsilon, \alpha) \) be the error probability of decoding a code \( g \in G^n \), after the \( \ell \)-th iteration, where \( \varepsilon \) is the channel noise parameter, and \( \alpha \) is the decoder missing wire probability. Density evolution evaluates the term:
\[
\bar{P}_e^{(\ell)} = \lim_{n \to \infty} E[\bar{P}_e^{(\ell)}(g, \varepsilon, \alpha)].
\]
The expectation is taken over the choice of code, channel noise realization, and missing wire realization.

Based on the proof of Thm. 2, we want to show that the decoding error probability at any iteration \( \ell \) for transiently and permanently missing connections, \( \bar{P}_{eT}^{(\ell)} \) and \( \bar{P}_{eP}^{(\ell)} \), become identical with the increase of the girth as blocklength \( n \) increases. In particular, in density evolution the state variable \( x_{\ell+1} \) is computed based on the \( x_{\ell} \) values of nodes immediately below in the infinite tree. Each connection in the tree is encountered only once, and in case of permanent failure each connection is present with probability 1 - \( \alpha \).

**Theorem 4:** For any arbitrarily small \( \delta = \delta(d_e, d_c, \ell) \), \( \sigma > 0 \), and \( \ell \geq 0 \), the following is true:
\[
\Pr[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma] \leq \delta.
\]

**Proof:** First, let \( N^{2\ell}_e \) be the neighborhood of an edge \( \vec{e} \) with depth 2\( \ell \) in the decoding graph. Define the event \( A_N \) as \( N^{2\ell}_e \) is not tree-like. It is shown that for a positive constant \( \tau = \tau(d_e, d_c, \ell) \), \( \Pr[A_N] \leq \frac{\tau}{n} \) [23, Thm 2]. It implies that the probability of exposing an edge multiple times decreases with increasing blocklength \( n \) at any iteration \( \ell \). Following the edge exposing procedure, \( \bar{P}_{eT}^{(\ell)} \) and \( \bar{P}_{eP}^{(\ell)} \) differ only when any edge \( \vec{e} \) is exposed multiple times and the presence of \( \vec{e} \) in the two decoding graphs with permanently and transiently missing connections differs. Hence, \( \Pr[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma] = \Pr[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma | A_N] \cdot \Pr[A_N] + \Pr[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma | A_N^c] \cdot \Pr[A_N^c]. \)

Since \( \Pr[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma | A_N^c] = 0 \), we can show that \( \Pr[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma \leq 1 \cdot \Pr[A_N] \leq \frac{\tau}{n} \). As \( n \to \infty \), this probability \( \frac{\tau}{n} = \delta \) approaches 0.  

Because of this theorem, in the sequel, no difference is made between the analysis for transiently and permanently missing connection cases.

E. Decoder Useful Region and Thresholds

Usually density evolution converges to a certain stable fixed point with increasing number of iterations \( \ell \). We define this fixed point as:
\[
\bar{P}_e^{(\infty)} = \lim_{\ell \to \infty} \bar{P}_e^{(\ell)} = \lim_{\ell \to \infty} \lim_{n \to \infty} E[\bar{P}_e^{(\ell)}(g, \varepsilon, \alpha)].
\]

In order to decide when to use a decoder with missing connections, a useful decoder is defined. A decoder is said to be useful and should be used instead of taking the codeword directly from the channel without decoding, if the asymptotic decoding error probability satisfies [11]:
\[
\bar{P}_e^{(\infty)} < \bar{P}_e^{(0)} = \varepsilon.
\]

A useful region of a decoder is defined as the set of parameters, which in our case are \( (\varepsilon, \alpha) \), that satisfies the above condition.

When decoding with a fault-free decoder where \( \alpha = 0 \), there exists an \( \varepsilon^* \) below which the final decoding error probability goes to 0 and a much larger value otherwise. We will see in the following sections that \( \bar{P}_e^{(\infty)} \) does not go to zero for positive \( \alpha \), but a threshold phenomenon still exists. For every fixed \( \alpha \), there exists a channel noise decoding threshold \( \varepsilon^* \), below which the final error probability \( \bar{P}_e^{(\infty)} \) goes to a small value \( \eta \). We call decoders that can achieve \( \bar{P}_e^{(\infty)} \) that is lower than this small value \( \eta \)-reliable [11], and the channel noise level that separates the region where decoder is \( \eta \)-reliable or not the decoding threshold \( \varepsilon^* \):
\[
\varepsilon^*(\eta, \alpha) = \sup\{\varepsilon \in [0, 0.5]| \bar{P}_e^{(\infty)} \text{ exists and } \bar{P}_e^{(\infty)} < \eta\}.
\]
IV. PEELING DECODER WITH MISSING CONNECTIONS OVER BINARY ERASURE CHANNEL

Consider the peeling decoder for communication over a BEC with alphabet \{±1, ?\}. The check node computation is a product of all messages ±1 it receives from neighboring variable nodes if none is “?”, otherwise an erasure symbol “?” is sent. The variable node computation is to send any ±1 symbol received either from the other check nodes or from the channel, otherwise send “?”. When the connection between two nodes is missing, the message exchanged is equivalent to “?”, so peeling extends naturally to decoders with missing connections. Note that this decoder satisfies the symmetry conditions C1–C4, so we can use density evolution assuming the all-one codeword was transmitted.

We see that even in the non-asymptotic regime, performance degrades with missing connections.

Lemma 1: Let \(P_e\) be the shorthand of \(P_e(\ell)(g, \varepsilon, \alpha)\), the error probability at the end of \(\ell\)th iteration of decoding a randomly drawn code realization \(g\). For any finite LDPC code \(g\) and finite number of decoding iterations \(\ell\), for both permanently and transiently missing connections, \(P_e\) increases monotonically with \(\alpha\) for a given \(\varepsilon\).

Proof: The proof for monotonicity of \(P_e\) follows by simple coupling arguments. For a specific LDPC code, consider two different missing connection probabilities \(\alpha_1\) and \(\alpha_2\), where \(\alpha_1 < \alpha_2\). Then, we couple the two missing connection processes as follows. Remove the wires with probability \(\alpha_1\), and from this check-variable connection graph, remove each of the remaining connections with probability \(\alpha_2 - \alpha_1\). This gives a second missing connection process. It is not hard to check that the probability of missing connection in the second process is \(\alpha_2\). Thus we can couple the missing connection processes to get a sample path dominance of connections. In this coupling, any realization of \(\alpha_2\) process has more missing connections than that of \(\alpha_1\).

Now consider the probability of correctly decoding any bit \(i\). Note that with the peeling decoder, no erroneous messages are exchanged between the check and variable nodes; only correct messages and erasures are passed along wires. A variable node \(v_i\) holding message bit \(i\) can be decoded correctly if either the received bit is correct, or the received bit is an erasure but \(v_i\) receives a correct message through a path on the computation tree passing through one of its check nodes. The probability that the received bit is correct is the same in case of both \(\alpha_1\) and \(\alpha_2\). So, let us compare the other probability. Now, by coupling as any realization of \(\alpha_1\) has more connections than \(\alpha_2\), if a correct message reaches \(i\) following a path in the \(\alpha_2\) graph, then that path also exists in the \(\alpha_1\) graph. Thus, the event of receiving a correct message in case of \(\alpha_2\) is a subset of that of \(\alpha_1\). This proves the monotonicity of probability of correct decoding. Hence, missing connections can only degrade the performance.

A similar coupling argument yields an ordering relationship with respect to channel erasure probability \(\varepsilon\) for a given \(\alpha\).

A. Density Evolution Equation

First, recall that the peeling decoding algorithm allows \{±1, ?\} to be sent, where “?” stands for an erasure caused by either the channel noise or a missing connection. In this case, the decoder only outputs either the correct message or an erasure symbol.

Consider a regular \((d_v, d_c)\) LDPC code, BEC channel with parameter \(\varepsilon\), and each wire that can be disconnected independently with probability \(\alpha\). Let \(x_0, x_1, ..., x_\ell\) denote the fraction of erasures existing in the code at each decoding iteration. The original received message from the channel is erased with probability \(\varepsilon\), so

\[P_e(0)(\varepsilon, \alpha) = x_0 = \varepsilon.\]

Let \(q_{in}\) be the probability that a node receives an erasure, and \(q_{out}\) be the probability that a node sends out an erasure. At a variable node, the probability of a given internal incident variable will be erased is the probability that both the external incident variable is erased and all other \(d_v - 1\) nodes are either disconnected or connected but erased.

\[q_{out} = x_0 \sum_{i=0}^{d_v-1} \binom{d_v-1}{i} \alpha^i[q_{in}(1-\alpha)](d_v-1-i)\]

At a check node, the probability of a given incident variable will not be erased is the probability that all \(d_c - 1\) other internal incident variables are not erased or disconnected. So the probability that a message is erased is

\[q_{out} = 1 - [(1-q_{in})(1-\alpha)]^{d_c-1}.\]
Hence, the density evolution of the fraction of erasure between two consecutive decoding iterations is
\[ x_{\ell+1} = \varepsilon \left( \alpha + (1 - \alpha)(1 - \rho[(1 - x_{\ell})(1 - \alpha)]) \right) \cdot (d_{-1})^{-1}. \]

The density evolution result can be extended to irregular LDPC codes:
\[ x_{\ell+1} = \varepsilon \lambda \left( \alpha + (1 - \alpha)(1 - \rho[(1 - x_{\ell})(1 - \alpha)]) \right). \]

Let \( f_{DE}(x_\ell, \varepsilon, \alpha) = x_{\ell+1} \) be the recursive update function for the fraction of erasure, where \( 0 \leq \varepsilon < 0.5 \) and \( 0 \leq \alpha \leq 1 \) is the domain of interest.

### B. Fixed Points

The density evolution function \( f_{DE} \) is non-decreasing in each of its arguments, given the other two. Thus, a monotonicity result similar to [24, Lem. 3.54] holds here for this density evolution function. This also implies that a convergence result for \( x_\ell \), similar to [24, Lem. 3.56] holds. So, for a given \( \alpha \) and \( \varepsilon \), \( x_\ell \) converges to the nearest fixed point of \( x = f_{DE}(\varepsilon, x, \alpha) \). Due to this existence of the fixed point, we can characterize the error probability when the decoding process is finished. The fixed points can be found by solving for the real solutions to the polynomial equation
\[ x - \varepsilon \lambda \left( \alpha + (1 - \alpha)(1 - \rho[(1 - x)(1 - \alpha)]) \right) = 0. \]

**Lemma 2:** For any irregular code ensemble \( C^\infty(\lambda, \rho) \), there exists a \( \delta > 0 \), such that the probability of error \( P_e(\infty) \) satisfies \( P_e(\infty) - \delta > \varepsilon \lambda (1 - (1 - \alpha)\rho (1 - \alpha)) > 0 \) with probability 1.

**Proof:** Since \( x_\ell \) is monotonic, if \( x_0 \leq x_1 \) then for any \( \ell \), \( x_{\ell+1} \geq x_\ell \geq x_{\ell-1} \). Now, for \( x_0 = 0 \), by substituting this value in \( f_{DE} \),
\[ x_1 = f_{DE}(0, \varepsilon, \alpha) = \varepsilon \lambda (1 - (1 - \alpha)\rho (1 - \alpha)) > 0 = x_0. \]

This implies that \( \lim_{\ell \rightarrow \infty} x_\ell \geq f_{DE}(0, \varepsilon, \alpha) \), for \( x_0 = 0 \). But, as \( x_\ell \) converges to the fixed point nearest to \( x_0 \), this implies there is no fixed point in \( (0, f_{DE}(0, \varepsilon, \alpha)) \) for any \( \alpha > 0 \).

### C. Performance Analysis

In the previous section, we developed the recursive function to characterize the final error probability achieved by a peeling decoder with missing wires. Now we want to characterize the performance of such decoders.

For a peeling decoder, when \( \varepsilon = 0 \), the error probability stays at 0 regardless of the quality of the decoder. When \( \alpha = 0 \), it has been shown that there exists decoding threshold on the channel noise \( \varepsilon \), below which the final error probability can be driven to 0 with the increase of decoding iterations [23]. For the following analysis, we consider the system when \( \varepsilon > 0 \) and \( \alpha > 0 \). Ideally, we want the error probability to be driven to 0, but as demonstrated in Lem. 2, this is impossible. Here we use the weaker notion of \( \eta \)-reliability defined in Sec. III-E where \( \eta \) limits the final decoding error probability \( P_e \).

Fig. 2 shows the final symbol error rate of decoding a \( C^\infty(3,6) \) LDPC code under peeling decoding with various missing connection probabilities \( \alpha \) over BEC(\( \varepsilon \)). It can be seen that given \( \alpha \), there exists a threshold in channel noise level where a phase transition in \( \eta \) happens. Fig. 3 illustrates such thresholds with the change of \( \alpha \) under different small \( \eta \)-reliable constraints.

Due to the design of the peeling decoder and the non-existence of fixed points from Lem. 2, it is always better to use the decoder even when there is missing connection than just taking the corrupted messages from the channel directly, since the peeling decoder never produces a wrong message.

An interesting phenomenon to notice in the decoding threshold is that there also exists a phase transition with the change of the decoder missing connection probability \( \alpha \). With the increase of \( \alpha \), for a fixed \( \eta \)-reliable decoder with missing connections, the decoding threshold first decreases linearly, and then experience more rapid decrease before convergence to zero.
V. GALLAGER A DECODER WITH MISSING CONNECTIONS OVER BSC

Consider a fault-free Gallager A decoder for communication over a BSC. The messages are passed along the edges in the corresponding Tanner graph during decoding. A check node computes the product of incoming variable-to-check node messages \( \{u_{v\rightarrow c}\} \); a variable node decides to flip the message from channel \( y_v \) if all of the incoming check-to-variable node messages are \(-y_v\) [1].

With the introduction of missing connections, the check node computation is not defined if an input is unknown ("?"). The product computed at the check node is the modulo-2 sum of all incoming messages to ensure that the parity constraints of the code are satisfied. When one of the bits involved in the parity is unknown, that parity check is no longer informative. This is because, any bit of a linear code is equally likely to be \( \pm 1 \) (as complementing a binary codeword gives a codeword). So, for decoders with missing connections we make a natural adaptation: \( u_{c\rightarrow v} = "?" \) if any of the incoming messages- is \"?\". We also make a natural adaptation for variable node computation: \(-y_v\) is sent if more than one non-erasure check node messages are \(-y_v\), and \(y_v\) is sent otherwise.

When it comes to Gallager A decoding over BSC, the messages being passed between nodes may carry erroneous information, unlike the peeling decoder for BEC, where the messages are either correct or erasure. So, for a sample path realization of channel and missing connections, a missing connection may prevent propagation of erroneous messages. Hence, unlike the peeling decoder for BEC, it is not apparent that there exists a stochastic dominance result like Lem.1 between two different probabilities of missing connections.
A. Restriction to All-One Codeword

Fault-free decoding using Gallager A algorithm satisfies conditions C1–C4. So, independence of probability of error from transmitted codeword follows by invoking Prop. 1. So, next, we study density evolution to understand probability of error performance in the large blocklength regime.

B. Density Evolution Equation

We find the probability for a variable node to compute \(-1\) at iteration \(\ell + 1\), in terms of \(x_\ell\). We consider a regular \((d_v, d_c)\) LDPC code and the adaptation of Gallager A decoding with erasure symbols for missing connections.

First note that since a BSC only outputs \(\pm 1\), a variable node never computes “?” with the Gallager A adaptation, even though it may receive (due to connection failure or check-node computes “?”) or send the erasure symbol “?” (only due to connection failure).

The probability that a check node computation is \(-1\) is:

\[
\Pr\{\text{all } (d_c - 1) \text{ variable nodes are connected and send odd number of } -1\} = (1 - \alpha)^{d_c - 1} \Pr\{\text{odd number of } (d_c - 1) \text{ nodes send } -1\}
\]

where the last line follows using results from [1, Sec. 4.3].

The probability that a check node computation is \(+1\) is:

\[
\Pr\{\text{all } (d_c - 1) \text{ variable nodes are connected and send even number of } -1\} = (1 - \alpha)^{d_c - 1} \Pr\{\text{even number of } (d_c - 1) \text{ nodes have } -1\}
\]

The probability that a check-to-variable message is “?” is the complement of the probability that a check node computes \(\pm 1\). Define \(p_0\) to be

\[
1 - (1 - \alpha)^{d_c - 1}.
\]  
(1)

Consider a random variable \(V \sim \text{Binomial}(d_v - 1, 1 - \alpha)\) with probability mass function \(p_V(v)\), capturing the distribution of number of check nodes connected to a variable node. Define \(p_+\) and \(p_-\) such that

\[
p_+ = (1 - \alpha)^{d_c - 1} \frac{1 + (1 - 2x_\ell)^{d_c - 1}}{2}
\]  
(2)

and

\[
p_- = (1 - \alpha)^{d_c - 1} \frac{1 - (1 - 2x_\ell)^{d_c - 1}}{2}.
\]  
(3)

Now consider \(x_{\ell+1}\), the error probability at a variable node at the \((\ell + 1)\)th iteration. The fraction of incorrect values held at this variable node is the sum of the probability of two events. The first event is that the message received from the channel is correct, and none of the incoming messages from the connected check nodes is correct, but not all of them are “?” and not only one says different while others are “?” . The second event is that the message received from the channel is wrong, and at least two of the incoming messages from the connected check nodes are wrong or all of them are “?”.

The probability of the first event is:

\[
\mathbb{E}_V \left[ (1 - \varepsilon) \left[ \Pr\{\text{no connected check nodes sends } 1\} - \Pr\{\text{all } V \text{ connected check nodes send } "?"\} - \Pr\{\text{one check node sends } -1 \text{ while others send } "?"\} \right] \right]
\]

\[
= \sum_{v=1}^{d_v-1} p_V(v)(1 - \varepsilon)\left[(p_- + p_0)^v - p_0^v - p_- p_0^{v-1}\right].
\]
The probability of the second event is:

\[ \mathbb{E}_V \left[ \varepsilon \left[ \Pr\{ \text{at least one connected check nodes sends } -1 \} \right. \right. \\
\left. + \Pr\{ \text{all } V \text{ connected check nodes send } "?" \} \right] \]

\[ = \mathbb{E}_V \left[ \varepsilon \left[ 1 - \Pr\{ \text{no connected check nodes sends } -1 \} \right. \right. \\
\left. + \Pr\{ \text{all } V \text{ connected check nodes send } "?" \} \right] \]

\[ = \sum_{v=0}^{d_v-1} p_V(v)\varepsilon[1 - (p_{+1} + p_0)^v + p_0^v]. \]

Let \( x_{\ell+1} = f_{DE}(x_\ell, \varepsilon, \alpha) \), and take the expectation of \( V \) according to the binomial distribution to get

\[ x_{\ell+1} = f_{DE}(x_\ell, \varepsilon, \alpha) \]

\[ = \varepsilon \alpha^{d_v-1} + \sum_{v=1}^{d_v-1} \binom{d_v-1}{v} (1-\alpha)^v \alpha^{d_v-1-v} \left[(1-\varepsilon)(p_{+1} + p_0)^v - p_0^v - p_{-1}p_0^{v-1} \right] \\
+ \varepsilon[1 - (p_{+1} + p_0)^v + p_0^v]. \]

To extend to irregular LDPC ensembles, we take the average of the check node distribution and get:

\[ p_{+1}^{(irr)} = \rho(1-\alpha) \frac{1-\rho(1-2x_\ell)}{2} \tag{4} \]

and

\[ p_{-1}^{(irr)} = \rho(1-\alpha) \frac{1+\rho(1-2x_\ell)}{2}. \tag{5} \]

The terms in \( f_{DE}(x_\ell, \varepsilon, \alpha) \) have to be averaged over the variable node degree distribution of \( d_v \) with function \( \lambda(\cdot) \).

C. Fixed Points

It can be seen that \( f_{DE}(x, \varepsilon, \alpha) \) is monotonic in \( x \) for a set of given \( \alpha \) and \( \varepsilon \). Hence, by the same arguments as in the case of density evolution for peeling decoders, for any initial \( 0 \leq \varepsilon = x_0 \leq 0.5 \), \( x_\ell \) converges to the nearest fixed point of the density evolution equation. We use \( \tau_1, \tau_2, \tau_3, \ldots \) with ascending order to denote these fixed points.

Note that for all \( \varepsilon > 0, \alpha > 0 \), and \( x_\ell = 0 \), \( f_{DE}(x_\ell, \varepsilon, \alpha) = x_{\ell+1} > 0 \). This implies a result similar to Lem.2 here. With the existence of channel noise and missing wiring, the decoding probability cannot be driven to 0. It is easy to show that for \( \varepsilon = 0 \), \( f_{DE}(x, \varepsilon, \alpha) \) has one fixed point at \( \tau_1 = 0 \). We then focus on the case where \( 0 < \varepsilon < 0.5 \) for the following analysis.

Define \( p^+(x) = (p_{-1} + p_0)^v - p_0^v - p_{-1}p_0^{v-1} \) and \( p^-(x) = 1 - (p_{+1} + p_0)^v + p_0^v \). An analytical expression for the channel threshold is the root \( \tau_2 \) of the following expression between 0 and 0.5:

\[ x\lambda(\alpha) + \lambda_v \left( p^+(x) - xp^+(x) + xp^-(x) \right) = x. \]

The solid line in Fig.4 shows the useful region of decoding for a \((3,6)\) regular LDPC code with missing wire, which is between \( \tau_1 \) and \( \tau_2 \) due to the monotonicity of function \( f_{DE} \). Compared to [11] Figure 2] where computation at each node is noisy with probability \( \alpha \), the useful region of a decoder with missing connection is larger. In this case, decoders with missing connections outperform those with noisy internal computation. At any node, if the corresponding incoming message is missing rather than noisy with probability \( \alpha \), the node is more likely to send out a correct message instead of an erroneous one.
Decoding a $C^{\infty}(3, 6)$ regular LDPC code with $\alpha$-missing wire Gallager A decoding algorithm over BSC($\varepsilon$). The useful region where it is beneficial to use decoder is between the curve and $\alpha$-axis.

Fig. 5. $\eta$-thresholds for decoding a $C^{\infty}(3, 6)$ regular LDPC code with $\alpha$-missing wire Gallager A decoding algorithm over BSC($\varepsilon$).

**D. Performance Analysis**

Fig. 5 shows $\eta$-thresholds for communication over BSC($\varepsilon$) with a Gallager A decoder with missing connections. Recall that for a $(3, 6)$ regular LDPC code with a fault-free Gallager A decoder, the threshold is roughly 0.039 [26]. Note that $P_e$ can be driven to a fairly small number even with the presence of missing wires. Decoding is robust to missing connection defects, though less than the peeling decoder over BEC.

As observed in Fig. 5, a phase transition of the decoding threshold $\varepsilon$ with the change of missing connection probability $\alpha$ noticed in the peeling decoder case also exists here.

Different from classic settings, degree-one nodes might exist in decoding graphs due to the random missing connections. Hence, a tie-breaker at a variable node is necessary when the only incoming message from a check node is different from the received message from the channel. Since the channel message is more reliable than internal messages when there are missing connections in the decoder, we choose not to flip the channel message when the only incoming non-erasure message is the opposite. With this minor twist, the decoding threshold increases significantly. The dotted line in Fig. 4 shows the useful region of the decoder without the tie-breaker for degree-one case and chooses to flip the channel message when all incoming non-erasure messages are different from channel message.

**E. Sensitivity Analysis**

It is also useful to analyze the sensitivity of the stable fixed point with respect to $\varepsilon$ and $\alpha$ in order to come up with new code design and decoding algorithms to improve the performance of decoders with missing connections.

By taking the derivatives of the density evolution function $x_{l+1} = f_{DE}(x_l, \varepsilon, \alpha)$ with respect to $\varepsilon$ and $\alpha$ and evaluating at $\hat{P}_e^{(\infty)} = x_\ell = x_{\ell+1}$, we can describe the impact of noise level of the channel and the missing
Fig. 6. Comparison between the derivative of $\bar{P}_e(\infty)(\varepsilon, \alpha)$ with respect to $\varepsilon$ and $\alpha$ of decoding a $C^\infty(3,6)$ regular LDPC code with $\alpha$-missing wire Gallager A decoding algorithm over BSC($\varepsilon$), when $\varepsilon$ and $\alpha$ are at the boundary of decoder useful region.

connection on the final error rate.

$$\bar{P}_e(\infty)(\varepsilon, \alpha) = \varepsilon \lambda(\alpha) + \lambda_v(1 - \varepsilon)p^+(\bar{P}_e(\infty)(\varepsilon, \alpha)) + \varepsilon p^-(\bar{P}_e(\infty)(\varepsilon, \alpha))$$

Denote

$$g(x) = (1 - \varepsilon)p^+(x(\varepsilon, \alpha)) + \varepsilon p^-(x(\varepsilon, \alpha)).$$

Take partial derivatives of each side with respect to $\varepsilon$:

$$\frac{\partial g(x(\varepsilon, \alpha))}{\partial \varepsilon} = \lambda(\alpha) + \lambda_v \frac{\partial g(x(\varepsilon, \alpha))}{\partial \varepsilon}$$

$$= \lambda(\alpha) + \frac{\partial \lambda_v(\varepsilon \lambda(\alpha))}{\partial \varepsilon}$$

Similarly,

$$\frac{\partial g(x(\varepsilon, \alpha))}{\partial \varepsilon} = \varepsilon \frac{\partial \lambda(\alpha)}{\partial \alpha} + \lambda_v \frac{\partial g(x(\varepsilon, \alpha))}{\partial \alpha}$$

$$+ \lambda_v \frac{\partial g(x(\varepsilon, \alpha))}{\partial \alpha}$$

$$= \varepsilon \frac{\partial \lambda(\alpha)}{\partial \alpha} + \lambda_v \frac{\partial g(x(\varepsilon, \alpha))}{\partial \alpha}$$

$$+ \lambda_v \frac{\partial g(x(\varepsilon, \alpha))}{\partial \alpha}.$$

Fig. 6 illustrates the ratio of the derivative of $\bar{P}_e(\infty)(\varepsilon, \alpha)$ with respect to $\varepsilon$ and $\alpha$, when $\alpha$ and $\varepsilon$ are at the boundary of useful region depicted in Fig. 5.

$$\left(\frac{\partial \bar{P}_e(\infty)(\varepsilon, \alpha)}{\partial \varepsilon} / \frac{\partial \bar{P}_e(\infty)(\varepsilon, \alpha)}{\partial \alpha}\right).$$

Different from our intuition, both derivate values are negative at the boundary of the useful region. Recall the linear relationship of $\varepsilon$ and $\alpha$ at the boundary of the useful region, with the increase of $\alpha$, $\varepsilon$ has to decrease in order to stay in the useful region, resulting in the decrease in $\bar{P}_e(\infty)(\varepsilon, \alpha)$.

For LDPC codes with variable node degree $d_v > 3$, it is possible to consider degree optimization and adding weight factors to the messages that are received from the channel and the messages passed between check and variable nodes; for example, higher weight can be given to the channel messages when $\alpha$ is large and $\varepsilon$ is small in the useful region so that the correct value held at a variable node is less likely to be flipped when there are missing incoming messages in order to lower the error probability.
VI. GALLAGER B DECODER WITH MISSING CONNECTIONS OVER BSC

The Gallager B decoders are usually more robust than the Gallager A decoders without any missing connection [15], so we modify the Gallager B algorithm by introducing erasure symbols for missing connections. In the Gallager B decoder, a check node performs the same operation with incoming variable-to-check node messages as in Gallager A in Sec.[V] sending an unknown symbol “?” if one of the incoming messages is from a disconnected node. At a variable node however, instead of flipping the current value $u$, a variable node in the Gallager B decoder decides to correct the current value $u$ when there are more than $b$ number of incoming messages that are $-u$. This threshold can be iteration-specific to reach optimality. In our case, we use the majority criterion and choose $b^*$ in all iterations because this threshold results in small error probability independent of iteration number in fault-free Gallager B work [27 Sec. 5]. We also choose $b^*$ based on the designed code without counting the number of actually connected nodes for simplicity, and it is verified numerically that there is no significant difference in performance.

Similar to the Gallager A model developed in Sec.[V] the codeword symmetry conditions C1–C4 are all satisfied by invoking Prop.[I].

A. Density Evolution Equation

The density evolution equation for the Gallager B decoder is similar to Gallager A. Consider a regular $(d_v, d_c)$ LDPC code and all-one codeword transmitted over BSC. At iteration $\ell$, the probability of a check-to-variable message is “?” is $p_0$:

$$p_0 = 1 - (1 - \alpha)^{d_v - 1}.\$$

Similarly,

$$p_{+1} = (1 - \alpha)^{d_v - 1} \frac{(1 + (1 - 2x_r)(d_v - 1))}{2}$$

and

$$p_{-1} = (1 - \alpha)^{d_v - 1} \frac{(1 - (1 - 2x_r)(d_v - 1))}{2}.\$$

Now consider $x_{\ell+1}$, the error probability at a variable node at the $(\ell + 1)$th iteration. The fraction of incorrect values held at this variable node is the sum of the probability of two events. The first event is that the message received from the channel is correct, and at least $b = \lfloor \frac{d_v - 1}{2} \rfloor$ check nodes are connected and send incorrect messages. The second event is that the message received from the channel is wrong, and at most $b - 1 = \lceil \frac{d_v - 1}{2} \rceil$ of the incoming messages from the check nodes are correct. Consider a random variable $V \sim \text{Binomial}(d_v - 1, 1 - \alpha)$ capturing the distribution of the number of check nodes connected to a variable node.

The probability of the first event is:

$$\mathbb{E}_V \left[ (1 - \varepsilon) \Pr \{ \text{at least } b \text{ check nodes are connected and send } -1 \} \right]$$

$$= \sum_{v=b}^{d_v-1} p_V(v)(1-\varepsilon)p_{-1}^v(1-p_{-1})^{d_v-1-v}.\$$

The probability of the second event is:

$$\mathbb{E}_V \left[ \varepsilon \Pr \{ \text{at most } (b - 1) \text{ check nodes send } +1 \} \right]$$

$$= \mathbb{E}_V \left[ \varepsilon [1 - \Pr \{ \text{at least } b \text{ check nodes are connected and send } +1 \}] \right]$$

$$= \sum_{v=b}^{d_v-1} p_V(v)\varepsilon[1 - p_{+1}^v(1-p_{+1})^{d_v-1-v}].\$$

Taking the expectation of $V$ according to the binomial distribution, we have

$$x_{\ell+1} = \sum_{v=b}^{d_v-1} \binom{d_v-1}{v} (1-\alpha)^{v}(d_v-1-v)^{\alpha(d_v-1-v)} \left[ (1-\varepsilon)[p_{-1}^v(1-p_{-1})^{d_v-1-v}] + \varepsilon[1 - p_{+1}^v(1-p_{+1})^{d_v-1-v}] \right].\$$

The density evolution equation can also be extended to irregular LDPC codes, with changes in parameters $b(x) = \lfloor \frac{x+1}{2} \rfloor$, $p_{+1}^{(irr)}$, and $p_{-1}^{(irr)}$ defined in expressions (44) and (5).
B. Performance Analysis

We carry out detailed performance characterization of Gallager B decoder with missing connection and show that such a decoder is indeed more robust to missing wiring compared to Gallager A.

Note that when variable node degree $d_v \leq 3$ for a regular LDPC code, Gallager B decoder with the defined threshold $b = \left\lfloor \frac{d_v+1}{2} \right\rfloor$ is equivalent to Gallager A. Hence, the example is for a $C^{\infty}(4, 8)$ regular LDPC code.

One interesting phenomenon here is that the decoding threshold first increases with the increasing decoder missing connection probability. This error enhancement phenomenon is introduced by the missing connections, essentially resulting in a change of choice for threshold $b$ in each iteration to achieve a lower error rate. This stochastic facilitation (SF) phenomenon demonstrates that optimization of degree distribution and threshold $b$ in each iteration can be utilized to combat the missing connections in noisy decoders. A similar SF result shows that the errors introduced in estimating Markov random field models can be partially canceled and benefit the end-to-end performance [28]. SF effects due to noise in computational elements, rather than graphical model structure misspecification as here, have been observed [21], [22], [29].

VII. Conclusion

This paper investigated the performance of message-passing decoders with transiently and permanently missing connections that might be caused by process variation in manufacturing or timing errors in intra-chip communications (or both). We derived density evolution equations to characterize the error probability in the peeling decoders over the BEC and the Gallager A and Gallager B decoders over the BSC, using erasure symbols to represent missing connections. Although the error probability cannot be driven to 0 in the presence of missing connections, it can be suppressed to a small value $\eta$ when the channel noise level is under a certain decoding threshold $\varepsilon^*$. That is, $\eta$-reliable communication is possible with faulty decoders with missing connections. In a sense, even when the encoder and decoder speak different languages, the result is not catastrophic. A novel structural stochastic facilitation is also observed in Gallager B decoders with missing connections.

Future work involves considering not just decoders with missing connections, but also miswired and noisy decoders. One may also design new decoder architectures to ensure reliable communication even with miswiring; for example, horizontal connections, a crucial structure in the cortex contributing to the filling in of the missing parts in visual images [30, Ch. 8.33], can be added to decoder designs. Code optimization and new decoding algorithms can also be utilized to take advantage of the stochastic facilitation effect.

ACKNOWLEDGMENT

The authors thank A. Patil and N. R. Shanbhag for discussions on nanoscale circuits.

APPENDIX A

PROBABILITY THEORY DEFINITIONS

Before diving into the proof of Thm. 1, some probability theory definitions and the Hoeffding-Azuma inequality are reviewed here. Consider a space $(\Omega, \mathcal{F})$, where $\Omega$ is a sample space, and a $\sigma$-algebra $\mathcal{F}$ contains subsets of
Ω. A random variable \( Z \) is an \( \mathcal{F} \)-measurable function from a probability space into the real number. If there is a collection \((Z_\gamma|\gamma \in C)\) of random variables \(Z_\gamma: \Omega \to \mathbb{R}\), then

\[
Z = \sigma(Z_\gamma|\gamma \in C)
\]

is defined to be the smallest \( \sigma \)-algebra \( Z \) on \( \Omega \) such that each map \((Z_\gamma|\gamma \in C)\) is \( Z \)-measurable.

**Definition 1 (Filtration):** Let \( \{\mathcal{F}_i\} \) be a sequence of \( \sigma \)-algebras with respect to the same sample space \( \Omega \). These \( \mathcal{F}_i \) are said to form a filtration if \( \mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \) are ordered by refinement in the sense that each subset of \( \Omega \) in \( \mathcal{F}_i \) is also in \( \mathcal{F}_j \) for \( i \leq j \). Also \( \mathcal{F}_0 = \{\emptyset, \Omega\} \).

The conditional expectation of a random variable \( Z \) given a \( \sigma \)-algebra \( \mathcal{F} \) is a random variable denoted by \( E[Z|\mathcal{F}] \).

**Definition 2 (Martingale):** Let \( \mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \) be a filtration on \( \Omega \) and let \( Z_0, Z_1, \ldots \) be a sequence of random variables on \( \Omega \) such that \( Z_i \) is \( \mathcal{F}_i \)-measurable. Then \( Z_0, Z_1, \ldots \) is a Martingale with respect to the filtration \( \mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \) if \( E[Z_i|\mathcal{F}_{i-1}] = Z_{i-1} \).

**Definition 3 (Doob’s Martingale):** Let \( \mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \) be a filtration on \( \Omega \) and let \( Z \) be a random variable on \( \Omega \). Then the sequence of random variables \( Z_0, Z_1, \ldots \) such that \( Z_i = E[Z|\mathcal{F}_i] \) is a Doob’s Martingale.

**Lemma 3 (Hoeffding-Azuma Inequality [23], [31], [32]):** Let \( Z_0, Z_1, \ldots \) be a Martingale with respect to the filtration \( \mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots \) such that for each \( i > 0 \), the following bounded difference condition is satisfied

\[
|Z_i - Z_{i-1}| \leq \alpha_i, \alpha_i \in [0, \infty).
\]

Then for all \( n > 0 \) and any \( \xi > 0 \),

\[
\Pr\left[\left|Z_n - Z_0\right| \geq \xi\right] \leq 2 \exp\left(-\frac{\xi^2}{2 \sum_{k=1}^{n} \alpha_k^2}\right).
\]

### Appendix B

**CONCENTRATION: PERMANENTLY MISSING CONNECTIONS**

The proof of Thm. [1] is an extension from and largely identical to [11] Thm. 2, [23] Thm. 2, or [24] Thm. 4.94. We want to construct a Doob’s Martingale with respect to the fraction of error held on each edge during the random revealing process and to show that the difference of the object of interest between each iteration is bounded by a number not related to the number of iterations.

Recall that \( Z \) denotes the number of incorrect values held at the end of the \( \ell \)th iteration for a specific \((g, y, w) \in \Omega\), where \( g \) is a specific bipartite Tanner graph to represent the choice of LDPC code with variable node degree \( d_v \) and check node degree \( d_c \), \( y \) is a specific input to the decoder, \( w \) is a particular realization of the decoder with missing wires, and \( \Omega \) is the sample space. Let \( \equiv_i, 0 \leq i \leq m \) be a sequence of equivalence relations on \( \Omega \) ordered by refinement, such that \((g', y', w') \equiv_i (g'', y'', w'') \) implies \((g', y', w') \equiv_{i-1} (g'', y'', w'') \).

Next we use the technique of exposing the edges in the decoding graph in sequence. The first case is when wires are permanently missing. Consider a random variable \( V \sim \text{Binomial}(d_v - 1, 1 - \alpha) \) capturing the distribution of number of check nodes connected to a variable node. Instead of \( nd_v \) edges in the fault-free decoder case, we expose \( nV \) edges one at a time, where \( \alpha \) is the probability of one specific edge missing in the graph. In all cases, the number of edges exposed is bounded by \( nd_v \). At step \( i \leq nd_v \), we expose the particular check node socket that is connected to the \( i \)th variable node socket. Next, in the following \( n \) steps, we expose the received values \( y_i \) from the channel one at a time. At the end of the \( n(d_v + 1) \) steps, the decoder missing wire probability is also realized, since the defect is permanent. Then we have \((g', y', w') \equiv_i (g'', y'', w'') \) if and only if the information revealed in the first \( i \) steps for both pairs is the same.

Now, define \( Z_0, Z_1, \ldots, Z_m \) by

\[
Z_i(g, y, w) = E[Z(g', y', w')|(g', y', w') \equiv_i (g, y, w)],
\]

where \( Z_0 = E[Z] \) and \( Z_m = Z \). By construction, \( Z_0, Z_1, \ldots, Z_m \) is a Doob’s Martingale. We then use Lem. 3 to give bounds on

\[
\Pr[|Z - E[Z]| > nd_v \varepsilon/2] = \Pr[|Z_m - Z_0| > nd_v \varepsilon/2].
\]

To use Azuma’s inequality, we first need to prove that for each consecutive member in the sequence \( Z_0, Z_1, \ldots, Z_m \), the difference is bounded:

\[
|Z_{i+1}(g, y, w) - Z_i(g, y, w)| \leq \delta_i, i = 0, 1, \ldots, m - 1
\]
where $\delta_i$ depends on $d_v, d_c$, and $\ell$.

It was shown by Richardson and Urbanke [23] that for the fault-free decoder without any missing wire, when edges are exposed,

$$|Z_{i+1}(g, y, w) - Z_i(g, y, w)| \leq 8(d_v d_c)^\ell, 0 \leq i \leq nd_v.$$ 

In our case when there exist permanently missing connections, the difference when exposing edges is that the number of edges existing is smaller, and bounded by $n d_v$. The expected number of edges left is $n d_v (1 - \alpha)$. The bound established above still holds with a change of the steps number:

$$|Z_{i+1}(g, y, w) - Z_i(g, y, w)| \leq 8(d_v d_c)^\ell, 0 \leq i \leq nd_v.$$

It was also shown that when channel outputs are revealed, the difference in each element in the sequence is bounded by

$$|Z_{i+1}(g, y, w) - Z_i(g, y, w)| \leq 2(d_v d_c)^\ell,$$

where $n (d_v + 1) \leq i \leq n d_v$ in the case where some wires are permanently missing. Then the theorem follows from applying Azuma’s inequality to the Martingale constructed.

**APPENDIX C**

**CONCENTRATION: TRANSIENTLY MISSING CONNECTIONS**

The second case is when wires are transiently missing at each decoding iteration. The Martingale is constructed differently. Instead of exposing edges, at $\ell$ iterations, we sequentially expose the realization of edges at different iterations. Since each edge can be missing independently from others with probability $\alpha$, only sockets whose nodes are connected through these edges are affected. In each iteration, there are 2 realizations for each edge (present or missing), then for all previous $\ell$ iterations, the total number affected edges is bounded by $2(2d_v d_c)\ell$. With symmetry of switching node sockets:

$$|Z_{i+1}(g, y, w) - Z_i(g, y, w)| \leq 8(2d_v d_c)^\ell$$

where $n (d_v + 1) \leq i \leq m$.

Hence, in the transiently missing wire case, the bounded difference $\delta_i = 8(2d_v d_c)^\ell$. The theorem follows from applying Azuma’s inequality to the Martingale constructed.

**REFERENCES**

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[2] A. A. Al-Yamani, S. Ramsundar, and D. K. Pradhan, “A defect tolerance scheme for nanotechnology circuits,” *IEEE Trans. Circuits Syst. I*, vol. 54, no. 11, pp. 2402–2409, Nov. 2007.

[3] S.-L. Jeng, J.-C. Lu, and K. Wang, “A review of reliability research on nanotechnology,” *IEEE Trans. Rel.*, vol. 56, no. 3, pp. 401–410, Sep. 2007.

[4] Y. M. Chee and A. C. H. Ling, “Limit on the addressability of fault-tolerant nanowire decoders,” *IEEE Trans. Comput.*, vol. 58, no. 1, pp. 60–68, Jan. 2009.

[5] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, “A defect-tolerant computer architecture: Opportunities for nanotechnology,” *Science*, vol. 280, no. 5370, pp. 1716–1721, Jun. 1998.

[6] M. Haselman and S. Hauck, “The future of integrated circuits: A survey of nanoelectronics,” *Proc. IEEE*, vol. 98, no. 1, pp. 11–38, Jan. 2010.

[7] M. A. Breuer, S. K. Gupta, and T. M. Mak, “Defect and error tolerance in the presence of massive numbers of defects,” *IEEE Des. Test. Comput.*, vol. 21, no. 3, pp. 216–227, May–June 2004.

[8] S. Ghosh and K. Roy, “Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era,” *Proc. IEEE*, vol. 98, no. 10, pp. 1718–1751, Oct. 2010.

[9] P. Elias, “Computation in the presence of noise,” *IBM J. Res. Develop.*, vol. 2, no. 4, pp. 346–353, Oct. 1958.

[10] S. Winograd and J. D. Cowan, *Reliable Computation in the Presence of Noise*. Cambridge, MA: MIT Press, 1963.

[11] L. R. Varshney, “Performance of LDPC codes under faulty iterative decoding,” *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4427–4444, Jul. 2011.

[12] S. M. S. Tabatabaei Yazdi, H. Cho, and L. Dolecek, “Gallager B decoder on noisy hardware,” *IEEE Trans. Commun.*, vol. 61, no. 5, pp. 1660–1673, May 2013.

[13] S. M. S. Tabatabaei Yazdi, C.-H. Huang, and L. Dolecek, “Optimal design of a Gallager B noisy decoder for irregular LDPC codes,” *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2052–2055, Dec. 2012.

[14] F. Leduc-Primeau and W. J. Gross, “Faulty Gallager-B decoding with optimal message repetition,” in *Proc. 50th Annu. Allerton Conf. Commun. Control Comput.*, Oct. 2012, pp. 549–556.
[15] C.-H. Huang, Y. Li, and L. Dolecek, “Gallager B LDPC decoder with transient and permanent errors,” *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 15–28, Jan. 2014.

[16] C. K. Ngassa, V. Savin, E. Dupraz, and D. Declercq, “Density evolution and functional threshold for the noisy min-sum decoder,” *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1497–1509, May 2015.

[17] B. Vasić and S. K. Chilappagari, “An information theoretical framework for analysis and design of nanoscale fault-tolerant memories based on low-density parity-check codes,” *IEEE Trans. Circuits Syst. I*, vol. 54, no. 11, pp. 2438–2446, Nov. 2007.

[18] E. Dupraz, D. Declercq, B. Vasić, and V. Savin, “Analysis and design of finite alphabet iterative decoders robust to faulty hardware,” *IEEE Trans. Commun.*, vol. 63, no. 8, pp. 2797–2809, Aug. 2015.

[19] O. A. Rasheed, P. Ivaniš, and B. Vasić, “Fault-tolerant probabilistic gradient-descent bit flipping decoder,” *IEEE Commun. Lett.*, vol. 18, no. 9, pp. 1487–1490, Sep. 2014.

[20] C.-H. Huang, Y. Li, and L. Dolecek, “Belief propagation algorithms on noisy hardware,” *IEEE Trans. Commun.*, vol. 63, no. 1, pp. 11–24, Jan. 2015.

[21] A. Karbasi, A. H. Salavati, A. Shokrollahi, and L. R. Varshney, “Noise facilitation in associative memories of exponential capacity,” *Neural Comput.*, vol. 26, no. 11, pp. 2493–2526, Nov. 2014.

[22] H. Chen, L. R. Varshney, and P. K. Varshney, “Noise-enhanced information systems,” *Proc. IEEE*, vol. 102, no. 10, pp. 1607–1621, Oct. 2014.

[23] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[24] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge: Cambridge University Press, 2008.

[25] H. Mori and T. Wadayama, “Performance analysis based on density evolution on fault erasure belief propagation decoder,” arXiv:1601.07239v1 [cs.IT], Jan. 2016.

[26] L. Bazzi, T. J. Richardson, and R. L. Urbanke, “Exact thresholds and optimal codes for the binary-symmetric channel and Gallager’s decoding algorithm A,” *IEEE Trans. Inf. Theory*, vol. 50, no. 9, pp. 2010–2021, Sep. 2004.

[27] V. Guruswami, “Iterative decoding of low-density parity check codes (a survey),” arXiv:cs/0610022 [cs.IT], 2006.

[28] M. J. Wainwright, “Estimating the “wrong” graphical model: Benefits in the computation-limited setting,” *J. Mach. Learn. Res.*, vol. 7, pp. 1829–1859, Dec. 2006.

[29] T. J. Hamilton, S. Afshar, A. van Schaik, and J. Tapson, “Stochastic electronics: A neuro-inspired design paradigm for integrated circuits,” *Proc. IEEE*, vol. 102, no. 5, pp. 843–859, May 2014.

[30] P. Brodal, *The Central Nervous System*. New York, NY: Oxford University Press, 2016.

[31] K. Azuma, “Weighted sums of certain dependent random variables,” *Tohoku Math. J.*, vol. 19, no. 3, pp. 357–367, 1967.

[32] W. Hoeffding, “Probability inequalities for sums of bounded random variables,” *J. Am. Stat. Assoc.*, vol. 58, no. 301, pp. 13–30, Mar. 1963.