Information-theoretic tradeoffs of throughput and chip power consumption for decoding error-correcting codes

Pulkit Grover, Hari Palaiyanur and Anant Sahai
Wireless Foundations, Department of EECS
University of California at Berkeley, CA-94720, USA
{pulkit, hpalaiya, sahai} @eecs.berkeley.edu

Abstract

The purpose of this paper is to develop an information-theoretic understanding of the tradeoffs between decoder power, probability of error and decoding throughput. We start by considering the power consumed in the interconnects of the decoder circuit, modeled as a lumped capacitor and resistor. After making simplifying assumptions on the decoder circuit, we use a sphere-packing technique to lower bound the decoding error probability for a given number of clock-cycles (or iterations). The analysis can be used to give lower bounds on probability of error versus total decoding power at a fixed decoding throughput, for example.

I. Introduction

Reducing power consumption is immensely important for wireless devices. Classically, information theory has mostly focused on the transmit power. However, modern indoor applications consume more power in signal processing at the transmitter and the receiver (see, for example, [1]) than in transmit power. In particular, the power required for *decoding* the error correcting code is itself significant, and in some cases, larger than the transmit power (see [2], [3]).

A full understanding of the problem requires a system level perspective where the design of the transmission scheme takes into account the architecture of the decoder as well. With the exception of [3]–[5] (and the references therein), this perspective has been largely ignored. In [4], an implementation-centric approach is taken. The authors design sparse-graph codes with structural regularity so that reduced complexity decoding architectures can be designed. In our work (see [3], [5]), we derive information-theoretic lower bounds on the decoding energy required with mild assumptions on the decoding architecture — the decoding circuit is synchronous, has bounded connectivity, and each computational node consumes a fixed amount of energy per clock-cycle. No assumption is made on the code structure.

A crucial piece of the puzzle that is missing thus far is the relation of power with the speed of decoding (or *decoding throughput*). Transmitting the data at high rates is useful only if the decoding throughput is equally high. Assuming a fully parallel decoding (as is the case with the modern day decoding architectures), high decoding throughput requires running the circuit at higher voltages so that parasitic capacitances can be charged and discharged quickly. This results in higher power consumption as well [2]. A model for how power consumption scales with throughput is therefore required. Not only is it hard to model power consumption in computation; due to improvements in the design and implementation of transistors and gates, the fraction of total energy consumed by the computational nodes has decreased considerably. Comparable, or even larger, power is consumed in the *interconnects*, or the wires connecting various processors and circuit elements (see, for example [6] and the references therein).

In this paper, by concentrating on the interconnect power consumption, we set up a framework for obtaining tradeoffs of decoding power with decoding throughput. Each interconnect is fairly accurately modeled by an Elmore lumped model (see, for example, [7, Ch. 4]) — a resistance connected to ground via a parasitic capacitance. The relation between power consumption and speed of charging/dischraging of this capacitance in Elmore model can be computed easily, thus allowing for our investigation. We ignore the power consumed in computational nodes assuming that its behavior is similar to that of the interconnects (after all, computational nodes also have similar parasitic capacitances and resistances). A more detailed circuit-theoretic analysis is needed to verify this assumption.

Such an investigation naturally involves a combination of circuit-theoretic and information-theoretic techniques. This paper presents bounds on the tradeoff between throughput and decoding power under some assumptions on the decoding architecture. Two tools are developed in order to arrive at these bounds. The first tool is a noisy computation lemma, that might be of more general interest. The second is a sphere-packing bound that allows us to lower bound the error probability of a fictitious decoder which is related to the actual decoder. The link between the two tools is based on an intuitive heuristic, with the hope that it can be made rigorous under limited assumptions on the code.

The organization of this paper is as follows. In Section II, we give the problem statement. In Section III, we introduce the model of the decoder and the model for power consumption in the interconnects. We also introduce tools that are useful in bounding the power consumption at the decoder. Section V provides the lower bound on error probability for given decoding throughput, number of clock-cycles (i.e. iterations), and the gap of the code rate from the channel capacity. These bounds are then used to plot lower bounds on error probability as a function of decoding power for a fixed throughput. Because of space limitations, most of the proofs are available in the extended version of this paper [8].

II. Problem statement

A block code encodes k iid equiprobable message bits $\mathbf{B}^k$ into a binary codeword $\mathbf{X}^m$ of blocklength m, resulting in a channel code rate of $R_{ch} = \frac{k}{m}$ bits/channel use. The transmission channel is a Binary Symmetric Channel (BSC) with crossover probability p_{ch}, the capacity of which is denoted $C_{BSC(p_{ch})} = 1 - h_b(p_{ch}) > R_{ch}$, where $h_b(\cdot)$ is the binary entropy in bits. The channel outputs $\mathbf{Y}^m = \mathbf{X}^m \oplus \mathbf{E}^m$, where $\mathbf{E}^m$ is a binary vector of channel noise. The decoder computes estimates $\widehat{\mathbf{B}}^k$ of the message bits using a message-passing decoding algorithm described in Section III. We will assume that the decoder uses a finite amount of power, and hence noise can also be added in the decoding process.

The objective is to minimize the average bit-error probability $\langle P_e \rangle$ which is given by

$$\langle P_e \rangle = \frac{1}{k} \sum_{i=1}^{k} \langle P_{e,i} \rangle, \tag{1}$$

where $\langle P_{e,i} \rangle$ is the error-probability of i-th bit given by $\langle P_{e,i} \rangle = \mathbb{P}\mathrm{r}(B_i \neq \widehat{B}_i)$. This error probability is averaged over the bits $\mathbf{B}^k$, the channel noise realizations $\mathbf{E}^m$, and noise in the decoder circuit.

III. Decoder model

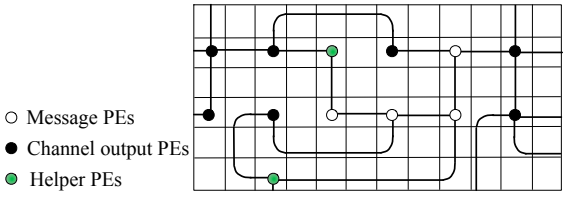

Fig. 1. A VLSI decoding architecture. The message PEs are responsible for computing and storing values of the decoded message bits. The channel output PEs store the values of the channel outputs. The helper PEs help decode by improving connectivity between the nodes.

The VLSI model of a decoder (see Figure 1) consists of registers and processors. The registers store either the received channel symbol or the decoded bits (after the decoding process is completed). Each register has an accompanying processor that can perform read/write operations on the register and other computational tasks. These register-processor pairs are referred to as Processing Elements (PEs) in VLSI complexity theory literature [9]. The PEs are connected to each other using wires, or interconnects. Implementation constraints dictate that each PE can be connected to α other PEs, a constraint that arises from limitations on the interconnect-density. The interconnects are assumed to be bi-directional. During one iteration, each PE passes one-bit messages to all the other PEs it is connected to. The message can depend on the messages received thus far from the other PEs, and the value stored in the PE's register. The circuit operates for L iterations before making its final estimates $\widehat{\mathbf{B}}^k$. The structure of the decoding circuit imposes limitations on the information that can be passed between the circuit elements. It is this limitation that we exploit to obtain a tradeoff between decoding energy and probability of error.

The total time for decoding is given by $T_{dec} = L(T + T_{node})$, where T_{node} is the time required for computation at a node and T is the time to transmit a bit on an interconnect. The throughput is, therefore,

$$R_{dec} = \frac{k}{L(T + T_{node})}. \tag{2}$$

In order to lower bound the energy for a given throughput, we assume that $T_{node} = 0$ and thus $R_{dec} = \frac{k}{LT}$.

A. Interconnect model and power consumption

In a departure from our earlier models [3], [5], in this paper we ignore the power consumed in the PEs themselves. Instead, we focus on the power consumed in the interconnects. The electrical model of interconnects that we consider is the Elmore-lumped elements model, shown in Figure 2. We assume that the transmission across the interconnect is binary using a voltage of $\pm V_0$ volts (corresponding to logical '1' and '0') for a time duration of T seconds. The decoder samples the output every T seconds. If there were no noise, the voltage at the capacitor at the end of T seconds would have been V_1. The output voltage has noise because of the inherent thermal noise and crosstalk from neighboring interconnects. The output signal is therefore, $V_{out} = V_1 + Z_{dec}$, where Z_{dec} is modeled as an additive white Gaussian noise distributed $\mathcal{N}(0, \sigma_z^2)$. Each PE then makes a thresholding hard-decision of V_{out}. Each interconnect is thus treated as an independent BSC with crossover probability $p_{dec} = \mathbb{P}\mathrm{r}(Z_{dec} > V_1) = Q\left(\frac{V_1}{\sigma_z^2}\right)$, where $Q(x) = \frac{1}{2\pi}\int_x^\infty e^{-\frac{s^2}{2}} ds$ is the tail probability of the standard Gaussian distribution.

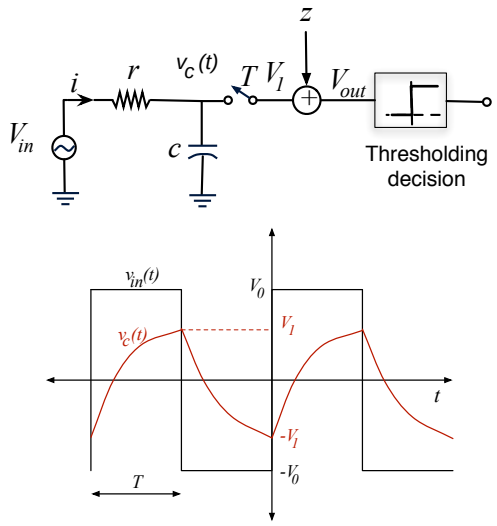

Fig. 2. The Elmore lumped model for an interconnect. Also shown is a sampler and additive noise that model the noisy processing at the receiver. The response to a square-wave input to the interconnect.

For simplicity, we assume that the power consumed in the wire is the same[1] as that consumed when the driving voltage is a sequence of alternating logical '0's and '1's. In Appendix I, we show that for this model, the average power consumed over one iteration in one interconnect is given by

$$\overline{P}_{wire} = \frac{E_{wire}}{T} = \frac{2cV_1V_0}{T}. \tag{3}$$

The total power consumed in the decoding circuit is therefore lower bounded by

$$\overline{P}_{tot} \geq \frac{\alpha}{2} m \overline{P}_{wire} = \frac{\alpha c V_1 V_0 m}{T}, \tag{4}$$

where the factor $\frac{\alpha}{2}$ comes from the observation that each edge is shared between two node (effectively giving each node a 'half-edge'), and each node has degree α.

B. The decoding neighborhood

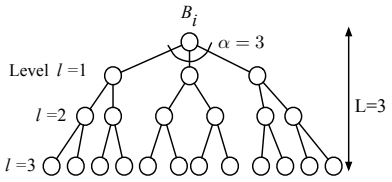

Fig. 3. The decoding neighborhood of a bit B_i. The nodes represent the processing elements.

The decoding neighborhood of a bit B_i after L iterations is the set of nodes at graphical distance L or smaller from the message node B_i. We assume that the decoding neighborhood of each bit is a tree (see Figure 3). Nodes at level l are those that are at a graphical distance of l from the root bit node. The channel outputs stored in PEs that lie in the decoding neighborhood of B_i are denoted by $\mathbf{Y}_{(i)}^n$, and are by the cycle-free assumption $n = \sum_{l=1}^{L} \alpha(\alpha-1)^{l-1}$. Similarly, the codeword bits in the neighborhood are denoted by $\mathbf{X}_{(i)}^n$, and the channel errors in the neighborhood are denoted by $\mathbf{E}_{(i)}^n$. Thus, $\mathbf{Y}_{(i)}^n = \mathbf{X}_{(i)}^n \oplus \mathbf{E}_{(i)}^n$. The number of nodes at level l is denoted by $n[l] = \alpha(\alpha-1)^{l-1}$.

[1]In practice, the decoding tends to stabilize after a few clock-cycles, reducing the number of switching messages in each clock cycle. However, in many practical implementations of LDPC-like codes (see, for example, [2]), to save chip-area, the same PE may act as different computational node in alternating iterations. Even as the decoding stabilizes, the switching activity in the wires does not because of this alternating behavior.

IV. Converting the noisy decoder to a concatenated channel

A. A fictitious decoder, and a concatenated channel

We define a 'fictitious decoder' $\widetilde{\mathcal{D}}_i$ for each bit i. This 'fictitious decoder' will then be converted to another 'fictitious decoder', $\overline{\mathcal{D}}_i$, that will ultimately be used to lower bound the probability of error for our true decoder.

First, for $1 \leq i \leq k$ and $1 \leq j \leq m$, let $\ell(i,j)$ denote the level of bit i's decoding neighborhood where channel output Y_j appears. If Y_j never appears in bit i's decoding neighborhood, let $\ell(i,j) = \infty$. The first fictitious decoder for bit i, $\widetilde{\mathcal{D}}_i$, receives $\widetilde{Y}_{i,j}(t) = Y_j \oplus \widetilde{E}_{i,j}(t)$ for $1 \leq t \leq L - \ell(i,j) + 1$, where $\widetilde{E}_{i,j}(t)$ are iid $Ber(p_{dec})$ noises independent of all other noises and channel outputs. That is, we give the decoder access to $L - \ell(i,j) + 1$ copies of channel outputs Y_j corrupted by the noise on an interconnect. Let

$$\widetilde{\mathbf{Y}}^n_{(i)} = \left(\widetilde{Y}_{i,j}(t) : \ell(i,j) \leq L, 1 \leq t \leq L - \ell + 1\right)$$

denote the observations $\widetilde{\mathcal{D}}_i$ has access to. Then the output of the fictitious decoder is $\widetilde{B}_i = \widetilde{\mathcal{D}}_i(\widetilde{\mathbf{Y}}^n_{(i)})$. Assuming this probability model for the information received by $\widetilde{\mathcal{D}}_i$, let $\langle \widetilde{P}_{e,i} \rangle = \mathbb{P}\text{r}(\widetilde{B}_i \neq B_i)$ and

$$\langle \widetilde{P}_e \rangle = \frac{1}{k} \sum_{i=1}^{k} \langle \widetilde{P}_{e,i} \rangle.$$

A 'concatenated channel' $CC(p_{ch}, p_{dec})$ is a fictitious memoryless channel on the decoding neighborhood of a bit. The channel is seen from the perspective of the root node responsible for decoding B_i. From this perspective, the channel inputs for PEs at level l are first corrupted by the channel noise Ber(p_{ch}) and are further corrupted by a decoder noise Ber($p_d^{(l)}$) which depends on the level l. The overall flip probability, $p_{CC,l}$, is therefore $p_{CC,l} = p_{ch}(1 - p_d^{(l)}) + p_d^{(l)}(1 - p_{ch})$. The decoder flip probability $p_d^{(l)}$ is based on the majority-logic decision based on $L - l + 1$ noisy observations of a random variable Y_i corrupted by noise Ber(p_{dec}). That is,

$$p_d^{(l)} = \begin{cases} \sum_{m=\lceil \frac{L-l+1}{2} \rceil}^{L-l+1} \binom{L-1+1}{m} p_{dec}^m (1-p_{dec})^{L-l+1-m} \\ \qquad \text{if } L-l+1 \text{ is odd} \\ 0.5 \times \binom{L-1+1}{\frac{L-l+1}{2}} p_{dec}^{\frac{L-l+1}{2}} (1-p_{dec})^{\frac{L-l+1}{2}} \\ + \sum_{m=\frac{L-l+3}{2}}^{L-l+1} \binom{L-1+1}{m} p_{dec}^m (1-p_{dec})^{L-l+1-m} \\ \qquad \text{if } L-l+1 \text{ is even} \end{cases} \tag{5}$$

The second fictitious decoder receives n messages; one message for each channel output in its decoding neighborhood[2]. Let $\overline{Y}_{i,j} = Y_j \oplus \overline{E}_{i,j}$ for j such that $\ell(i,j) \leq L$, where $\overline{E}_{i,j}$ are independent Bernoulli random variables with distribution $Ber(p_d^{(\ell(i,j))})$, where $p_d^{(l)}$ is defined above. Letting

$$\overline{\mathbf{Y}}^n_{(i)} = \left(\overline{Y}_{i,j} : \ell(i,j) \leq L\right)$$

denote the observations received by this decoder, the decoded bit is $\overline{B}_i = \overline{\mathcal{D}}_i(\overline{\mathbf{Y}}^n_{(i)})$. We think of the second fictitious decoder as being like the first, except that it performs a majority-logic decision to determine the value of the channel outputs in its neighborhood. It then decides the value of the required bit based on the values of those $\overline{Y}_j$ that have $\ell(i,j) \leq L$. Assuming this probability model for the information received by $\overline{\mathcal{D}}_i$, let $\langle P_{e,i} \rangle_{CC} = \mathbb{P}\text{r}(\overline{B}_i \neq B_i)$ and

$$\langle P_e \rangle_{CC} = \frac{1}{k} \sum_{i=1}^{k} \langle P_{e,i} \rangle_{CC}.$$

Note that this is called a concatenated channel because from $\overline{\mathcal{D}}_i$'s perspective, $\overline{Y}_{i,j} = Y_j \oplus \overline{E}_{i,j} = X_j \oplus E_{CC,i,j}$, where $E_{CC,i,j}$ are $Ber(p_{CC,\ell(i,j)})$ random variables, independent over j for a fixed i. Let $\mathbf{E}^n_{CC,(i)} := (E_{CC,i,j} : \ell(i,j) \leq l)$.

B. Noisy computation

In order to relate the $\langle P_e \rangle$ in our noisy circuit model to $\langle \widetilde{P}_e \rangle$, we first prove a lemma about noisy computations.

Lemma 1 (A noisy computation lemma): Given a noisy circuit that takes as inputs A, a binary random variable, and $\mathbf{V}^m$, a possibly dependent binary random vector, and outputs a binary function $Y = f(A, \mathbf{V}^m) \oplus Z_A$, where $Z_A \sim$Ber(p_A) is a binary noise independent of $(A, \mathbf{V}^m)$, the circuit operation can be simulated by a circuit that has $\mathbf{V}^m$ and $A \oplus Z_A$ as its inputs such that the two circuits are statistically identical, i.e. $\mathbb{P}\text{r}(Y = y | A, \mathbf{V}^m) = \mathbb{P}\text{r}(Y' = y | A, \mathbf{V}^m)$. See Figure 4 for a diagram.

[2]The rate-limitation on the messages is not well accounted for by this fictitious decoder. Taking this rate-limitation into account can often tighten the bounds, as shown in [10].

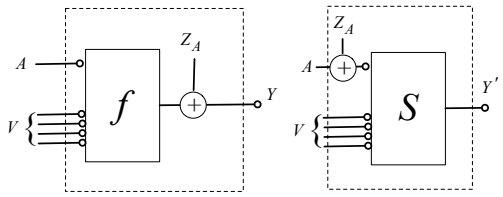

Fig. 4. The circuit S on the right can (statistically) simulate the circuit on the left if Z_A is independent of (A, V).

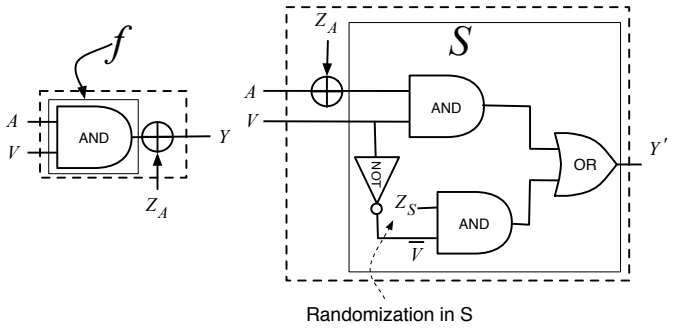

Fig. 5. Simulation of a circuit that implements f with noise Z_A at output by a circuit S that has noise Z_A at input A. The randomization Z_S is needed at S.

Proof: For a fixed $\mathbf{V}^m$, $f(A, \mathbf{V}^m)$ is a binary function of A. Since A is binary itself, this function can be either $f(A, \mathbf{V}^m) = 0$, $f(A, \mathbf{V}^m) = 1$, $f(A, \mathbf{V}^m) = A$, or $f(A, \mathbf{V}^m) = \overline{A}$. In the first two cases, the simulating circuit can merely generate an independent Ber(p_A) noise and add it (modulo two) to 0 or 1, as the case may be. In the third case, $f(A, \mathbf{V}^m) \oplus Z_A = A \oplus Z_A$, which is available at the simulating circuit. In the fourth case, $f(A, \mathbf{V}^m) \oplus Z_A = \overline{A} \oplus Z_A = (A \oplus Z_A) \oplus 1$ which can again be obtained from $A \oplus Z_A$.

That is, the noisy function $f(A, \mathbf{V}^m) \oplus Z_A$ can be (statistically) simulated by a circuit S that has noiseless outputs and has $A \oplus Z_A$ and $\mathbf{V}^m$ as the inputs. ■

Figure 4 illustrates the lemma. It is important to note that the simulator circuit S can be different from the actual circuit f. Simple calculations on an AND gate show that the lemma does not hold if S is forced to be equal to f. S needs an extra randomization to simulate the f circuit with noise at the output, as shown in Figure 5.

Now, we use the noisy computation lemma to convert our actual decoder to a form for which it is easier to give lower bounds on the error probability. At this time, we focus on the node responsible for decoding a given bit, say B_i. We make the following assumption about the messages being passed in the decoding neighborhood.

Assumption 1: For any two nodes A, B, the message passed from node A to node B is a function of all messages received by node A *except those sent by node B itself.*

This assumption is based on the present-day (belief-propagation-based) message-passing decoders, which have this property in absence of cycles.[3] We also assume that all nodes in the tree apart from the root (decoding) node hold the value of a distinct channel output.[4]

The root node receives a total of $L\alpha$ messages. Of these messages, only $L - l + 1$ can contain information about any one channel output located at level l. We let $\widehat{B}_i$ denote the output of the root node after L iterations. Formally, we think of the root node as outputting an arbitrary noise-free binary function of $L\alpha$ binary inputs. Let us call the messages received by the decoding node in this tree as $M^{L\alpha}$ and the decoding function itself $\mathcal{D}_i$, so $\widehat{B}_i = \mathcal{D}_i(M^{L\alpha})$.

For any decoder with structure as above, we will show that there exists a fictitious decoder (see Section IV-A) with the same joint distribution between the channel outputs and the output of the decoder. Because the bit and decoded bit are independent given the channel outputs in the neighborhood, a lower bound to the probability of error for the fictitious decoder is also a lower bound to our actual decoder.

Lemma 2: For each i, there exists $\widetilde{\mathcal{D}}_i$ with access to $\widetilde{\mathbf{Y}}^n_{(i)}$ such that

$$\mathbb{P}\mathrm{r}\left(\mathbf{Y}^n_{(i)}, \mathcal{D}_i(M^{L\alpha})\right) = \mathbb{P}\mathrm{r}\left(\mathbf{Y}^n_{(i)}, \widetilde{\mathcal{D}}_i(\widetilde{\mathbf{Y}}^n_{(i)})\right),$$

and therefore $\langle P_{e,i} \rangle = \langle \widetilde{P}_{e,i} \rangle$.

[3]We note that this assumption disallows schemes based on feedback between nodes, which might be a useful scheme in presence of noise in computation. In this case, it allows us to simplify the analysis without losing applicability to present-day decoders.

[4]If there are nodes in the middle of the tree merely used for computational/storage/connectivity purposes, we can assume that they store a channel output and the decoder chooses to ignore this channel output.

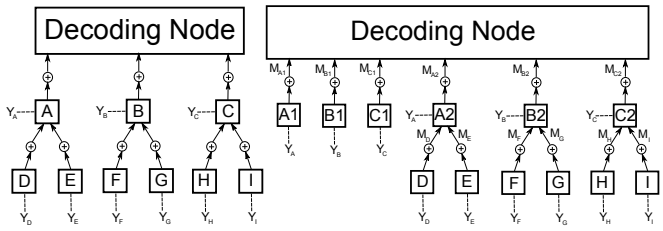

Fig. 6. The decoding tree for one node, $\alpha = 3$, $L = 1$ (left) and $L = 2$ (right). A circle with a plus inside indicates a noisy edge where Bernoulli noise is added as per the interconnect model. A total of 6 messages arrive at the decoding node over two iterations, with messages taking one iteration to move one level up in the tree.

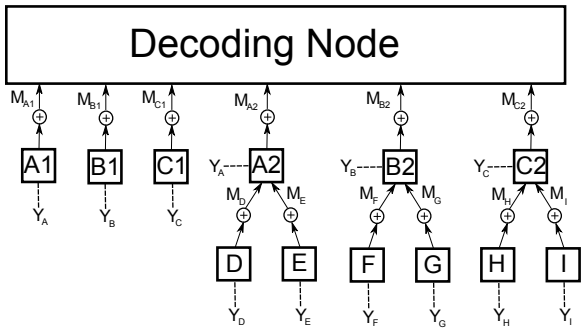

Fig. 7. Since there are two iterations, one can think of an expanded tree in which each message that is received during the L iterations comes from one subtree, for which all computations are performed in one clock cycle.

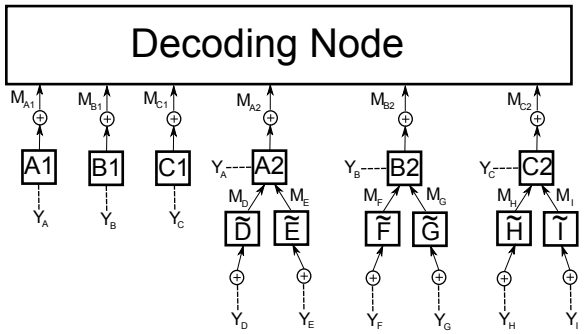

Fig. 8. The noisy computation lemma is first used on the lowest level of the decoding neighborhood.

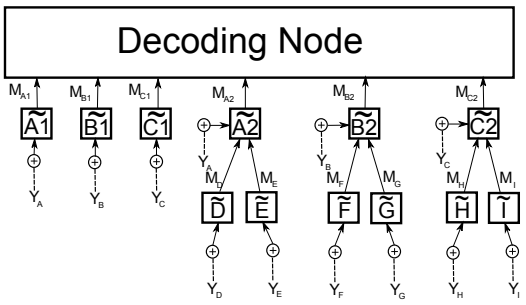

Fig. 9. Using the noisy computation lemma, one can move the noises at the lowest level in the tree to the channel outputs, without changing the joint probability distribution of the intermediate messages with the channel outputs. The noisy computation lemma is then used once again in the next level up.

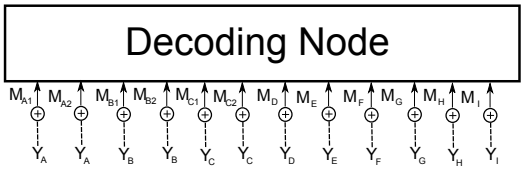

Fig. 10. Since $L = 2$, all the noises in the decoding tree are now being added directly to channel outputs and within the function computations. All the function computations can be moved inside the decoding node.

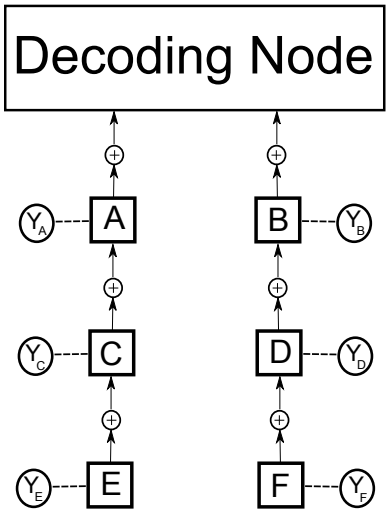

Fig. 11. An example of a decoding neighborhood when $\alpha = 2$ and $L = 3$. With a larger number of iterations, messages received in the past are allowed to be stored in the PEs and used to compute the values of messages passed towards the decoding node in the future.

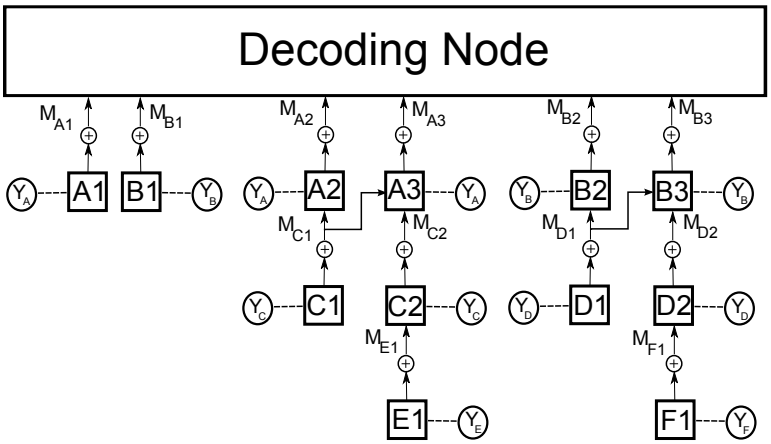

Fig. 12. In this example, since $L = 3$, message M_{C1} is stored at node A and potentially used to compute the value of M_{A3}, along with M_{C2} and Y_A. The same is true of M_{D1} being stored at node B for use in computation of M_{B3} along with M_{D2} and Y_B. Even though these messages can be stored, the graph formed is still acyclic.

Proof: This can be proved by using Lemma 1 iteratively up the tree, starting at the leaves. We view the original decoding node as collecting $L\alpha$ one-bit messages over L iterations and then applying a binary function with these messages as its input. The process is illustrated via an example in Figures 6 to 10.

We first expand the decoding tree in time in Figure 7. That is, the decoding node is now connected to $L\alpha$ subtrees corresponding to the $L\alpha$ messages it can receive. The subtrees for each message are composed of the nodes that can contribute to that particular message. Essentially, this large tree allows us to view all messages passed in the tree during L iterations on one graph, as well as what these messages can depend on. Note that this large tree forms a directed acyclic graph (DAG) because we have assumed that there are no cycles and messages from one node to another are not allowed to depend of messages received in the reverse direction (Assumption 1). The fact that the decoding tree is a DAG after being unfolded in time allows us to apply the noisy computation lemma from the leaves up to the root node and maintain the joint distribution between the output of the root node and the channel outputs.

Any PE in the decoding neighborhood of B_i has a channel output connected to it, as shown in Figure 6. As shown in Figure 8, at the first step, noises at the output of leaf nodes at the lowest level in the tree are moved to the inputs that correspond to channel outputs using Lemma 1. The functions are replaced by respective simulating functions that are denoted by a ‘ $\widetilde{\ }$ ’ at the top. The joint distribution between $\mathbf{Y}_{(i)}^n$ and all messages in Figure 7 has been maintained in this process. The same operation can now be performed by moving up the tree. The resulting tree has all the noises between channel outputs and PEs, but none on the links connecting the PEs, as shown in Fig. 8.

The previous example was fairly simple as L was 2, so there was no need for any nodes to store messages received in the past. In Figure 11, an example with $L = 3$ and $\alpha = 2$ is shown. As L becomes large, the PEs could potentially store past received messages to use for computation of future outgoing messages. Figure 12 shows the decoding neighborhood of

Figure 11 unfurled after 3 iterations. It can be seen that messages M_{C1} and M_{D1} can be stored at nodes A and B for use in computing the values of M_{A3} and M_{B3} respectively. However, we can still apply the noisy computation lemma iteratively up the graph starting at the lowest level. This is because the graph is still a DAG, and storing past messages does not destroy that property. The reason is that all messages are passed up one level in the direction towards the decoding node.

To finish the proof, we go back to Figure 9, after the noisy computation lemma has been applied so that every noise on the interconnects has been moved to noises directly on the channel outputs. At this point, there is no noise between any of the PEs and the decoding node. The decoding node may therefore implement the same functions as the PEs (after the noisy computation lemma has been applied), based on direct observation of the channel outputs with additional noises, as shown in Figure 10. Therefore, there is a $\widetilde{\mathcal{D}}_i$ with access to $\widetilde{\mathbf{Y}}^n_{(i)}$ that has

$$\mathbb{P}\mathrm{r}\left(\mathbf{Y}^n_{(i)}, \mathcal{D}_i(M^{L\alpha})\right) = \mathbb{P}\mathrm{r}\left(\mathbf{Y}^n_{(i)}, \widetilde{\mathcal{D}}_i(\widetilde{\mathbf{Y}}^n_{(i)})\right).$$

Hence, since we have the Markov chain B_i—$\mathbf{Y}^n_{(i)}$—$\widetilde{\mathbf{Y}}^n_{(i)}$, $\langle P_{e,i}\rangle = \langle \widetilde{P}_{e,i}\rangle$. ■

C. Converting to the concatenated channel

Conjecture 1: For each i, and given $\widetilde{\mathcal{D}}_i$, there exists $\overline{\mathcal{D}}_i$ with access to $\overline{\mathbf{Y}}^n_{(i)}$ such that $\langle \widetilde{P}_{e,i}\rangle \geq \langle P_{e,i}\rangle_{CC}$.

It is plausible that this conjecture may be true because the noise added in multiple observations of a channel output is independent of the channel output, hence the majority-logic decision may be a sufficient statistic for the purposes of estimating B_i. However, the majority-logic operation implicitly assumes that the channel output is equiprobably 0 or 1, and that to some degree, Y_j's in the decoding neighborhood are approximately independent. The equiprobability of Y_j's is not true of all codes, but certainly true of linear codes such as LDPCs where all channel input symbols are non-degenerate (i.e. not always taking on the value 0). An approximate independence of Y_j's also holds for capacity-approaching codes.

The effect of this conjecture, if it can be proved, is that a lower bound to $\langle P_e\rangle_{CC}$ on all decoders that have access to $\overline{\mathbf{Y}}^n_{(i)}$ becomes a lower bound to $\langle P_e\rangle$ in this noisy circuit model.

Lemma 3: Assuming the validity of Conjecture 1, for each i there exist $\overline{\mathcal{D}}_i$ with access to $\overline{\mathbf{Y}}^n_{(i)}$ such that

$$\langle P_e\rangle \geq \langle P_e\rangle_{CC}.$$

Proof: Assuming the conjecture and combining with the bound in Lemma 2 gives the desired result. ■

V. Main results

Theorem 1: The average bit-error probability under the concatenated channel is lower bounded by

$$\langle P_e\rangle_{CC} \geq \sup_{g:C_{BSC(g)}<R} f_G\left(h_b^{-1}\left(\delta(g)\right)\right) \tag{6}$$

for any $g \in (0, 0.5)$ satisfying $C_{BSC(g)} < R$, where $\delta(g) = 1 - \frac{C_{BSC(g)}}{R}$, $C_{BSC(g)} = 1 - h_b(g)$,

$$f_G(x) = \frac{x}{2}\prod_{l=1}^{l^*}\left(2^{-n[l]D(g\|p_{CC,l})} \times \left(\frac{p_{CC,l}(1-g)}{g(1-p_{CC,l})}\right)^{\epsilon_{l^*}\left(\frac{x}{2}\right)2^{\frac{l^*-l}{2}}\sqrt{n[l]}}\right),$$

, $D(p\|q)$ is the KL-divergence between Ber(p) and Ber(q) distributions,

$$l^* = \begin{cases} \max\{l : p_{CC,l} < g \ s.t. \ \leq L\} & \text{if } p_{CC,L} > g \\ L & \text{otherwise,} \end{cases}$$

$$\epsilon_{l^*}(x) = \sqrt{\frac{1}{\phi(g)}\log_2\left(\frac{1}{x}+2\right)}, \tag{7}$$

$\phi(g) = \min\limits_{0<\eta<1-g} \frac{D(g+\eta\|g)}{\eta^2} = \frac{1}{1-2g}\log_2\left(\frac{1-g}{g}\right)$, $n[l]$ denotes the number of nodes at the l-th level in the decoding neighborhood of a bit, and $p_{CC,l}$ is as defined in (5).

Proof: The proof is an extension of the sphere-packing bound for neighborhood sizes [3], [5], [11] to a case of unequal errors in nodes on each level.

Recall that $\mathbf{E}^n_{CC,(i)}$ denotes the error vector constituted by $E_{CC,i,j}$, the errors through the concatenated channel in the neighborhood of Bit i. Similarly, let $\mathbf{E}^{n[l]}_{CC,(i),l}$ denote the error vector $[E_{CC,i,j} : \ell(i,j) = l]$, i.e. the vector of errors at level l in the decoding neighborhood of the i-th bit. Let $w(\mathbf{E}^{n[l]}_{CC,(i),l})$ be the weight (i.e. the number of 1's in a binary vector) of $\mathbf{E}^{n[l]}_{CC,(i),l}$.

Define a test channel G as follows

Definition 1 (test channel G): Given a concatenated channel $CC(p_{ch}, p_{dec})$ with independent errors $E_{i,j} \sim Ber(p_{CC,\ell(i,j)})$, the test channel G is defined as a channel with independent errors $E_{i,j} \sim Ber(\max(g, p_{CC,\ell(i,j)}))$, i.e. a channel with crossover probability g on all levels g such that $p_{CC,l} < g$, and crossover probability of $p_{CC,l}$ otherwise.

We first need the following lemma on $\langle P_e \rangle_G$, the average bit error probability under the test channel G.

Lemma 4:

$$\langle P_e \rangle_G \geq h_b^{-1}\left(\delta(g)\right), \tag{8}$$

where $\delta(g) = 1 - \frac{C_{BSC(g)}}{R}$, $C_{BSC(g)} = 1 - h_b(g)$, and $h_b(\cdot)$ is the binary entropy function.

Proof: See Appendix IV. ■

Define the typical set:

$$\mathcal{T}_{\epsilon,G}(i) := \{\mathbf{E}^n_{CC,(i)} : w(\mathbf{E}^{n[l]}_{CC,(i),l}) - gn[l] \leq \epsilon_l \sqrt{n[l]}, l = 1, \ldots, l^*\}$$

for a given vector $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_{l^*})$. The following lemma provides a lower bound on the probability of $\mathcal{T}_{\epsilon,G}(i)$ for a particular choice of the vector ϵ^{l^*}.

Lemma 5: Fix an $x \in (0,1)$. For the choice $\epsilon_l = 2^{\frac{l^*-l}{2}} \epsilon_{l^*}(x)$, $l = 1, \ldots, l^* - 1$, and $\epsilon_{l^*}(x) = \sqrt{\frac{1}{\phi(g)} \log_2\left(\frac{1}{x} + 2\right)}$, $\Pr(\mathcal{T}_{\epsilon,G}(i))$ is lower bounded as follows

$$\Pr(\mathcal{T}_{\epsilon,G}(i)) \geq 1 - x. \tag{9}$$

Proof: See Appendix II. ■

The following lemma lower bounds the error probability of the i-th bit under the concatenated channel given its error probability under the test channel G.

Lemma 6:

$$\langle P_{e,i} \rangle_{CC} \geq f_G(\langle P_{e,i} \rangle_G). \tag{10}$$

Proof: Define $A_i(\mathbf{B}^k) := \{\mathbf{E}^n_{CC,(i)} : \overline{\mathcal{D}}_i(\mathbf{X}^n_{(i)}(\mathbf{B}^k) \oplus \mathbf{E}^n_{CC,(i)}) \neq B_i\}$ as the collection of error sequences that result in a decoding error for any decoding function $\overline{\mathcal{D}}_i$. Then,

$$\langle P_{e,i} \rangle_{CC} = \frac{1}{2^k} \sum_{\mathbf{B}^k} \mathbb{P}\mathrm{r}_{CC}(A_i(\mathbf{B}^k)). \tag{11}$$

Now,

$$\begin{aligned}
\mathbb{P}\mathrm{r}_{CC}(A_i(\mathbf{B}^k)) &= \sum_{\mathbf{E}^n_{CC,(i)}\in A_i(\mathbf{B}^k)} \mathbb{P}\mathrm{r}_{CC}(\mathbf{E}^n_{CC,(i)}) \\
&= \sum_{\mathbf{E}^n_{CC,(i)}\in A_i(\mathbf{B}^k)} \frac{\mathbb{P}\mathrm{r}_{CC}(\mathbf{E}^n_{CC,(i)})}{\mathbb{P}\mathrm{r}_G(\mathbf{E}^n_{CC,(i)})}\mathbb{P}\mathrm{r}_G(\mathbf{E}^n_{CC,(i)}) \\
&= \sum_{\mathbf{E}^n_{CC,(i)}\in A_i(\mathbf{B}^k)} \mathbb{P}\mathrm{r}_G(\mathbf{E}^n_{CC,(i)}) \left(\prod_{l=1}^{l^*} \frac{(p_{CC,l})^{w(\mathbf{E}^{n[l]}_{(i),l})}(1-p_{CC,l})^{n[l]-w(\mathbf{E}^{n[l]}_{(i),l})}}{(g)^{w(\mathbf{E}^{n[l]}_{(i),l})}(1-g)^{n[l]-w(\mathbf{E}^{n[l]}_{(i),l})}}\right) \times \\
&\qquad \left(\prod_{l=l^*+1}^{L} \frac{(p_{CC,l})^{w(\mathbf{E}^{n[l]}_{(i),l})}(1-p_{CC,l})^{n[l]-w(\mathbf{E}^{n[l]}_{(i),l})}}{(p_{CC,l})^{w(\mathbf{E}^{n[l]}_{(i),l})}(1-p_{CC,l})^{n[l]-w(\mathbf{E}^{n[l]}_{(i),l})}}\right) \\
&= \sum_{\mathbf{E}^n_{CC,(i)}\in A_i(\mathbf{B}^k)} \mathbb{P}\mathrm{r}_G(\mathbf{E}^n_{CC,(i)}) \prod_{l=1}^{l^*} \frac{(p_{CC,l})^{w(\mathbf{E}^{n[l]}_{(i),l})}(1-p_{CC,l})^{n[l]-w(\mathbf{E}^{n[l]}_{(i),l})}}{(g)^{w(\mathbf{E}^{n[l]}_{(i),l})}(1-g)^{n[l]-w(\mathbf{E}^{n[l]}_{(i),l})}} \\
&= \sum_{\mathbf{E}^n_{CC,(i)}\in A_i(\mathbf{B}^k)} \mathbb{P}\mathrm{r}_G(\mathbf{E}^n_{CC,(i)}) \prod_{l=1}^{l^*} \left(\frac{p_{CC,l}(1-g)}{(1-p_{CC,l})g}\right)^{w(\mathbf{E}^{n[l]}_{(i),l})} \frac{(1-p_{CC,l})^{n[l]}}{(1-g)^{n[l]}} \\
&\geq \sum_{\mathbf{E}^n_{CC,(i)}\in A_i(\mathbf{B}^k)\cap \mathcal{T}_{\epsilon,G}(i)} \mathbb{P}\mathrm{r}_G(\mathbf{E}^n_{CC,(i)}) \prod_{l=1}^{l^*} \left(\frac{p_{CC,l}(1-g)}{(1-p_{CC,l})g}\right)^{w(\mathbf{E}^{n[l]}_{(i),l})} \frac{(1-p_{CC,l})^{n[l]}}{(1-g)^{n[l]}} \\
&\geq \sum_{\mathbf{E}^n_{CC,(i)}\in A_i(\mathbf{B}^k)\cap \mathcal{T}_{\epsilon,G}(i)} \mathbb{P}\mathrm{r}_G(\mathbf{E}^n_{CC,(i)}) \prod_{l=1}^{l^*} \left(\frac{p_{CC,l}(1-g)}{(1-p_{CC,l})g}\right)^{gn[l]+\epsilon_l\sqrt{n[l]}} \frac{(1-p_{CC,l})^{n[l]}}{(1-g)^{n[l]}} \\
&= \sum_{\mathbf{E}^n_{CC,(i)}\in A_i(\mathbf{B}^k)\cap \mathcal{T}_{\epsilon,G}(i)} \mathbb{P}\mathrm{r}_G(\mathbf{E}^n_{CC,(i)}) \prod_{l=1}^{l^*} \left(\frac{p_{CC,l}(1-g)}{(1-p_{CC,l})g}\right)^{\epsilon_l\sqrt{n[l]}} 2^{-n[l]D(g\|p_{CC,l})} \\
&= \left(\prod_{l=1}^{l^*} \left(\frac{p_{CC,l}(1-g)}{(1-p_{CC,l})g}\right)^{\epsilon_l\sqrt{n[l]}} 2^{-n[l]D(g\|p_{CC,l})}\right) \sum_{\mathbf{E}^n_{CC,(i)}\in A_i(\mathbf{B}^k)\cap \mathcal{T}_{\epsilon,G}(i)} \mathbb{P}\mathrm{r}_G(\mathbf{E}^n_{CC,(i)}) \\
&= \left(\prod_{l=1}^{l^*} \left(\frac{p_{CC,l}(1-g)}{(1-p_{CC,l})g}\right)^{\epsilon_l\sqrt{n[l]}} 2^{-n[l]D(g\|p_{CC,l})}\right) \mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k)\cap \mathcal{T}_{\epsilon,G}(i))
\end{aligned}$$

Now, choosing $\epsilon_l = 2^{\frac{l^*-l}{2}}\epsilon_{l^*}$, $l = 1,\ldots,l^*-1$, and $\epsilon_{l^*} = \sqrt{\frac{2}{\phi(g)}\log_2\left(\frac{1}{\mathbb{P}\mathrm{r}(A_i(\mathbf{B}^k))}+2\right)}$, Lemma 5 guarantees that $\mathbb{P}\mathrm{r}(\mathcal{T}_{\epsilon,G}(i)) \geq 1 - \frac{\mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k))}{2}$. Using the equality $\mathbb{P}\mathrm{r}(S)+\mathbb{P}\mathrm{r}(T) = \mathbb{P}\mathrm{r}(S\cup T)+\mathbb{P}\mathrm{r}(S\cap T)$ for any two measurable sets S and T,

$$\begin{aligned}
\mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k)\cap \mathcal{T}_{\epsilon,G}(i)) &= \mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k)) + \mathbb{P}\mathrm{r}_G(\mathcal{T}_{\epsilon,G}(i)) - \mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k)\cup \mathcal{T}_{\epsilon,G}(i)) \\
&\geq \mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k)) + 1 - \frac{\mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k))}{2} - \mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k)\cup \mathcal{T}_{\epsilon,G}(i)) \\
&\geq \mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k)) + 1 - \frac{\mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k))}{2} - 1 \\
&= \frac{\mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k))}{2}.
\end{aligned}$$

Thus, we obtain

$$\begin{aligned}
\mathbb{P}\mathrm{r}_{CC}(A_i(\mathbf{B}^k)) &\geq \frac{\mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k))}{2} \prod_{l=1}^{l^*} \left(\frac{p_{CC,l}(1-g)}{(1-p_{CC,l})g}\right)^{\epsilon_{l^*}(\mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k)))2^{\frac{l^*-l}{2}}\sqrt{n[l]}} 2^{-n[l]D(g\|p_{CC,l})} \\
&= f_G(\mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k)))
\end{aligned}$$

Using (11),

$$\begin{aligned}\langle P_{e,i}\rangle_{CC} &= \frac{1}{2^k}\sum_{\mathbf{B}^k} f_G(\mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k))) \\ &\overset{(a)}{\geq} f_G\left(\frac{1}{2^k}\sum_{\mathbf{B}^k}\mathbb{P}\mathrm{r}_G(A_i(\mathbf{B}^k))\right) \\ &= f_G(\langle P_{e,i}\rangle_G),\end{aligned}$$

thus proving the lemma. Inequality (a) follows from convexity of $f_G(\cdot)$ which is proved in Appendix III. ■

To complete the proof of the theorem,

$$\begin{aligned}\langle P_e\rangle_{CC} &= \frac{1}{k}\sum_{i=1}^{k}\langle P_{e,i}\rangle_{CC} \\ &\geq \frac{1}{k}\sum_{i=1}^{k} f_G(\langle P_{e,i}\rangle_G) \\ &\overset{(a)}{\geq} f_G\left(\frac{1}{k}\sum_{i=1}^{k}\langle P_{e,i}\rangle_G\right) \\ &= f_G(\langle P_e\rangle_G) \\ &\overset{(b)}{\geq} f_G(\delta(g)),\end{aligned}$$

where (a) follows from the convexity of $f_G(\cdot)$, and (b) follows from Lemma 4 and the monotonicity of $f_G(\cdot)$, which is also proved in Appendix III. ■

A. *Tradeoff of throughput with power consumption*

Using Theorem 1, in Figure 13, we plot lower bounds on error probability for given throughput using the energy consumption model of Section III-A. It can be seen that the larger the throughput for a fixed decoding power, the higher the lower bound on bit error probability.

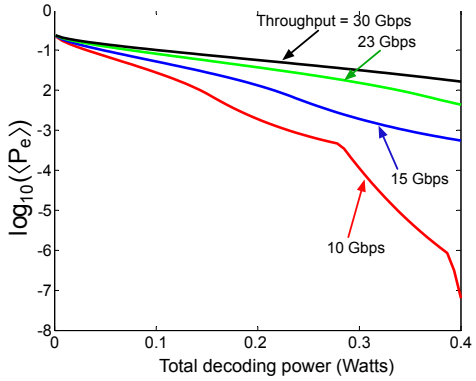

Fig. 13. Tradeoff of the error probability with power consumption for $\sigma_z^2 = 0.005\ V^2$, $R_{ch} = 0.4$, $p_{ch} = 0.11$ (corresponding to a channel capacity of 0.5 bits per channel use), $k = 1000$ bits, $\alpha = 4$, $r = 100\Omega$, and $c = 100$ pF.

ACKNOWLEDGMENTS

Discussions with Animesh Kumar, Bora Nikolic and Matt Wiener are gratefully acknowledged. We also thank the reviewers of this paper for ISIT for their feedback. This research is supported by NSF grants CCF-0729122, CCF-0917212 and CNS-0932410.

REFERENCES

[1] PY Massaad, M Medard, and L Zheng. Impact of Processing Energy on the Capacity of Wireless Channels. In *International Symposium on Information Theory and its Applications (ISITA)*, 2004.

[2] Z Zhang, V Anantharam, MJ Wainwright, and B Nikolic. A 47 Gb/s LDPC decoder with improved low error rate performance. *Symposium on VLSI Circuits*, pages 286–287, June 2009.

[3] Pulkit Grover and Anant Sahai. Green codes: Energy-efficient short-range communication. In *Proceedings of the 2008 IEEE Symposium on Information Theory*, Toronto, Canada, July 2008.

[4] Mohammad M. Mansour and Naresh R. Shanbhag. Low-power vlsi decoder architectures for ldpc codes. In *ISLPED '02: Proceedings of the 2002 international symposium on Low power electronics and design*, pages 284–289, New York, NY, USA, 2002. ACM.

[5] Anant Sahai and Pulkit Grover. The price of certainty : "waterslide curves" and the gap to capacity. December 2007.

[6] D Chen, J Cong, and P Pan. *FPGA Design Automation: A Survey*, volume 1. Foundations and Trends inElectronic Design Automation, NOW Publishing, Hanover, MA, November 2006.

[7] B Lu and SS Sapatnekar D Du. *Layout Optimization In Vlsi Design*. Kluwer Academic Publishers, 2001.

[8] P Grover, H Palaiyanur, and A Sahai. Information-theoretic tradeoffs on throughput and chip power consumption for decoding error-correcting codes. extended version. `http://www.eecs.berkeley.edu/~pulkit/ISIT10Power.pdf`, 2010.

[9] CD Thompson. *A complexity theory for VLSI*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1980.

[10] Wee Peng Tay, John N. Tsitsiklis, and Moe Z Win. Data fusion trees for detection: Does architecture matter. *IEEE Trans. Inform. Theory*, pages 4155–4168, 2007.

[11] Anant Sahai. Why block-length and delay behave differently if feedback is present. *IEEE Trans. Inform. Theory*, 54(5):1860–1886, May 2008.

APPENDIX I
POWER CONSUMED IN AN INTERCONNECT ON SQUARE-WAVE INPUT

Figure 2 shows the circuit model being analyzed, and the notation. Assume that a square-wave input is applied, alternating between $+V_0$ and $-V_0$ every T seconds. The voltage as a function of time across the capacitor is $v_c(t)$. We wish to find the steady state behavior of $v_c(t)$, especially the value $v_c(t)$ takes when $v_{in}(t)$ alternates between $+V_0$ to $-V_0$ or vice-versa, which is denoted V_1. From Kirchoff's laws, we have

$$\begin{aligned}
& V_0 - i(t)r - v_c(t) = 0 \\
& i(t) = c\frac{dv_c(t)}{dt} \\
\Rightarrow \quad & V_0 - v_c(t) = rc\frac{dv_c}{dt} \\
\text{i.e.} \quad & \frac{dv_c}{V_0 - v_c(t)} = \frac{dt}{rc} \\
\text{i.e.} \quad & \int_{-V_1}^{v_c} \frac{dv_c}{V_0 - v_c(t)} = \frac{t}{rc} \\
\Rightarrow \quad & v_c(t) = V_0(1 - e^{-\frac{t}{rc}}) - V_1 e^{-\frac{t}{rc}}.
\end{aligned}$$

The voltage V_1 is, therefore

$$\begin{aligned}
V_1 &= V_0\left(1 - e^{-\frac{T}{rc}}\right) - V_1 e^{-\frac{T}{rc}} \\
\Rightarrow V_1 &= \frac{V_0(1 - e^{-\frac{T}{rc}})}{1 + e^{-\frac{T}{rc}}}
\end{aligned}$$

Now, analyzing the current $i(t)$,

$$\begin{aligned}
i(t) &= c\frac{dv_c(t)}{dt} = \frac{cV_0}{rc}e^{-\frac{t}{rc}} + c\frac{V_1}{rc}e^{-\frac{t}{rc}} \\
&= \frac{V_0 + V_1}{r}e^{-\frac{t}{rc}} = \frac{2V_0}{(1 + e^{-\frac{T}{rc}})r}e^{-\frac{t}{rc}}.
\end{aligned}$$

Thus the energy consumed at the resistance in an interconnect in one clock-cycle is

$$\begin{aligned}
E_{wire}(V_0, T) &= \int_0^T i^2(t) r dt \\
&= \frac{4V_0^2}{r(1 + e^{-\frac{T}{rc}})^2}\int_0^T e^{-2\frac{t}{rc}}dt \\
&= 2cV_0^2\frac{1 - e^{-\frac{2T}{rc}}}{(1 + e^{-\frac{T}{rc}})^2} = 2cV_0^2\frac{1 - e^{-\frac{T}{rc}}}{1 + e^{-\frac{T}{rc}}} = 2cV_1V_0.
\end{aligned}$$

Thus the average power consumed over one clock-cycle at the resistor is given by

$$\overline{P}_{wire} = \frac{E_{wire}}{T} = \frac{2cV_1V_0}{T}. \tag{12}$$

APPENDIX II
PROOF OF LEMMA 5

We need the following lemma.

Lemma 7: For $q_l > 0$, $\sum_l q_l < 1$,

$$\prod_{l=1}^{l^*}(1-q_l) \geq \exp\left(-\frac{\sum_{l=1}^{l^*} q_l}{1-\sum_{l=1}^{l^*} q_l}\right). \tag{13}$$

Proof:

$$\begin{aligned}
\ln\left(\prod_{l=1}^{l^*}(1-q_l)\right) &= \sum_{l=1}^{l^*}\ln(1-q_l) \\
&= \sum_{l=1}^{l^*}\left(-q_l - \frac{q_l^2}{2} - \frac{q_l^3}{3} - \dots\right) \\
&= -\sum_{l=1}^{l^*} q_l - \sum_{l=1}^{l^*}\frac{q_l^2}{2} - \sum_{l=1}^{l^*}\frac{q_l^3}{3} - \dots \\
&\geq -\sum_{l=1}^{l^*} q_l - \frac{\left(\sum_{l=1}^{l^*} q_l\right)^2}{2} - \frac{\left(\sum_{l=1}^{l^*} q_l\right)^3}{3} - \dots \\
&\geq -\sum_{l=1}^{l^*} q_l - \left(\sum_{l=1}^{l^*} q_l\right)^2 - \left(\sum_{l=1}^{l^*} q_l\right)^3 - \dots \\
&= -\frac{\sum_{l=1}^{l^*} q_l}{1-\sum_{l=1}^{l^*} q_l}.
\end{aligned}$$

■

Let $q = \mathbb{P}\mathrm{r}_G(\mathcal{T}_{\epsilon,G}(i)^c)$. Then, $1-q = \Pr(\mathcal{T}_{\epsilon,G}(i)) = \prod_{i=1}^{l^*}(1-q_l)$, where $q_l = \mathbb{P}\mathrm{r}_G(w(\mathbf{E}_{(i),l}^{n[l]}) - gn[l] > \epsilon_l\sqrt{n[l]})$. A simple Chernoff bound (see, for example, Lemma 9 of [newWaterslide]) shows that $q_l \leq 2^{-\phi(g)\epsilon_l^2}$ for each l. Using $\epsilon_l = \epsilon_{l^*} 2^{\frac{l^*-l}{2}}$ and the Chernoff bound on q_l,

$$\begin{aligned}
\sum_{i=1}^{l^*} q_l &\leq \sum_{i=1}^{l^*} 2^{-\phi(g)\epsilon_l^2} = \sum_{i=1}^{l^*} 2^{-\phi(g)\epsilon_{l^*}^2 2^{l^*-l}} \\
&= 2^{-\phi(g)\epsilon_{l^*}^2 2^{l^*-1}} + 2^{-\phi(g)\epsilon_{l^*}^2 2^{l^*-2}} + \dots + 2^{-\phi(g)\epsilon_{l^*}^2 2} + 2^{-\phi(g)\epsilon_{l^*}^2} \\
&\leq \sum_{r=1}^{\infty} 2^{-r\phi(g)\epsilon_{l^*}^2} = \frac{2^{-\phi(g)\epsilon_{l^*}^2}}{1-2^{-\phi(g)\epsilon_{l^*}^2}}.
\end{aligned}$$

Thus, using Lemma 7,

$$\begin{aligned}
\mathbb{P}\mathrm{r}_G(\mathcal{T}_{\epsilon,G}(i)) &\geq \exp\left(-\frac{\frac{2^{-\phi(g)\epsilon_{l^*}^2}}{1-2^{-\phi(g)\epsilon_{l^*}^2}}}{1-\frac{2^{-\phi(g)\epsilon_{l^*}^2}}{1-2^{-\phi(g)\epsilon_{l^*}^2}}.}\right) \\
&= \exp\left(-\frac{2^{-\phi(g)\epsilon_{l^*}^2}}{1-2^{-\phi(g)\epsilon_{l^*}^2+1}}\right).
\end{aligned}$$

Thus, to ensure $\mathbb{P}\mathrm{r}_G(\mathcal{T}_{\epsilon,G}(i)) \geq 1-x$, it is enough to ensure that

$$\exp\left(-\frac{2^{-\phi(g)\epsilon_{l^*}^2}}{1-2^{-\phi(g)\epsilon_{l^*}^2+1}}\right) \geq 1-x, \tag{14}$$

which amounts to

$$\epsilon_{l^*}^2 \geq \frac{1}{\phi(g)}\log_2\left(\frac{1}{\ln\left(\frac{1}{1-x}\right)}+2\right). \tag{15}$$

Since the value of $\epsilon_{l^*}^2$ given by the RHS is sufficient, a larger value would be sufficient as well. Now notice that

$$\begin{aligned}\ln\left(\frac{1}{1-x}\right) &= -\ln(1-x)\\ &= x+\frac{1}{2}x^2+\frac{1}{3}x^3+\ldots\\ &\geq x\end{aligned}$$

Thus,

$$\frac{1}{\phi(g)}\log_2\left(\frac{1}{\ln\left(\frac{1}{1-x}\right)}+2\right) \leq \frac{1}{\phi(g)}\log_2\left(\frac{1}{x}+2\right).$$

It is therefore sufficient to choose $\epsilon_{l^*}(x)=\sqrt{\frac{1}{\phi(g)}\log_2\left(\frac{1}{x}+2\right)}$ to ensure that $\mathbb{P}\mathrm{r}_G(\mathcal{T}_{\epsilon,G}(i))\geq 1-x$.

Appendix III
Convexity and monotonicity of $f_G(x)$

Simplifying the expression for $f_G(x)$,

$$\begin{aligned}f_G(x) &= \frac{x}{2}\prod_{l=1}^{l^*}\left(2^{-n[l]D(g\|p_{CC,l})}\times\left(\frac{p_{CC,l}(1-g)}{g(1-p_{CC,l})}\right)^{\epsilon_{l^*}\left(\frac{x}{2}\right)2^{\frac{l^*-l}{2}}\sqrt{n[l]}}\right)\\ &= \frac{x}{2}\left[\prod_{l=1}^{l^*}2^{-n[l]D(g\|p_{CC,l})}\right]\exp\left(\sum_{l=1}^{l^*}\ln\left(\frac{p_{CC,l}(1-g)}{g(1-p_{CC,l})}\right)\epsilon_{l^*}\left(\frac{x}{2}\right)2^{\frac{l^*-l}{2}}\sqrt{n[l]}\right)\\ &= \beta x\exp\left(-d\sqrt{\ln\left(\frac{2}{x}+2\right)}\right),\end{aligned}$$

where $\beta=\frac{\prod_{l=1}^{l^*}2^{-n[l]D(g\|p_{CC,l})}}{2}$, and $d=-\sqrt{\frac{1}{\ln(2)\phi(g)}}\sum_{l=1}^{l^*}\ln\left(\frac{p_{CC,l}(1-g)}{g(1-p_{CC,l})}\right)2^{\frac{l^*-l}{2}}\sqrt{n[l]}$ do not depend on x and are both larger than 0.

Differentiating $f_G(x)=\beta x\exp\left(-d\sqrt{\ln\left(\frac{2}{x}+2\right)}\right)$ with respect to x,

$$\frac{d}{dx}f_G(x)=\beta\exp\left(-d\sqrt{\ln\left(2+\frac{2}{x}\right)}\right)+\beta\frac{d\exp\left(-d\sqrt{\ln\left(2+\frac{2}{x}\right)}\right)}{x\left(2+\frac{2}{x}\right)\sqrt{\ln\left(2+\frac{2}{x}\right)}}, \tag{16}$$

which is positive for $d>0$ for all $x>0$. Thus $f_G(\cdot)$ is a monotonically increasing function. Differentiating $f_G(\cdot)$ once more,

$$\frac{d^2}{dx^2}f_G(x)=\frac{\beta d\exp\left(-d\sqrt{\ln\left(2+\frac{2}{x}\right)}\right)}{\left(2+\frac{2}{x}\right)^2x^3\left(\ln\left(2+\frac{2}{x}\right)\right)^{\frac{3}{2}}}+\frac{\beta d^2\exp\left(-d\sqrt{\ln\left(2+\frac{2}{x}\right)}\right)}{\left(2+\frac{2}{x}\right)^2x^3\ln\left(2+\frac{2}{x}\right)}+\frac{2\beta d\exp\left(-d\sqrt{\ln\left(2+\frac{2}{x}\right)}\right)}{\left(2+\frac{2}{x}\right)^2x^3\sqrt{\ln\left(2+\frac{2}{x}\right)}}, \tag{17}$$

which is again strictly positive for $d>0$ for all x. Thus $f_G(x)$ is a convex-$\cup$ increasing function of x.

Appendix IV
Fano-style inequality for test channel G

Under the test channel G, all the channel errors until level l^* behave as Ber(g). All the channel errors at subsequent levels $l=l^*+1,\ldots,L$ behave as Ber($p_{CC,l}$), $p_{CC,l}>g$. Clearly, $\mathbf{Y}_{CC,(i)}^n$ can be written as $\mathbf{Y}_{CC,(i)}^n=\mathbf{Y}_{(i)}^n\oplus\overline{\mathbf{E}}_{(i)}^n$, where $\mathbf{Y}_{(i)}^n=\mathbf{X}_{(i)}^n\oplus\mathbf{E}_{(i)}^n$, where the elements of $\mathbf{E}_{(i)}^n$ are distributed Ber(g), and $\overline{\mathbf{E}}_{(i)}^n$ is an additional independent (and identical across each level l) noise such that $E_{i,j}\oplus\overline{E}_{i,j}\sim$ Ber($p_{CC,l}$) if $\ell(i,j)=l$. In this proof, all probabilities are with respect to this G test channel.

We wish to show $\langle P_e\rangle_G\geq h_b^{-1}(\delta(g))$. Notice the Markov chain B_i— $\mathbf{Y}_{(i)}^n$ — $\mathbf{Y}_{CC,(i)}^n$—$\widehat{B}_i$, where $\widehat{B}_i$ is the decoded estimate of B_i. Then, we obtain the following sequence of inequalities

$$\begin{aligned}H(B_i)-H(B_i|\mathbf{Y}_{(i)}^n) &= I(B_i;\mathbf{Y}_{(i)}^n)\\ &\overset{(a)}{\geq} I(B_i;\mathbf{Y}_{CC,(i)}^n)\\ &= H(B_i)-H(B_i|\mathbf{Y}_{CC,(i)}^n),\end{aligned}$$

where (a) follows from the data-processing inequality. Thus,

$$H(B_i|\mathbf{Y}^n_{CC,(i)}) \geq H(B_i|\mathbf{Y}^n_{(i)}). \tag{18}$$

Again using the Markov chain relationship, $I(B_i;\widehat{B}_i|\mathbf{Y}^n_{(i)}) = 0$. Therefore,

$$\begin{aligned}
H(B_i \oplus \widehat{B}_i|\mathbf{Y}^n_{CC,(i)}) &= H(B_i \oplus \widehat{B}_i|\mathbf{Y}^n_{CC,(i)}) + I(B_i;\widehat{B}_i|\mathbf{Y}^n_{(i)}) \\
&= H(B_i \oplus \widehat{B}_i|\mathbf{Y}^n_{CC,(i)}) + H(B_i|\mathbf{Y}^n_{CC,(i)}) - H(B_i|\widehat{B}_i,\mathbf{Y}^n_{CC,(i)}) \\
&= H(B_i \oplus \widehat{B}_i|\mathbf{Y}^n_{CC,(i)}) + H(B_i|\mathbf{Y}^n_{CC,(i)}) - H(B_i \oplus \widehat{B}_i|\widehat{B}_i,\mathbf{Y}^n_{CC,(i)}) \\
&= H(B_i \oplus \widehat{B}_i|\mathbf{Y}^n_{CC,(i)}) + H(B_i|\mathbf{Y}^n_{CC,(i)}) - H(B_i \oplus \widehat{B}_i|\widehat{B}_i,\mathbf{Y}^n_{CC,(i)}) \\
&= I(B_i \oplus \widehat{B}_i;\widehat{B}_i|\mathbf{Y}^n_{CC,(i)}) + H(B_i|\mathbf{Y}^n_{CC,(i)}) \\
&\geq H(B_i|\mathbf{Y}^n_{CC,(i)}).
\end{aligned}$$

Since conditioning reduces entropy,

$$\begin{aligned}
H(B_i \oplus \widehat{B}_i) &\geq H(B_i \oplus \widehat{B}_i|\mathbf{Y}^n_{CC,(i)}) \\
&\geq H(B_i|\mathbf{Y}^n_{CC,(i)}) \\
&\geq H(B_i|\mathbf{Y}^n_{(i)}) \\
&\geq H(B_i|\mathbf{Y}^m) \\
&\geq H(B_i|\mathbf{Y}^m,\mathbf{B}^{i-1}).
\end{aligned}$$

Now notice that $B_i \oplus \widehat{B}_i$ is a binary random variable with $\mathbb{P}\text{r}(B_i \oplus \widehat{B}_i = 1) = \langle P_{e,i}\rangle$. Thus,

$$h_b(\langle P_{e,i}\rangle) = H(B_i \oplus \widehat{B}_i). \tag{19}$$

Therefore,

$$\begin{aligned}
\frac{1}{k}\sum_{i=1}^{k} h_b(\langle P_{e,i}\rangle) &\geq \frac{1}{k}\sum_{i=1^k} H(B_i|\mathbf{Y}^m,\mathbf{B}^{i-1}) \\
&= \frac{1}{k}H(\mathbf{B}^k|\mathbf{Y}^m) \\
&= \frac{1}{k}\left(H(\mathbf{B}^k) - I(\mathbf{B}^k;\mathbf{Y})\right) \\
&= 1 - \frac{1}{k}I(\mathbf{B}^k;\mathbf{Y}^m) \\
&\geq 1 - \frac{C_{BSC(g)}}{R}.
\end{aligned}$$

The lemma now follows from the observation that $h_b(\cdot)$ is a concave function, and thus $h_b(\langle P_e\rangle_G) \geq \frac{1}{k}\sum_{i=1}^{k} h_b(\langle P_{e,i}\rangle_G) \geq 1 - \frac{C_{BSC(g)}}{R}$.