Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels

Erdal Arıkan, *Senior Member, IEEE*

***Abstract*—A method is proposed, called channel polarization, to construct code sequences that achieve the symmetric capacity $I(W)$ of any given binary-input discrete memoryless channel (B-DMC) W. The symmetric capacity is the highest rate achievable subject to using the input letters of the channel with equal probability. Channel polarization refers to the fact that it is possible to synthesize, out of N independent copies of a given B-DMC W, a second set of N binary-input channels $\{W_N^{(i)} : 1 \le i \le N\}$ such that, as N becomes large, the fraction of indices i for which $I(W_N^{(i)})$ is near 1 approaches $I(W)$ and the fraction for which $I(W_N^{(i)})$ is near 0 approaches $1 - I(W)$. The polarized channels $\{W_N^{(i)}\}$ are well-conditioned for channel coding: one need only send data at rate 1 through those with capacity near 1 and at rate 0 through the remaining. Codes constructed on the basis of this idea are called polar codes. The paper proves that, given any B-DMC W with $I(W) > 0$ and any target rate $R < I(W)$, there exists a sequence of polar codes $\{\mathcal{C}_n; n \ge 1\}$ such that $\mathcal{C}_n$ has block-length $N = 2^n$, rate $\ge R$, and probability of block error under successive cancellation decoding bounded as $P_e(N, R) \le O(N^{-\frac{1}{4}})$ independently of the code rate. This performance is achievable by encoders and decoders with complexity $O(N \log N)$ for each.**

***Index Terms*—Capacity-achieving codes, channel capacity, channel polarization, Plotkin construction, polar codes, Reed–Muller (RM) codes, successive cancellation decoding.**

I. Introduction and Overview

A FASCINATING aspect of Shannon's proof of the noisy channel coding theorem is the random-coding method that he used to show the existence of capacity-achieving code sequences without exhibiting any specific such sequence [1]. Explicit construction of provably capacity-achieving code sequences with low encoding and decoding complexities has since then been an elusive goal. This paper is an attempt to meet this goal for the class of binary-input discrete memoryless channels (B-DMCs).

We will give a description of the main ideas and results of the paper in this section. First, we give some definitions and state some basic facts that are used throughout the paper.

Manuscript received October 14, 2007; revised August 13, 2008. Current version published June 24, 2009. This work was supported in part by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under Project 107E216 and in part by the European Commission FP7 Network of Excellence NEWCOM++ under Contract 216715. The material in this paper was presented in part at the IEEE International Symposium on Information Theory (ISIT), Toronto, ON, Canada, July 2008.

The author is with the Department of Electrical-Electronics Engineering, Bilkent University, Ankara, 06800, Turkey (e-mail: arikan@ee.bilkent.edu.tr).

Communicated by Y. Steinberg, Associate Editor for Shannon Theory.

Color versions of Figures 4 and 7 in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2009.2021379

A. Preliminaries

We write $W : \mathcal{X} \to \mathcal{Y}$ to denote a generic B-DMC with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and transition probabilities $W(y|x)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$. The input alphabet $\mathcal{X}$ will always be $\{0, 1\}$, the output alphabet and the transition probabilities may be arbitrary. We write W^N to denote the channel corresponding to N uses of W; thus, $W^N : \mathcal{X}^N \to \mathcal{Y}^N$ with $W^N(y_1^N|x_1^N) = \prod_{i=1}^N W(y_i|x_i)$.

Given a B-DMC W, there are two channel parameters of primary interest in this paper: the symmetric capacity

$$I(W) \triangleq \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)}$$

and the Bhattacharyya parameter

$$Z(W) \triangleq \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)}.$$

These parameters are used as measures of *rate* and *reliability*, respectively. $I(W)$ is the highest rate at which reliable communication is possible across W using the inputs of W with equal frequency. $Z(W)$ is an upper bound on the probability of maximum-likelihood (ML) decision error when W is used only once to transmit a 0 or 1.

It is easy to see that $Z(W)$ takes values in $[0, 1]$. Throughout, we will use base-2 logarithms; hence, $I(W)$ will also take values in $[0, 1]$. The unit for code rates and channel capacities will be *bits*.

Intuitively, one would expect that $I(W) \approx 1$ iff $Z(W) \approx 0$, and $I(W) \approx 0$ iff $Z(W) \approx 1$. The following bounds, proved in the Appendix, make this precise.

Proposition 1: For any B-DMC W, we have

$$I(W) \ge \log \frac{2}{1 + Z(W)} \tag{1}$$

$$I(W) \le \sqrt{1 - Z(W)^2}. \tag{2}$$

The symmetric capacity $I(W)$ equals the Shannon capacity when W is a *symmetric* channel, i.e., a channel for which there exists a permutation π of the output alphabet $\mathcal{Y}$ such that i) $\pi^{-1} = \pi$ and ii) $W(y|1) = W(\pi(y)|0)$ for all $y \in \mathcal{Y}$. The binary symmetric channel (BSC) and the binary erasure channel (BEC) are examples of symmetric channels. A BSC is a B-DMC W with $\mathcal{Y} = \{0, 1\}$, $W(0|0) = W(1|1)$, and $W(1|0) = W(0|1)$. A B-DMC W is called a BEC if for each $y \in \mathcal{Y}$, either $W(y|0)W(y|1) = 0$ or $W(y|0) = W(y|1)$. In the latter case,

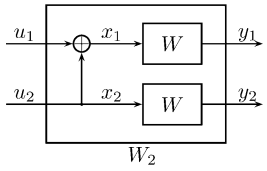

Fig. 1. The channel W_2.

y is said to be an *erasure* symbol. The sum of $W(y|0)$ over all erasure symbols y is called the erasure probability of the BEC.

We denote random variables (RVs) by upper case letters, such as X, Y, and their realizations (sample values) by the corresponding lower case letters, such as x, y. For X an RV, P_X denotes the probability assignment on X. For a joint ensemble of RVs (X, Y), $P_{X,Y}$ denotes the joint probability assignment. We use the standard notation $I(X;Y), I(X;Y|Z)$ to denote the mutual information and its conditional form, respectively.

We use the notation a_1^N as shorthand for denoting a row vector $(a_1, \ldots, a_N)$. Given such a vector a_1^N, we write a_i^j, $1 \leq i, j \leq N$, to denote the subvector $(a_i, \ldots, a_j)$; if $j < i$, a_i^j is regarded as void. Given a_1^N and $\mathcal{A} \subset \{1, \ldots, N\}$, we write $a_{\mathcal{A}}$ to denote the subvector $(a_i : i \in \mathcal{A})$. We write $a_{1,o}^j$ to denote the subvector with odd indices $(a_k : 1 \leq k \leq j; k \text{ odd})$. We write $a_{1,e}^j$ to denote the subvector with even indices $(a_k : 1 \leq k \leq j; k \text{ even})$. For example, for $a_1^5 = (5, 4, 6, 2, 1)$, we have $a_2^4 = (4, 6, 2)$, $a_{1,e}^5 = (4, 2)$, $a_{1,o}^4 = (5, 6)$. The notation 0_1^N is used to denote the all-zero vector.

Code constructions in this paper will be carried out in vector spaces over the binary field GF (2). Unless specified otherwise, all vectors, matrices, and operations on them will be over GF (2). In particular, for a_1^N, b_1^N vectors over GF (2) we write $a_1^N \oplus b_1^N$ to denote their componentwise mod-2 sum. The Kronecker product of an m-by-n matrix $A = [A_{ij}]$ and an r-by-s matrix $B = [B_{ij}]$ is defined as

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{bmatrix}$$

which is an mr-by-ns matrix. The Kronecker power $A^{\otimes n}$ is defined as $A \otimes A^{\otimes(n-1)}$ for all $n \geq 1$. We will follow the convention that $A^{\otimes 0} \triangleq [1]$.

We write $|\mathcal{A}|$ to denote the number of elements in a set $\mathcal{A}$. We write $1_{\mathcal{A}}$ to denote the indicator function of a set $\mathcal{A}$; thus, $1_{\mathcal{A}}(x)$ equals 1 if $x \in \mathcal{A}$ and 0 otherwise.

We use the standard Landau notation $O(N)$, $o(N)$, $\omega(N)$ to denote the asymptotic behavior of functions.

B. Channel Polarization

Channel polarization is an operation by which one manufactures out of N independent copies of a given B-DMC W a second set of N channels $\{W_N^{(i)} : 1 \leq i \leq N\}$ that show a polarization effect in the sense that, as N becomes large, the symmetric capacity terms $\{I(W_N^{(i)})\}$ tend towards 0 or 1 for all but a vanishing fraction of indices i. This operation consists of a channel combining phase and a channel splitting phase.

1) Channel Combining: This phase combines copies of a given B-DMC W in a recursive manner to produce a vector channel $W_N : \mathcal{X}^N \to \mathcal{Y}^N$, where N can be any power of two, $N = 2^n$, $n \geq 0$. The recursion begins at the 0th level $(n = 0)$ with only one copy of W and we set $W_1 \triangleq W$. The first level $(n = 1)$ of the recursion combines two independent copies of W_1 as shown in Fig. 1 and obtains the channel $W_2 : \mathcal{X}^2 \to \mathcal{Y}^2$ with the transition probabilities

$$W_2(y_1, y_2|u_1, u_2) = W(y_1|u_1 \oplus u_2)W(y_2|u_2). \qquad (3)$$

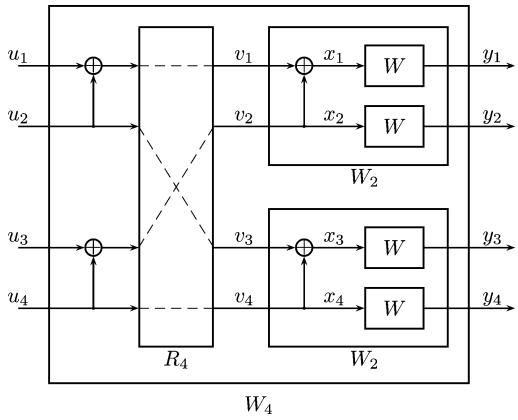

Fig. 2. The channel W_4 and its relation to W_2 and W.

The next level of the recursion is shown in Fig. 2 where two independent copies of W_2 are combined to create the channel $W_4 : \mathcal{X}^4 \to \mathcal{Y}^4$ with transition probabilities $W_4(y_1^4|u_1^4) = W_2(y_1^2|u_1 \oplus u_2, u_3 \oplus u_4)W_2(y_3^4|u_2, u_4)$.

In Fig. 2, R_4 is the permutation operation that maps an input (s_1, s_2, s_3, s_4) to $v_1^4 = (s_1, s_3, s_2, s_4)$. The mapping $u_1^4 \mapsto x_1^4$ from the input of W_4 to the input of W^4 can be written as $x_1^4 = u_1^4 G_4$ with

$$G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Thus, we have the relation $W_4(y_1^4|u_1^4) = W^4(y_1^4|u_1^4 G_4)$ between the transition probabilities of W_4 and those of W^4.

The general form of the recursion is shown in Fig. 3 where two independent copies of $W_{N/2}$ are combined to produce the channel W_N. The input vector u_1^N to W_N is first transformed into s_1^N so that $s_{2i-1} = u_{2i-1} \oplus u_{2i}$ and $s_{2i} = u_{2i}$ for $1 \leq i \leq N/2$. The operator R_N in the figure is a permutation, known as the *reverse shuffle* operation, and acts on its input s_1^N to produce $v_1^N = (s_1, s_3, \ldots, s_{N-1}, s_2, s_4, \ldots, s_N)$, which becomes the input to the two copies of $W_{N/2}$ as shown in the figure.

We observe that the mapping $u_1^N \mapsto v_1^N$ is linear over GF (2). It follows by induction that the overall mapping $u_1^N \mapsto x_1^N$, from the input of the synthesized channel W_N to the input of the underlying raw channels W^N, is also linear and may be represented by a matrix G_N so that $x_1^N = u_1^N G_N$. We call G_N the *generator matrix* of size N. The transition probabilities of the two channels W_N and W^N are related by

$$W_N\left(y_1^N|u_1^N\right) = W^N\left(y_1^N|u_1^N G_N\right) \qquad (4)$$

for all $y_1^N \in \mathcal{Y}^N$, $u_1^N \in \mathcal{X}^N$. We will show in Section VII that G_N equals $B_N F^{\otimes n}$ for any $N = 2^n$, $n \geq 0$, where B_N is a permutation matrix known as *bit-reversal* and $F \triangleq \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

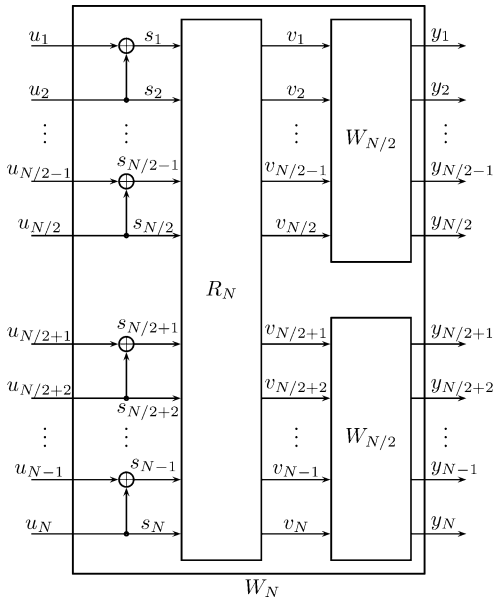

Fig. 3. Recursive construction of W_N from two copies of $W_{N/2}$.

Note that the channel combining operation is fully specified by the matrix F. Also note that G_N and $F^{\otimes n}$ have the same set of rows, but in a different (bit-reversed) order; we will discuss this topic more fully in Section VII.

2) Channel Splitting: Having synthesized the vector channel W_N out of W^N, the next step of channel polarization is to split W_N back into a set of N binary-input coordinate channels $W_N^{(i)} : \mathcal{X} \to \mathcal{Y}^N \times \mathcal{X}^{i-1}, 1 \leq i \leq N$, defined by the transition probabilities

$$W_N^{(i)}\left(y_1^N, u_1^{i-1}|u_i\right) \triangleq \sum_{u_{i+1}^N \in \mathcal{X}^{N-i}} \frac{1}{2^{N-1}} W_N\left(y_1^N|u_1^N\right) \tag{5}$$

where (y_1^N, u_1^{i-1}) denotes the output of $W_N^{(i)}$ and u_i its input.

To gain an intuitive understanding of the channels $\{W_N^{(i)}\}$, consider a genie-aided successive cancellation decoder in which the ith decision element estimates u_i after observing y_1^N and the *past* channel inputs u_1^{i-1} (supplied correctly by the genie regardless of any decision errors at earlier stages). If u_1^N is *a priori* uniform on $\mathcal{X}^N$, then $W_N^{(i)}$ is the effective channel seen by the ith decision element in this scenario.

3) Channel Polarization:

Theorem 1: For any B-DMC W, the channels $\{W_N^{(i)}\}$ *polarize* in the sense that, for any fixed $\delta \in (0,1)$, as N goes to infinity through powers of two, the fraction of indices $i \in \{1,\ldots,N\}$ for which $I(W_N^{(i)}) \in (1-\delta, 1]$ goes to $I(W)$ and the fraction for which $I(W_N^{(i)}) \in [0,\delta)$ goes to $1 - I(W)$.

This theorem is proved in Section IV.

The polarization effect is illustrated in Fig. 4 for the case W is a BEC with erasure probability $\epsilon = 0.5$. The numbers $\{I(W_N^{(i)})\}$ have been computed using the recursive relations

$$\begin{aligned} I\left(W_N^{(2i-1)}\right) &= I\left(W_{N/2}^{(i)}\right)^2 \\ I\left(W_N^{(2i)}\right) &= 2I\left(W_{N/2}^{(i)}\right) - I\left(W_{N/2}^{(i)}\right)^2 \end{aligned} \tag{6}$$

with $I(W_1^{(1)}) = 1 - \epsilon$. This recursion is valid only for BECs and it is proved in Section III. No efficient algorithm is known for calculation of $\{I(W_N^{(i)})\}$ for a general B-DMC W.

Fig. 4 shows that $I(W^{(i)})$ tends to be near 0 for small i and near 1 for large i. However, $I(W_N^{(i)})$ shows an erratic behavior for an intermediate range of i. For general B-DMCs, determining the subset of indices i for which $I(W_N^{(i)})$ is above a given threshold is an important computational problem that will be addressed in Section IX.

4) Rate of Polarization: For proving coding theorems, the speed with which the polarization effect takes hold as a function of N is important. Our main result in this regard is given in terms of the parameters

$$\begin{aligned} &Z\left(W_N^{(i)}\right) \\ &= \sum_{y_1^N \in \mathcal{Y}^N} \sum_{u_1^{i-1} \in \mathcal{X}^{i-1}} \sqrt{W_N^{(i)}\left(y_1^N, u_1^{i-1}|0\right) W_N^{(i)}\left(y_1^N, u_1^{i-1}|1\right)}. \end{aligned} \tag{7}$$

Theorem 2: For any B-DMC W with $I(W) > 0$, and any fixed $R < I(W)$, there exists a sequence of sets $\mathcal{A}_N \subset \{1,\ldots,N\}$, $N \in \{1,2,\ldots,2^n,\ldots\}$, such that $|\mathcal{A}_N| \geq NR$ and $Z(W_N^{(i)}) \leq O(N^{-5/4})$ for all $i \in \mathcal{A}_N$.

This theorem is proved in Section IV-B.

We stated the polarization result in Theorem 2 in terms $\{Z(W_N^{(i)})\}$ rather than $\{I(W_N^{(i)})\}$ because this form is better suited to the coding results that we will develop. A rate of polarization result in terms of $\{I(W_N^{(i)})\}$ can be obtained from Theorem 2 with the help of Proposition 1.

C. Polar Coding

We take advantage of the polarization effect to construct codes that achieve the symmetric channel capacity $I(W)$ by a method we call *polar coding*. The basic idea of polar coding is to create a coding system where one can access each coordinate channel $W_N^{(i)}$ individually and send data only through those for which $Z(W_N^{(i)})$ is near 0.

1) G_N-Coset Codes: We first describe a class of block codes that contain polar codes—the codes of main interest—as a special case. The block lengths N for this class are restricted to powers of two, $N = 2^n$ for some $n \geq 0$. For a given N, each code in the class is encoded in the same manner, namely

$$x_1^N = u_1^N G_N \tag{8}$$

where G_N is the generator matrix of order N, defined above. For $\mathcal{A}$ an arbitrary subset of $\{1,\ldots,N\}$, we may write (8) as

$$x_1^N = u_{\mathcal{A}} G_N(\mathcal{A}) \oplus u_{\mathcal{A}^c} G_N(\mathcal{A}^c) \tag{9}$$

where $G_N(\mathcal{A})$ denotes the submatrix of G_N formed by the rows with indices in $\mathcal{A}$.

If we now fix $\mathcal{A}$ and $u_{\mathcal{A}^c}$, but leave $u_{\mathcal{A}}$ as a free variable, we obtain a mapping from source blocks $u_{\mathcal{A}}$ to codeword blocks x_1^N. This mapping is a *coset code*: it is a coset of the linear block

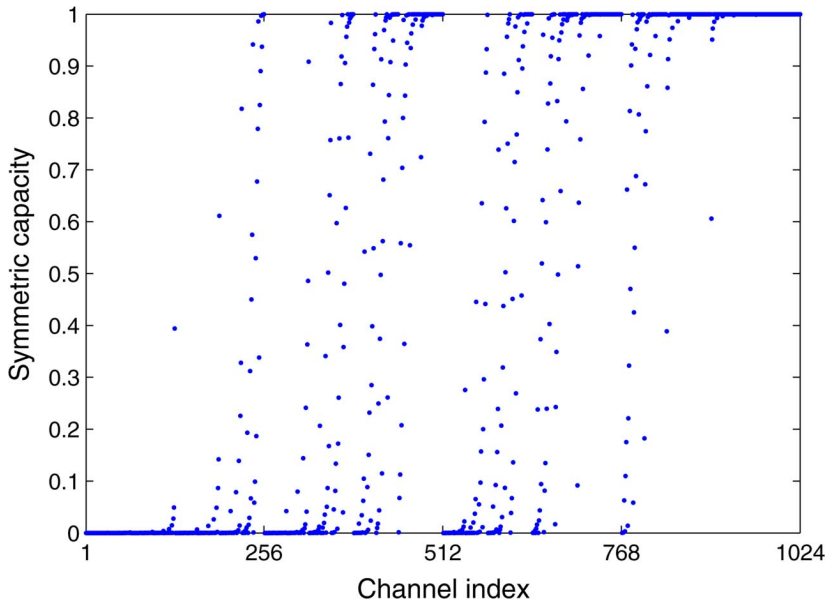

Fig. 4. Plot of $I(W_N^{(i)})$ versus $i = 1, \ldots, N = 2^{10}$ for a BEC with $\epsilon = 0.5$.

code with generator matrix $G_N(\mathcal{A})$, with the coset determined by the fixed vector $u_{\mathcal{A}^c}G_N(\mathcal{A}^c)$. We will refer to this class of codes collectively as G_N-*coset codes*. Individual G_N-coset codes will be identified by a parameter vector $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$, where K is the code dimension and specifies the size of $\mathcal{A}$.[1] The ratio K/N is called the *code rate*. We will refer to $\mathcal{A}$ as the *information set* and to $u_{\mathcal{A}^c} \in \mathcal{X}^{N-K}$ as *frozen* bits or vector.

For example, the $(4, 2, \{2, 4\}, (1, 0))$ code has the encoder mapping

$$x_1^4 = u_1^4 G_4 = (u_2, u_4)\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} + (1, 0)\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}. \quad (10)$$

For a source block $(u_2, u_4) = (1, 1)$, the coded block is $x_1^4 = (1, 1, 0, 1)$.

Polar codes will be specified shortly by giving a particular rule for the selection of the information set $\mathcal{A}$.

2) A Successive Cancellation Decoder: Consider a G_N-coset code with parameter $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$. Let u_1^N be encoded into a codeword x_1^N, let x_1^N be sent over the channel W^N, and let a channel output y_1^N be received. The decoder's task is to generate an estimate $\hat{u}_1^N$ of u_1^N, given knowledge of $\mathcal{A}$, $u_{\mathcal{A}^c}$, and y_1^N. Since the decoder can avoid errors in the frozen part by setting $\hat{u}_{\mathcal{A}^c} = u_{\mathcal{A}^c}$, the real decoding task is to generate an estimate $\hat{u}_{\mathcal{A}}$ of $u_{\mathcal{A}}$.

The coding results in this paper will be given with respect to a specific successive cancellation (SC) decoder, unless some other decoder is mentioned. Given any $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ G_N-coset code, we will use an SC decoder that generates its decision $\hat{u}_1^N$ by computing

$$\hat{u}_i \triangleq \begin{cases} u_i, & \text{if } i \in \mathcal{A}^c \\ h_i\left(y_1^N, \hat{u}_1^{i-1}\right), & \text{if } i \in \mathcal{A} \end{cases} \quad (11)$$

[1]We include the redundant parameter K in the parameter set because often we consider an ensemble of codes with K fixed and $\mathcal{A}$ free.

in the order i from 1 to N, where $h_i : \mathcal{Y}^N \times \mathcal{X}^{i-1} \to \mathcal{X}, i \in \mathcal{A}$, are *decision functions* defined as

$$h_i\left(y_1^N, \hat{u}_1^{i-1}\right) \triangleq \begin{cases} 0, & \text{if } \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)} \geq 1 \\ 1, & \text{otherwise} \end{cases} \quad (12)$$

for all $y_1^N \in \mathcal{Y}^N$, $\hat{u}_1^{i-1} \in \mathcal{X}^{i-1}$. We will say that a decoder *block error* occurred if $\hat{u}_1^N \neq u_1^N$ or equivalently if $\hat{u}_{\mathcal{A}} \neq u_{\mathcal{A}}$.

The decision functions $\{h_i\}$ defined above resemble ML decision functions but are not exactly so, because they treat the *future* frozen bits $(u_j : j > i, j \in \mathcal{A}^c)$ as RVs, rather than as known bits. In exchange for this suboptimality, $\{h_i\}$ can be computed efficiently using recursive formulas, as we will show in Section II. Apart from algorithmic efficiency, the recursive structure of the decision functions is important because it renders the performance analysis of the decoder tractable. Fortunately, the loss in performance due to not using true ML decision functions happens to be negligible: $I(W)$ is still achievable.

3) Code Performance: The notation $P_e(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ will denote the probability of block error for an $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ code, assuming that each data vector $u_{\mathcal{A}} \in \mathcal{X}^K$ is sent with probability 2^{-K} and decoding is done by the above SC decoder. More precisely

$$P_e(N, K, \mathcal{A}, u_{\mathcal{A}^c}) \triangleq \sum_{u_{\mathcal{A}} \in \mathcal{X}^K} \frac{1}{2^K} \sum_{y_1^N \in \mathcal{Y}^N : \hat{u}_1^N(y_1^N) \neq u_1^N} W_N\left(y_1^N | u_1^N\right).$$

The average of $P_e(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ over all choices for $u_{\mathcal{A}^c}$ will be denoted by $P_e(N, K, \mathcal{A})$, i.e.,

$$P_e(N, K, \mathcal{A}) \triangleq \sum_{u_{\mathcal{A}^c} \in \mathcal{X}^{N-K}} \frac{1}{2^{N-K}} P_e(N, K, \mathcal{A}, u_{\mathcal{A}^c}).$$

A key bound on block error probability under SC decoding is the following.

Proposition 2: For any B-DMC W and any choice of the parameters $(N, K, \mathcal{A})$

$$P_e(N, K, \mathcal{A}) \leq \sum_{i \in \mathcal{A}} Z\left(W_N^{(i)}\right). \tag{13}$$

Hence, for each $(N, K, \mathcal{A})$, there exists a frozen vector $u_{\mathcal{A}^c}$ such that

$$P_e(N, K, \mathcal{A}, u_{\mathcal{A}^c}) \leq \sum_{i \in \mathcal{A}} Z\left(W_N^{(i)}\right). \tag{14}$$

This is proved in Section V-B. This result suggests choosing $\mathcal{A}$ from among all K-subsets of $\{1, \ldots, N\}$ so as to minimize the right-hand side (RHS) of (13). This idea leads to the definition of polar codes.

4) Polar Codes: Given a B-DMC W, a G_N-coset code with parameter $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ will be called a *polar code* for W if the information set $\mathcal{A}$ is chosen as a K-element subset of $\{1, \ldots, N\}$ such that $Z(W_N^{(i)}) \leq Z(W_N^{(j)})$ for all $i \in \mathcal{A}$, $j \in \mathcal{A}^c$.

Polar codes are channel-specific designs: a polar code for one channel may not be a polar code for another. The main result of this paper will be to show that polar coding achieves the symmetric capacity $I(W)$ of any given B-DMC W.

An alternative rule for polar code definition would be to specify $\mathcal{A}$ as a K-element subset of $\{1, \ldots, N\}$ such that $I(W_N^{(i)}) \geq I(W_N^{(j)})$ for all $i \in \mathcal{A}$, $j \in \mathcal{A}^c$. This alternative rule would also achieve $I(W)$. However, the rule based on the Bhattacharyya parameters has the advantage of being connected with an explicit bound on block error probability.

The polar code definition does not specify how the frozen vector $u_{\mathcal{A}^c}$ is to be chosen; it may be chosen at will. This degree of freedom in the choice of $u_{\mathcal{A}^c}$ simplifies the performance analysis of polar codes by allowing averaging over an ensemble. However, it is not for analytical convenience alone that we do not specify a precise rule for selecting $u_{\mathcal{A}^c}$, but also because it appears that the code performance is relatively insensitive to that choice. In fact, we prove in Section VI-B that, for symmetric channels, any choice for $u_{\mathcal{A}^c}$ is as good as any other.

5) Coding Theorems: Fix a B-DMC W and a number $R \geq 0$. Let $P_e(N, R)$ be defined as $P_e(N, \lfloor NR \rfloor, \mathcal{A})$ with $\mathcal{A}$ selected in accordance with the polar coding rule for W. Thus, $P_e(N, R)$ is the probability of block error under SC decoding for polar coding over W with block length N and rate R, averaged over all choices for the frozen bits $u_{\mathcal{A}^c}$. The main coding result of this paper is the following.

Theorem 3: For any given B-DMC W and fixed $R < I(W)$, block error probability for polar coding under successive cancellation decoding satisfies

$$P_e(N, R) = O\left(N^{-\frac{1}{4}}\right). \tag{15}$$

This theorem follows as an easy corollary to Theorem 2 and the bound (13), as we show in Section V-B. For symmetric channels, we have the following stronger version of Theorem 3.

Theorem 4: For any symmetric B-DMC W and any fixed $R < I(W)$, consider any sequence of G_N-coset codes $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ with N increasing to infinity, $K = \lfloor NR \rfloor$, $\mathcal{A}$ chosen in accordance with the polar coding rule for W, and $u_{\mathcal{A}^c}$ fixed arbitrarily. The block error probability under successive cancellation decoding satisfies

$$P_e(N, K, \mathcal{A}, u_{\mathcal{A}^c}) = O\left(N^{-\frac{1}{4}}\right). \tag{16}$$

This is proved in Section VI-B. Note that for symmetric channels $I(W)$ equals the Shannon capacity of W.

6) Complexity: An important issue about polar coding is the complexity of encoding, decoding, and code construction. The recursive structure of the channel polarization construction leads to low-complexity encoding and decoding algorithms for the class of G_N-coset codes, and in particular, for polar codes.

Theorem 5: For the class of G_N-coset codes, the complexity of encoding and the complexity of successive cancellation decoding are both $O(N \log N)$ as functions of code block length N.

This theorem is proved in Sections VII and VIII. Notice that the complexity bounds in Theorem 5 are independent of the code rate and the way the frozen vector is chosen. The bounds hold even at rates above $I(W)$, but clearly this has no practical significance.

As for code construction, we have found no low-complexity algorithms for constructing polar codes. One exception is the case of a BEC for which we have a polar code construction algorithm with complexity $O(N)$. We discuss the code construction problem further in Section IX and suggest a low-complexity statistical algorithm for approximating the exact polar code construction.

D. Relations To Previous Work

This paper is an extension of work begun in [2], where channel combining and splitting were used to show that improvements can be obtained in the sum cutoff rate for some specific DMCs. However, no recursive method was suggested there to reach the ultimate limit of such improvements.

As the present work progressed, it became clear that polar coding had much in common with Reed–Muller (RM) coding [3], [4]. Indeed, recursive code construction and SC decoding, which are two essential ingredients of polar coding, appear to have been introduced into coding theory by RM codes.

According to one construction of RM codes, for any $N = 2^n$, $n \geq 0$, and $0 \leq K \leq N$, an RM code with block length N and dimension K, denoted $\mathrm{RM}(N, K)$, is defined as a linear code whose generator matrix $G_{\mathrm{RM}}(N, K)$ is obtained by deleting $(N - K)$ of the rows of $F^{\otimes n}$ so that none of the deleted rows has a larger Hamming weight (number of 1's in that row) than any of the remaining K rows. For instance

$$G_{\mathrm{RM}}(4, 4) = F^{\otimes 2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{and}$$

$$G_{\mathrm{RM}}(4, 2) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

This construction brings out the similarities between RM codes and polar codes. Since G_N and $F^{\otimes n}$ have the same set of rows (only in a different order) for any $N = 2^n$, it is clear that RM codes belong to the class of G_N-coset codes.

For example, $\mathrm{RM}(4,2)$ is the G_4-coset code with parameter $(4,2,\{2,4\},(0,0))$. So, RM coding and polar coding may be regarded as two alternative rules for selecting the information set $\mathcal{A}$ of a G_N-coset code of a given size (N,K). Unlike polar coding, RM coding selects the information set in a channel-independent manner; it is not as fine-tuned to the channel polarization phenomenon as polar coding is. We will show in Section X that, at least for the class of BECs, the RM rule for information set selection leads to asymptotically unreliable codes under SC decoding. So, polar coding goes beyond RM coding in a nontrivial manner by paying closer attention to channel polarization.

Another connection to existing work can be established by noting that polar codes are multilevel $|u|u+v|$ codes, which are a class of codes originating from Plotkin's method for code combining [5]. This connection is not surprising in view of the fact that RM codes are also multilevel $|u|u+v|$ codes [6, pp. 114–125]. However, unlike typical multilevel code constructions, where one begins with specific small codes to build larger ones, in polar coding the multilevel code is obtained by expurgating rows of a full-order generator matrix G_N, with respect to a channel-specific criterion. The special structure of G_N ensures that, no matter how expurgation is done, the resulting code is a multilevel $|u|u+v|$ code. In essence, polar coding enjoys the freedom to pick a multilevel code from an ensemble of such codes so as to suit the channel at hand, while conventional approaches to multilevel coding do not have this degree of flexibility.

Finally, we wish to mention a "spectral" interpretation of polar codes which is similar to Blahut's treatment of Bose–Chaudhuri–Hocquenghem (BCH) codes [7, Ch. 9]; this type of similarity has already been pointed out by Forney [8, Ch. 11] in connection with RM codes. From the spectral viewpoint, the encoding operation (8) is regarded as a transform of a "frequency" domain information vector u_1^N to a "time" domain codeword vector x_1^N. The transform is invertible with $G_N^{-1} = G_N$. The decoding operation is regarded as a spectral estimation problem in which one is given a time domain observation y_1^N, which is a noisy version of x_1^N, and asked to estimate u_1^N. To aid the estimation task, one is allowed to freeze a certain number of spectral components of u_1^N. This spectral interpretation of polar coding suggests that it may be possible to treat polar codes and BCH codes in a unified framework. The spectral interpretation also opens the door to the use of various signal processing techniques in polar coding; indeed, in Section VII, we exploit some fast transform techniques in designing encoders for polar codes.

E. Paper Outline

The rest of the paper is organized as follows. Section II explores the recursive properties of the channel splitting operation. In Section III, we focus on how $I(W)$ and $Z(W)$ get transformed through a single step of channel combining and splitting. We extend this to an asymptotic analysis in Section IV and complete the proofs of Theorems 1 and 2. This completes the part of the paper on channel polarization; the rest of the paper is mainly about polar coding. Section V develops an upper bound on the block error probability of polar coding under SC decoding and proves Theorem 3. Section VI considers polar coding for symmetric B-DMCs and proves Theorem 4. Section VII gives an analysis of the encoder mapping G_N, which results in efficient encoder implementations. In Section VIII, we give an implementation of SC decoding with complexity $O(N\log N)$. In Section IX, we discuss the code construction complexity and propose an $O(N\log N)$ statistical algorithm for approximate code construction. In Section X, we explain why RM codes have a poor asymptotic performance under SC decoding. In Section XI, we point out some generalizations of the present work, give some complementary remarks, and state some open problems.

II. Recursive Channel Transformations

We have defined a blockwise channel combining and splitting operation by (4) and (5) which transformed N independent copies of W into $W_N^{(1)},\ldots,W_N^{(N)}$. The goal in this section is to show that this blockwise channel transformation can be broken recursively into single-step channel transformations.

We say that a pair of binary-input channels $W':\mathcal{X}\to\tilde{\mathcal{Y}}$ and $W'':\mathcal{X}\to\tilde{\mathcal{Y}}\times\mathcal{X}$ are obtained by a single-step transformation of two independent copies of a binary-input channel $W:\mathcal{X}\to\mathcal{Y}$ and write

$$(W,W)\mapsto(W',W'')$$

iff there exists a one-to-one mapping $f:\mathcal{Y}^2\to\tilde{\mathcal{Y}}$ such that

$$W'(f(y_1,y_2)|u_1)=\sum_{u_2'}\frac{1}{2}W(y_1|u_1\oplus u_2')W(y_2|u_2') \quad (17)$$

$$W''(f(y_1,y_2),u_1|u_2)=\frac{1}{2}W(y_1|u_1\oplus u_2)W(y_2|u_2) \quad (18)$$

for all $u_1,u_2\in\mathcal{X}$, $y_1,y_2\in\mathcal{Y}$.

According to this, we can write $(W,W)\mapsto(W_2^{(1)},W_2^{(2)})$ for any given B-DMC W because

$$\begin{aligned}W_2^{(1)}\left(y_1^2|u_1\right)&\triangleq\sum_{u_2}\frac{1}{2}W_2\left(y_1^2|u_1^2\right)\\&=\sum_{u_2}\frac{1}{2}W(y_1|u_1\oplus u_2)W(y_2|u_2) \quad (19)\\W_2^{(2)}(y_1^2,u_1|u_2)&\triangleq\frac{1}{2}W_2(y_1^2|u_1^2)\\&=\frac{1}{2}W(y_1|u_1\oplus u_2)W(y_2|u_2) \quad (20)\end{aligned}$$

which are in the form of (17) and (18) by taking f as the identity mapping.

It turns out we can write more generally

$$\left(W_N^{(i)},W_N^{(i)}\right)\mapsto\left(W_{2N}^{(2i-1)},W_{2N}^{(2i)}\right). \quad (21)$$

This follows as a corollary to the following.

Proposition 3: For any $n\geq 0$, $N=2^n$, $1\leq i\leq N$,

$$\begin{aligned}&W_{2N}^{(2i-1)}\left(y_1^{2N},u_1^{2i-2}|u_{2i-1}\right)\\&\quad=\sum_{u_{2i}}\frac{1}{2}W_N^{(i)}\left(y_1^N,u_{1,o}^{2i-2}\oplus u_{1,e}^{2i-2}|u_{2i-1}\oplus u_{2i}\right)\\&\qquad\cdot W_N^{(i)}\left(y_{N+1}^{2N},u_{1,e}^{2i-2}|u_{2i}\right) \quad (22)\end{aligned}$$

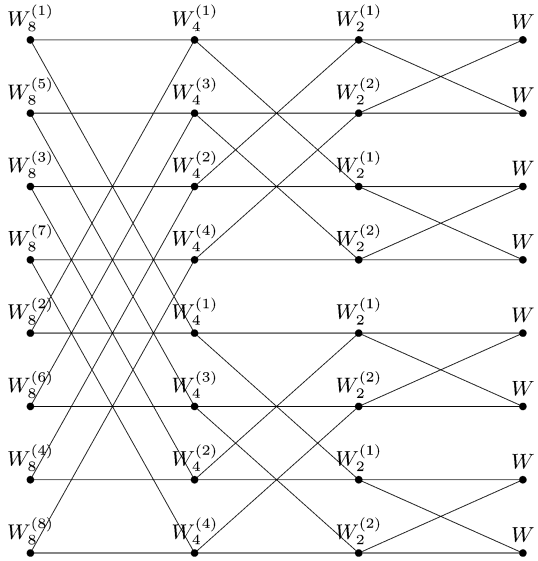

Fig. 5. The channel transformation process with $N = 8$ channels.

and

$$W_{2N}^{(2i)}\left(y_1^{2N}, u_1^{2i-1} | u_{2i}\right) = \frac{1}{2} W_N^{(i)}\left(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2} | u_{2i-1} \oplus u_{2i}\right) \cdot W_N^{(i)}\left(y_{N+1}^{2N}, u_{1,e}^{2i-2} | u_{2i}\right). \quad (23)$$

This proposition is proved in the Appendix. The transform relationship (21) can now be justified by noting that (22) and (23) are identical in form to (17) and (18), respectively, after the following substitutions:

$$W \leftarrow W_N^{(i)}, \quad W' \leftarrow W_{2N}^{(2i-1)},$$
$$W'' \leftarrow W_{2N}^{(2i)}, \quad u_1 \leftarrow u_{2i-1},$$
$$u_2 \leftarrow u_{2i}, \quad y_1 \leftarrow \left(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}\right),$$
$$y_2 \leftarrow \left(y_{N+1}^{2N}, u_{1,e}^{2i-2}\right), \quad f(y_1, y_2) \leftarrow \left(y_1^{2N}, u_1^{2i-2}\right).$$

Thus, we have shown that the blockwise channel transformation from W^N to $(W_N^{(1)}, \ldots, W_N^{(N)})$ breaks at a local level into single-step channel transformations of the form (21). The full set of such transformations form a fabric as shown in Fig. 5 for $N = 8$. Reading from right to left, the figure starts with four copies of the transformation $(W, W) \mapsto (W_2^{(1)}, W_2^{(2)})$ and continues in *butterfly* patterns, each representing a channel transformation of the form $(W_{2^i}^{(j)}, W_{2^i}^{(j)}) \mapsto (W_{2^{i+1}}^{(2j-1)}, W_{2^{i+1}}^{(2j)})$. The two channels at the right endpoints of the butterflies are always identical and independent. At the rightmost level there are eight independent copies of W; at the next level to the left, there are four independent copies of $W_2^{(1)}$ and $W_2^{(2)}$ each; and so on. Each step to the left doubles the number of channel types, but halves the number of independent copies.

III. Transformation of Rate and Reliability

We now investigate how the rate and reliability parameters, $I(W_N^{(i)})$ and $Z(W_N^{(i)})$, change through a local (single-step) transformation (21). By understanding the local behavior, we will be able to reach conclusions about the overall transformation from W^N to $(W_N^{(1)}, \ldots, W_N^{(N)})$. Proofs of the results in this section are given in the Appendix.

A. Local Transformation of Rate and Reliability

Proposition 4: Suppose $(W, W) \mapsto (W', W'')$ for some set of binary-input channels. Then

$$I(W') + I(W'') = 2I(W) \quad (24)$$
$$I(W') \leq I(W'') \quad (25)$$

with equality iff $I(W)$ equals 0 or 1.

The equality (24) indicates that the single-step channel transform preserves the symmetric capacity. The inequality (25) together with (24) implies that the symmetric capacity remains unchanged under a single-step transform, $I(W') = I(W'') = I(W)$, iff W is either a perfect channel or a completely noisy one. If W is neither perfect nor completely noisy, the single-step transform moves the symmetric capacity away from the center in the sense that $I(W') < I(W) < I(W'')$, thus helping polarization.

Proposition 5: Suppose $(W, W) \mapsto (W', W'')$ for some set of binary-input channels. Then

$$Z(W'') = Z(W)^2 \quad (26)$$
$$Z(W') \leq 2Z(W) - Z(W)^2 \quad (27)$$
$$Z(W') \geq Z(W) \geq Z(W''). \quad (28)$$

Equality holds in (27) iff W is a BEC. We have $Z(W') = Z(W'')$ iff $Z(W)$ equals 0 or 1, or equivalently, iff $I(W)$ equals 1 or 0.

This result shows that reliability can only improve under a single-step channel transform in the sense that

$$Z(W') + Z(W'') \leq 2Z(W) \quad (29)$$

with equality iff W is a BEC.

Since the BEC plays a special role with respect to (w.r.t.) extremal behavior of reliability, it deserves special attention.

Proposition 6: Consider the channel transformation $(W, W) \mapsto (W', W'')$. If W is a BEC with some erasure probability ϵ, then the channels W' and W'' are BECs with erasure probabilities $2\epsilon - \epsilon^2$ and ϵ^2, respectively. Conversely, if W' or W'' is a BEC, then W is BEC.

B. Rate and Reliability for $W_N^{(i)}$

We now return to the context at the end of Section II.

Proposition 7: For any B-DMC W, $N = 2^n$, $n \geq 0$, $1 \leq i \leq N$, the transformation $(W_N^{(i)}, W_N^{(i)}) \mapsto (W_{2N}^{(2i-1)}, W_{2N}^{(2i)})$ is rate-preserving and reliability-improving in the sense that

$$I\left(W_{2N}^{(2i-1)}\right) + I\left(W_{2N}^{(2i)}\right) = 2I\left(W_N^{(i)}\right) \quad (30)$$
$$Z\left(W_{2N}^{(2i-1)}\right) + Z\left(W_{2N}^{(2i)}\right) \leq 2Z\left(W_N^{(i)}\right) \quad (31)$$

with equality in (31) iff W is a BEC. Channel splitting moves the rate and reliability away from the center in the sense that

$$I\left(W_{2N}^{(2i-1)}\right) \leq I\left(W_N^{(i)}\right) \leq I\left(W_{2N}^{(2i)}\right) \tag{32}$$

$$Z\left(W_{2N}^{(2i-1)}\right) \geq Z\left(W_N^{(i)}\right) \geq Z\left(W_{2N}^{(2i)}\right) \tag{33}$$

with equality in (32) and (33) iff $I(W)$ equals 0 or 1. The reliability terms further satisfy

$$Z\left(W_{2N}^{(2i-1)}\right) \leq 2Z\left(W_N^{(i)}\right) - Z\left(W_N^{(i)}\right)^2 \tag{34}$$

$$Z\left(W_{2N}^{(2i)}\right) = Z\left(W_N^{(i)}\right)^2 \tag{35}$$

with equality in (34) iff W is a BEC. The cumulative rate and reliability satisfy

$$\sum_{i=1}^{N} I(W_N^{(i)}) = NI(W) \tag{36}$$

$$\sum_{i=1}^{N} Z\left(W_N^{(i)}\right) \leq NZ(W) \tag{37}$$

with equality in (37) iff W is a BEC. □

This result follows from Propositions 4 and 5 as a special case and no separate proof is needed. The cumulative relations (36) and (37) follow by repeated application of (30) and (31), respectively. The conditions for equality in Proposition 4 are stated in terms of W rather than $W_N^{(i)}$; this is possible because i) by Proposition 4, $I(W) \in \{0,1\}$ iff $I(W_N^{(i)}) \in \{0,1\}$; and ii) W is a BEC iff $W_N^{(i)}$ is a BEC, which follows from Proposition 6 by induction.

For the special case that W is a BEC with an erasure probability ϵ, it follows from Propositions 4 and 6 that the parameters $\{Z(W_N^{(i)})\}$ can be computed through the recursion

$$Z\left(W_N^{(2j-1)}\right) = 2Z\left(W_{N/2}^{(j)}\right) - Z\left(W_{N/2}^{(j)}\right)^2$$

$$Z\left(W_N^{(2j)}\right) = Z\left(W_{N/2}^{(j)}\right)^2 \tag{38}$$

with $Z(W_1^{(1)}) = \epsilon$. The parameter $Z(W_N^{(i)})$ equals the erasure probability of the channel $W_N^{(i)}$. The recursive relations (6) follow from (38) by the fact that $I(W_N^{(i)}) = 1 - Z(W_N^{(i)})$ for W a BEC.

IV. Channel Polarization

We prove the main results on channel polarization in this section. The analysis is based on the recursive relationships depicted in Fig. 5; however, it will be more convenient to re-sketch Fig. 5 as a binary tree as shown in Fig. 6. The root node of the tree is associated with the channel W. The root W gives birth to an upper channel $W_2^{(1)}$ and a lower channel $W_2^{(2)}$, which are associated with the two nodes at level 1. The channel $W_2^{(1)}$ in turn gives birth to channels $W_4^{(1)}$ and $W_4^{(2)}$, and so on. The channel

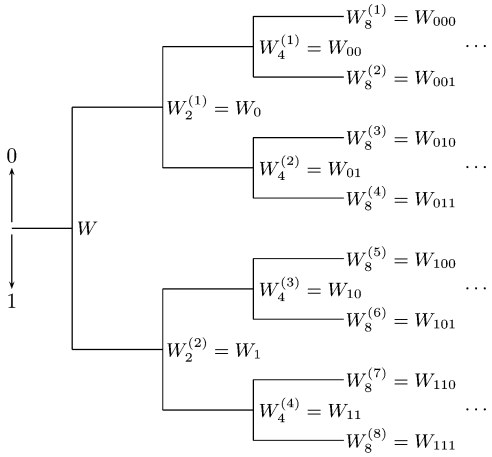

Fig. 6. The tree process for the recursive channel construction.

$W_{2^n}^{(i)}$ is located at level n of the tree at node number i counting from the top.

There is a natural indexing of nodes of the tree in Fig. 6 by bit sequences. The root node is indexed with the null sequence. The upper node at level 1 is indexed with 0 and the lower node with 1. Given a node at level n with index $b_1 b_2 \cdots b_n$, the upper node emanating from it has the label $b_1 b_2 \cdots b_n 0$ and the lower node $b_1 b_2 \cdots b_n 1$. According to this labeling, the channel $W_{2^n}^{(i)}$ is situated at the node $b_1 b_2 \cdots b_n$ with $i = 1 + \sum_{j=1}^{n} b_j 2^{n-j}$. We denote the channel $W_{2^n}^{(i)}$ located at node $b_1 b_2 \cdots b_n$ alternatively as $W_{b_1 \ldots b_n}$.

We define a random tree process, denoted $\{K_n; n \geq 0\}$, in connection with Fig. 6. The process begins at the root of the tree with $K_0 = W$. For any $n \geq 0$, given that $K_n = W_{b_1 \ldots b_n}$, K_{n+1} equals $W_{b_1 \ldots b_n 0}$ or $W_{b_1 \ldots b_n 1}$ with probability $1/2$ each. Thus, the path taken by $\{K_n\}$ through the channel tree may be thought of as being driven by a sequence of independent and identically distributed (i.i.d.) Bernoulli RVs $\{B_n; n = 1, 2, \ldots\}$ where B_n equals 0 or 1 with equal probability. Given that $B_1, \ldots, B_n$ has taken on a sample value $b_1, \ldots, b_n$, the random channel process takes the value $K_n = W_{b_1 \ldots b_n}$. In order to keep track of the rate and reliability parameters of the random sequence of channels K_n, we define the random processes $I_n = I(K_n)$ and $Z_n = Z(K_n)$.

For a more precise formulation of the problem, we consider the probability space $(\Omega, \mathfrak{F}, P)$ where Ω is the space of all binary sequences $(b_1, b_2, \ldots) \in \{0,1\}^\infty$, $\mathfrak{F}$ is the Borel field (BF) generated by the *cylinder sets* $S(b_1, \ldots, b_n) \triangleq \{\omega \in \Omega : \omega_1 = b_1, \ldots, \omega_n = b_n\}$, $n \geq 1$, $b_1, \ldots, b_n \in \{0,1\}$, and P is the probability measure defined on $\mathfrak{F}$ such that $P(S(b_1, \ldots, b_n)) = 1/2^n$. For each $n \geq 1$, we define $\mathfrak{F}_n$ as the BF generated by the cylinder sets $S(b_1, \ldots, b_i)$, $1 \leq i \leq n$, $b_1, \ldots, b_i \in \{0,1\}$. We define $\mathfrak{F}_0$ as the trivial BF consisting of the null set and Ω only. Clearly, $\mathfrak{F}_0 \subset \mathfrak{F}_1 \subset \cdots \subset \mathfrak{F}$.

The random processes described above can now be formally defined as follows. For $\omega = (\omega_1, \omega_2, \ldots) \in \Omega$ and $n \geq 1$, define $B_n(\omega) = \omega_n$, $K_n(\omega) = W_{\omega_1 \ldots \omega_n}$, $I_n(\omega) = I(K_n(\omega))$, and $Z_n(\omega) = Z(K_n(\omega))$. For $n = 0$, define $K_0 = W$, $I_0 = I(W)$,

$Z_0 = Z(W)$. It is clear that, for any fixed $n \geq 0$, the RVs B_n, K_n, I_n, and Z_n are measurable with respect to the BF $\mathfrak{F}_n$.

A. Proof of Theorem 1

We will prove Theorem 1 by considering the stochastic convergence properties of the random sequences $\{I_n\}$ and $\{Z_n\}$.

Proposition 8: The sequence of random variables and Borel fields $\{I_n, \mathfrak{F}_n; n \geq 0\}$ is a martingale, i.e.,

$$\mathfrak{F}_n \subset \mathfrak{F}_{n+1} \text{ and } I_n \text{ is } \mathfrak{F}_n\text{-measurable} \tag{39}$$
$$E[|I_n|] < \infty \tag{40}$$
$$I_n = E[I_{n+1}|\mathfrak{F}_n]. \tag{41}$$

Furthermore, the sequence $\{I_n; n \geq 0\}$ converges almost everywhere (a.e.) to a random variable I_∞ such that $E[I_\infty] = I_0$.

Proof: Condition (39) is true by construction and (40) by the fact that $0 \leq I_n \leq 1$. To prove (41), consider a cylinder set $S(b_1, \ldots, b_n) \in \mathfrak{F}_n$ and use Proposition 7 to write

$$\begin{aligned} E[I_{n+1}|S(b_1, \cdots, b_n)] &= \frac{1}{2}I(W_{b_1 \cdots b_n 0}) + \frac{1}{2}I(W_{b_1 \cdots b_n 1}) \\ &= I(W_{b_1 \cdots b_n}). \end{aligned} \tag{42}$$

Since $I(W_{b_1 \cdots b_n})$ is the value of I_n on $S(b_1, \ldots, b_n)$, (41) follows. This completes the proof that $\{I_n, \mathfrak{F}_n\}$ is a martingale. Since $\{I_n, \mathfrak{F}_n\}$ is a uniformly integrable martingale, by general convergence results about such martingales (see, e.g., [9, Theorem 9.4.6]), the claim about I_∞ follows. □

It should not be surprising that the limit RV I_∞ takes values a.e. in $\{0, 1\}$, which is the set of fixed points of $I(W)$ under the transformation $(W, W) \mapsto (W_2^{(1)}, W_2^{(2)})$, as determined by the condition for equality in (25). For a rigorous proof of this statement, we take an indirect approach and bring the process $\{Z_n; n \geq 0\}$ also into the picture.

Proposition 9: The sequence of random variables and Borel fields $\{Z_n, \mathfrak{F}_n; n \geq 0\}$ is a supermartingale, i.e.,

$$\mathfrak{F}_n \subset \mathfrak{F}_{n+1} \text{ and } Z_n \text{ is } \mathfrak{F}_n\text{-measurable} \tag{43}$$
$$E[|Z_n|] < \infty \tag{44}$$
$$Z_n \geq E[Z_{n+1}|\mathfrak{F}_n]. \tag{45}$$

Furthermore, the sequence $\{Z_n; n \geq 0\}$ converges a.e. to a random variable Z_∞ which takes values a.e. in $\{0, 1\}$.

Proof: Conditions (43) and (44) are clearly satisfied. To verify (45), consider a cylinder set $S(b_1, \ldots, b_n) \in \mathfrak{F}_n$ and use Proposition 7 to write

$$\begin{aligned} E[Z_{n+1}|S(b_1, \ldots, b_n)] &= \frac{1}{2}Z(W_{b_1 \cdots b_n 0}) + \frac{1}{2}Z(W_{b_1 \cdots b_n 1}) \\ &\leq Z(W_{b_1 \cdots b_n}). \end{aligned}$$

Since $Z(W_{b_1 \cdots b_n})$ is the value of Z_n on $S(b_1, \ldots, b_n)$, (45) follows. This completes the proof that $\{Z_n, \mathfrak{F}_n\}$ is a supermartingale. For the second claim, observe that the supermartingale $\{Z_n, \mathfrak{F}_n\}$ is uniformly integrable; hence, it converges a.e. and in $\mathcal{L}^1$ to an RV Z_∞ such that $E[|Z_n - Z_\infty|] \to 0$ (see, e.g., [9, Theorem 9.4.5]). It follows that $E[|Z_{n+1} - Z_n|] \to 0$. But, by Proposition 7, $Z_{n+1} = Z_n^2$ with probability $1/2$; hence, $E[|Z_{n+1} - Z_n|] \geq (1/2)E[Z_n(1 - Z_n)] \geq 0$. Thus, $E[Z_n(1 - Z_n)] \to 0$, which implies $E[Z_\infty(1 - Z_\infty)] = 0$. This, in turn, means that Z_∞ equals 0 or 1 a.e. □

Proposition 10: The limit RV I_∞ takes values a.e. in the set $\{0, 1\}$: $P(I_\infty = 1) = I_0$ and $P(I_\infty = 0) = 1 - I_0$.

Proof: The fact that Z_∞ equals 0 or 1 a.e., combined with Proposition 1, implies that $I_\infty = 1 - Z_\infty$ a.e. Since $E[I_\infty] = I_0$, the rest of the claim follows. □

As a corollary to Proposition 10, we can conclude that, as N tends to infinity, the symmetric capacity terms $\{I(W_N^{(i)}) : 1 \leq i \leq N\}$ cluster around 0 and 1, except for a vanishing fraction. This completes the proof of Theorem 1.

It is interesting that the above discussion gives a new interpretation to $I_0 = I(W)$ as the probability that the random process $\{Z_n; n \geq 0\}$ converges to zero. We may use this to strengthen the lower bound in (1). (This stronger form is given as a side result and will not be used in the sequel.)

Proposition 11: For any B-DMC W, we have $I(W) + Z(W) \geq 1$ with equality iff W is a BEC.

This result can be interpreted as saying that, among all B-DMCs W, the BEC presents the most favorable rate–reliability tradeoff: it minimizes $Z(W)$ (maximizes reliability) among all channels with a given symmetric capacity $I(W)$; equivalently, it minimizes $I(W)$ required to achieve a given level of reliability $Z(W)$.

Proof of Proposition 11: Consider two channels W and W' with $Z(W) = Z(W') \triangleq z_0$. Suppose that W' is a BEC. Then, W' has erasure probability z_0 and $I(W') = 1 - z_0$. Consider the random processes $\{Z_n\}$ and $\{Z_n'\}$ corresponding to W and W', respectively. By the condition for equality in (34), the process $\{Z_n\}$ is stochastically dominated by $\{Z_n'\}$ in the sense that $P(Z_n \leq z) \geq P(Z_n' \leq z)$ for all $n \geq 1$, $0 \leq z \leq 1$. Thus, the probability of $\{Z_n\}$ converging to zero is lower-bounded by the probability that $\{Z_n'\}$ converges to zero, i.e., $I(W) \geq I(W')$. This implies $I(W) + Z(W) \geq 1$. □

B. Proof of Theorem 2

We will now prove Theorem 2, which strengthens the above polarization results by specifying a rate of polarization. Consider the probability space $(\Omega, \mathfrak{F}, P)$. For $\omega \in \Omega$, $i \geq 0$, by Proposition 7, we have $Z_{i+1}(\omega) = Z_i^2(\omega)$ if $B_{i+1}(\omega) = 1$ and $Z_{i+1}(\omega) \leq 2Z_i(\omega) - Z_i(\omega)^2 \leq 2Z_i(\omega)$ if $B_{i+1}(\omega) = 0$. For $\zeta \geq 0$ and $m \geq 0$, define

$$\mathcal{T}_m(\zeta) \triangleq \{\omega \in \Omega : Z_i(\omega) \leq \zeta \text{ for all } i \geq m\}.$$

For $\omega \in \mathcal{T}_m(\zeta)$ and $i \geq m$, we have

$$\frac{Z_{i+1}(\omega)}{Z_i(\omega)} \leq \begin{cases} 2, & \text{if } B_{i+1}(\omega) = 0 \\ \zeta, & \text{if } B_{i+1}(\omega) = 1 \end{cases}$$

which implies

$$Z_n(\omega) \leq \zeta \cdot 2^{n-m} \cdot \prod_{i=m+1}^{n} (\zeta/2)^{B_i(\omega)}, \quad \omega \in T_m(\zeta), n > m.$$

For $n > m \geq 0$ and $0 < \eta < 1/2$, define

$$\mathcal{U}_{m,n}(\eta) \triangleq \{\omega \in \Omega : \sum_{i=m+1}^{n} B_i(\omega) > (1/2 - \eta)(n-m)\}.$$

Then, we have

$$Z_n(\omega) \leq \zeta \cdot \left[2^{\frac{1}{2}+\eta}\zeta^{\frac{1}{2}-\eta}\right]^{n-m}, \quad \omega \in \mathcal{T}_m(\zeta) \cap \mathcal{U}_{m,n}(\eta)$$

from which, by putting $\zeta_0 \triangleq 2^{-4}$ and $\eta_0 \triangleq 1/20$, we obtain

$$Z_n(\omega) \leq 2^{-4-5(n-m)/4}, \quad \omega \in \mathcal{T}_m(\zeta_0) \cap \mathcal{U}_{m,n}(\eta_0). \quad (46)$$

Now, we show that (46) occurs with sufficiently high probability. First, we use the following result, which is proved in the Appendix.

Lemma 1: For any fixed $\zeta > 0$, $\delta > 0$, there exists a finite integer $m_0(\zeta, \delta)$ such that

$$P[\mathcal{T}_{m_0}(\zeta)] \geq I_0 - \delta/2.$$

Second, we use Chernoff's bound [10, p. 531] to write

$$P[\mathcal{U}_{m,n}(\eta)] \geq 1 - 2^{-(n-m)[1-H(1/2-\eta)]} \quad (47)$$

where H is the binary entropy function. Define $n_0(m, \eta, \delta)$ as the smallest n such that the RHS of (47) is greater than or equal to $1 - \delta/2$; it is clear that $n_0(m, \eta, \delta)$ is finite for any $m \geq 0$, $0 < \eta < 1/2$, and $\delta > 0$. Now, with $m_1 = m_1(\delta) \triangleq m_0(\zeta_0, \delta)$ and $n_1 = n_1(\delta) \triangleq n_0(m_1, \eta_0, \delta)$, we obtain the desired bound

$$P[\mathcal{T}_{m_1}(\zeta_0) \cap \mathcal{U}_{m_1,n}(\eta_0)] \geq I_0 - \delta, \quad n \geq n_1.$$

Finally, we tie the above analysis to the claim of Theorem 2. Define $c \triangleq 2^{-4+5m_1/4}$ and

$$\mathcal{V}_n \triangleq \{\omega \in \Omega : Z_n(\omega) \leq c\,2^{-5n/4}\}, \quad n \geq 0,$$

and note that

$$\mathcal{T}_{m_1}(\zeta_0) \cap \mathcal{U}_{m_1,n}(\eta_0) \subset \mathcal{V}_n, \quad n \geq n_1.$$

So, $P(\mathcal{V}_n) \geq I_0 - \delta$ for $n \geq n_1$. On the other hand

$$\begin{aligned} P(\mathcal{V}_n) &= \sum_{\omega_1^n \in \mathcal{X}^n} \frac{1}{2^n} 1\left\{Z\left(W_{\omega_1^n}\right) \leq c\,2^{-5n/4}\right\} \\ &= \frac{1}{N}|\mathcal{A}_N| \end{aligned}$$

where $\mathcal{A}_N \triangleq \{i \in \{1, \ldots, N\} : Z(W_N^{(i)}) \leq cN^{-5/4}\}$ with $N = 2^n$. We conclude that $|\mathcal{A}_N| \geq N(I_0 - \delta)$ for $n \geq n_1(\delta)$. This completes the proof of Theorem 2.

Given Theorem 2, it is an easy exercise to show that polar coding can achieve rates approaching $I(W)$, as we will show in the next section. It is clear from the above proof that Theorem 2 gives only an *ad hoc* result on the asymptotic rate of channel polarization; this result is sufficient for proving a capacity theorem for polar coding; however, finding the exact asymptotic rate of polarization remains an important goal for future research.[2]

[2]A recent result in this direction is discussed in Section XI-A.

V. Performance of Polar Coding

We show in this section that polar coding can achieve the symmetric capacity $I(W)$ of any B-DMC W. The main technical task will be to prove Proposition 2. We will carry out the analysis over the class of G_N-coset codes before specializing the discussion to polar codes. Recall that individual G_N-coset codes are identified by a parameter vector $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$. In the analysis, we will fix the parameters $(N, K, \mathcal{A})$ while keeping $u_{\mathcal{A}^c}$ free to take any value over $\mathcal{X}^{N-K}$. In other words, the analysis will be over the ensemble of 2^{N-K} G_N-coset codes with a fixed $(N, K, \mathcal{A})$. The decoder in the system will be the SC decoder described in Section I-C.2.

A. A Probabilistic Setting for the Analysis

Let $(\mathcal{X}^N \times \mathcal{Y}^N, P)$ be a probability space with the probability assignment

$$P\left(\{(u_1^N, y_1^N)\}\right) \triangleq 2^{-N} W_N\left(y_1^N | u_1^N\right) \quad (48)$$

for all $(u_1^N, y_1^N) \in \mathcal{X}^N \times \mathcal{Y}^N$. On this probability space, we define an ensemble of random vectors $(U_1^N, X_1^N, Y_1^N, \hat{U}_1^N)$ that represent, respectively, the input to the synthetic channel W_N, the input to the product–form channel W^N, the output of W^N (and also of W_N), and the decisions by the decoder. For each sample point $(u_1^N, y_1^N) \in \mathcal{X}^N \times \mathcal{Y}^N$, the first three vectors take on the values $U_1^N(u_1^N, y_1^N) = u_1^N$, $X_1^N(u_1^N, y_1^N) = u_1^N G_N$, and $Y_1^N(u_1^N, y_1^N) = y_1^N$, while the decoder output takes on the value $\hat{U}_1^N(u_1^N, y_1^N)$ whose coordinates are defined recursively as

$$\hat{U}_i\left(u_1^N, y_1^N\right) = \begin{cases} u_i, & i \in \mathcal{A}^c \\ h_i\left(y_1^N, \hat{U}_1^{i-1}\left(u_1^N, y_1^N\right)\right), & i \in \mathcal{A} \end{cases} \quad (49)$$

for $i = 1, \ldots, N$.

A realization $u_1^N \in \mathcal{X}^N$ for the input random vector U_1^N corresponds to sending the data vector $u_{\mathcal{A}}$ together with the frozen vector $u_{\mathcal{A}^c}$. As random vectors, the data part $U_{\mathcal{A}}$ and the frozen part $U_{\mathcal{A}^c}$ are uniformly distributed over their respective ranges and statistically independent. By treating $U_{\mathcal{A}^c}$ as a random vector over $\mathcal{X}^{N-K}$, we obtain a convenient method for analyzing code performance averaged over all codes in the ensemble $(N, K, \mathcal{A})$.

The main event of interest in the following analysis is the block error event under SC decoding, defined as

$$\mathcal{E} \triangleq \left\{\left(u_1^N, y_1^N\right) \in \mathcal{X}^N \times \mathcal{Y}^N : \hat{U}_{\mathcal{A}}\left(u_1^N, y_1^N\right) \neq u_{\mathcal{A}}\right\}. \quad (50)$$

Since the decoder never makes an error on the frozen part of U_1^N, i.e., $\hat{U}_{\mathcal{A}^c}$ equals $U_{\mathcal{A}^c}$ with probability one, that part has been excluded from the definition of the block error event.

The probability of error terms $P_e(N, K, \mathcal{A})$ and $P_e(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ that were defined in Section I-C.3 can be expressed in this probability space as

$$\begin{aligned} P_e(N, K, \mathcal{A}) &= P(\mathcal{E}) \\ P_e(N, K, \mathcal{A}, u_{\mathcal{A}^c}) &= P(\mathcal{E} | \{U_{\mathcal{A}^c} = u_{\mathcal{A}^c}\}) \end{aligned} \quad (51)$$

where $\{U_{\mathcal{A}^c} = u_{\mathcal{A}^c}\}$ denotes the event $\{(\tilde{u}_1^N, y_1^N) \in \mathcal{X}^N \times \mathcal{Y}^N : \tilde{u}_{\mathcal{A}^c} = u_{\mathcal{A}^c}\}$.

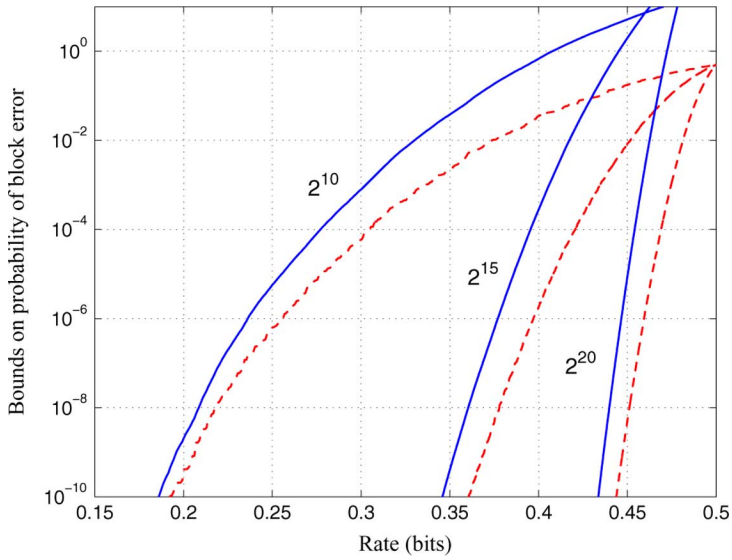

Fig. 7. Rate versus reliability for polar coding and SC decoding at block lengths $2^{10}, 2^{15}$, and 2^{20} on a BEC with erasure probability $1/2$.

B. Proof of Proposition 2

We may express the block error event as $\mathcal{E} = \cup_{i\in\mathcal{A}}\mathcal{B}_i$ where

$$\mathcal{B}_i \triangleq \Big\{ \left(u_1^N, y_1^N\right) \in \mathcal{X}^N \times \mathcal{Y}^N : u_1^{i-1} = \hat{U}_1^{i-1}\left(u_1^N, y_1^N\right), \; u_i \neq \hat{U}_i\left(u_1^N, y_1^N\right) \Big\} \quad (52)$$

is the event that the first decision error in SC decoding occurs at stage i. We notice that

$$\begin{aligned}\mathcal{B}_i &= \Big\{ \left(u_1^N, y_1^N\right) \in \mathcal{X}^N \times \mathcal{Y}^N : u_1^{i-1} = \hat{U}_1^{i-1}\left(u_1^N, y_1^N\right), \\ &\qquad u_i \neq h_i(y_1^N, \hat{U}_1^{i-1}\left(u_1^N, y_1^N\right))\Big\} \\ &= \Big\{ \left(u_1^N, y_1^N\right) \in \mathcal{X}^N \times \mathcal{Y}^N : u_1^{i-1} = \hat{U}_1^{i-1}\left(u_1^N, y_1^N\right), \\ &\qquad u_i \neq h_i\left(y_1^N, u_1^{i-1}\right) \Big\} \\ &\subset \Big\{ \left(u_1^N, y_1^N\right) \in \mathcal{X}^N \times \mathcal{Y}^N : u_i \neq h_i\left(y_1^N, u_1^{i-1}\right) \Big\} \subset \mathcal{E}_i\end{aligned}$$

where

$$\mathcal{E}_i \triangleq \Big\{ \left(u_1^N, y_1^N\right) \in \mathcal{X}^N \times \mathcal{Y}^N : W_N^{(i-1)}\left(y_1^N, u_1^{i-1} | u_i\right) \leq W_N^{(i-1)}\left(y_1^N, u_1^{i-1} | u_i \oplus 1\right) \Big\}. \quad (53)$$

Thus, we have

$$\mathcal{E} \subset \bigcup_{i\in\mathcal{A}} \mathcal{E}_i, \quad P(\mathcal{E}) \leq \sum_{i\in\mathcal{A}} P(\mathcal{E}_i).$$

For an upper bound on $P(\mathcal{E}_i)$, note that

$$\begin{aligned}P(\mathcal{E}_i) &= \sum_{u_1^N, y_1^N} \frac{1}{2^N} W_N(y_1^N | u_1^N) 1_{\mathcal{E}_i}\left(u_1^N, y_1^N\right) \\ &\leq \sum_{u_1^N, y_1^N} \frac{1}{2^N} W_N\left(y_1^N | u_1^N\right) \sqrt{\frac{W_N^{(i)}\left(y_1^N, u_1^{i-1} | u_i \oplus 1\right)}{W_N^{(i)}\left(y_1^N, u_1^{i-1} | u_i\right)}} \\ &= Z\left(W_N^{(i)}\right). \end{aligned} \quad (54)$$

We conclude that

$$P(\mathcal{E}) \leq \sum_{i\in\mathcal{A}} Z\left(W_N^{(i)}\right)$$

which is equivalent to (13). This completes the proof of Proposition 2. The main coding theorem of the paper now follows readily.

C. Proof of Theorem 3

By Theorem 2, for any given rate $R < I(W)$, there exists a sequence of information sets $\mathcal{A}_N$ with size $|\mathcal{A}_N| \geq NR$ such that

$$\sum_{i\in\mathcal{A}_N} Z\left(W_N^{(i)}\right) \leq N \max_{i\in\mathcal{A}_N} \left\{ Z\left(W_N^{(i)}\right) \right\} = O\left(N^{-\frac{1}{4}}\right). \quad (55)$$

In particular, the bound (55) holds if $\mathcal{A}_N$ is chosen in accordance with the polar coding rule because by definition this rule minimizes the sum in (55). Combining this fact about the polar coding rule with Proposition 2, Theorem 3 follows.

D. A Numerical Example

Although we have established that polar codes achieve the symmetric capacity, the proofs have been of an asymptotic nature and the exact asymptotic rate of polarization has not been found. It is of interest to understand how quickly the polarization effect takes hold and what performance can be expected of polar codes under SC decoding in the nonasymptotic regime. To investigate these, we give here a numerical study.

Let W be a BEC with erasure probability $1/2$. Fig. 7 shows the rate versus reliability tradeoff for W using polar codes with block lengths $N \in \{2^{10}, 2^{15}, 2^{20}\}$. This figure is obtained by using codes whose information sets are of the form $\mathcal{A}(\eta) \triangleq \{i \in \{1, \ldots, N\} : Z(W_N^{(i)}) < \eta\}$, where $0 \leq \eta \leq 1$ is a variable threshold parameter. There are two sets of three curves in the plot. The solid lines are plots of $R(\eta) \triangleq |\mathcal{A}(\eta)|/N$ versus $B(\eta) \triangleq \sum_{i\in\mathcal{A}(\eta)} Z(W_N^{(i)})$. The dashed lines are plots

of $R(\eta)$ versus $L(\eta) \triangleq \max_{i\in\mathcal{A}(\eta)}\{Z(W_N^{(i)})\}$. The parameter η is varied over a subset of $[0,1]$ to obtain the curves.

The parameter $R(\eta)$ corresponds to the code rate. The significance of $B(\eta)$ is also clear: it is an upper bound on $P_e(\eta)$, the probability of block error for polar coding at rate $R(\eta)$ under SC decoding. The parameter $L(\eta)$ is intended to serve as a lower bound to $P_e(\eta)$.

This example provides empirical evidence that polar coding achieves channel capacity as the block length is increased—a fact already established theoretically. More significantly, the example also shows that the rate of polarization is too slow to make near-capacity polar coding under SC decoding feasible in practice.

VI. Symmetric Channels

The main goal of this section is to prove Theorem 4, which is a strengthened version of Theorem 3 for symmetric channels.

A. Symmetry Under Channel Combining and Splitting

Let $W : \mathcal{X} \to \mathcal{Y}$ be a symmetric B-DMC with $\mathcal{X} = \{0,1\}$ and $\mathcal{Y}$ arbitrary. By definition, there exists a permutation π_1 on $\mathcal{Y}$ such that i) $\pi_1^{-1} = \pi_1$ and ii) $W(y|1) = W(\pi_1(y)|0)$ for all $y \in \mathcal{Y}$. Let π_0 be the identity permutation on $\mathcal{Y}$. Clearly, the permutations (π_0, π_1) form an Abelian group under function composition. For a compact notation, we will write $x \cdot y$ to denote $\pi_x(y)$, for $x \in \mathcal{X}$, $y \in \mathcal{Y}$.

Observe that $W(y|x \oplus a) = W(a \cdot y|x)$ for all $a, x \in \mathcal{X}$, $y \in \mathcal{Y}$. This can be verified by exhaustive study of possible cases or by noting that $W(y|x \oplus a) = W((x \oplus a) \cdot y|0) = W(x \cdot (a \cdot y)|0) = W(a \cdot y|x)$. Also, observe that $W(y|x \oplus a) = W(x \cdot y|a)$ as $\oplus$ is a commutative operation on $\mathcal{X}$.

For $x_1^N \in \mathcal{X}^N$, $y_1^N \in \mathcal{Y}^N$, let

$$x_1^N \cdot y_1^N \triangleq (x_1 \cdot y_1, \ldots, x_N \cdot y_N). \tag{56}$$

This associates to each element of $\mathcal{X}^N$ a permutation on $\mathcal{Y}^N$.

Proposition 12: If a B-DMC W is symmetric, then W^N is also symmetric in the sense that

$$W^N\left(y_1^N | x_1^N \oplus a_1^N\right) = W^N\left(x_1^N \cdot y_1^N | a_1^N\right) \tag{57}$$

for all $x_1^N, a_1^N \in \mathcal{X}^N$, $y_1^N \in \mathcal{Y}^N$.

The proof is immediate and omitted.

Proposition 13: If a B-DMC W is symmetric, then the channels W_N and $W_N^{(i)}$ are also symmetric in the sense that

$$\begin{aligned} &W_N\left(y_1^N | u_1^N\right) \\ &\quad = W_N\left(a_1^N G_N \cdot y_1^N | u_1^N \oplus a_1^N\right), \end{aligned} \tag{58}$$

$$\begin{aligned} &W_N^{(i)}\left(y_1^N, u_1^{i-1} | u_i\right) \\ &\quad = W_N^{(i)}\left(a_1^N G_N \cdot y_1^N, u_1^{i-1} \oplus a_1^{i-1} | u_i \oplus a_i\right) \end{aligned} \tag{59}$$

for all $u_1^N, a_1^N \in \mathcal{X}^N$, $y_1^N \in \mathcal{Y}^N$, $N = 2^n$, $n \geq 0$, $1 \leq i \leq N$.

Proof: Let $x_1^N = u_1^N G_N$ and observe that $W_N(y_1^N|u_1^N) = \prod_{i=1}^N W(y_i|x_i) = \prod_{i=1}^N W(x_i \cdot y_i|0) = W_N(x_1^N \cdot y_1^N|0_1^N)$. Now, let $b_1^N = a_1^N G_N$, and use the same reasoning to see that $W_N(b_1^N \cdot y_1^N | u_1^N \oplus a_1^N) = W_N((x_1^N \oplus b_1^N) \cdot (b_1^N \cdot y_1^N)|0_1^N) = W_N(x_1^N \cdot y_1^N|0_1^N)$. This proves the first claim. To prove the second claim, we use the first result

$$\begin{aligned} W_N^{(i)}\left(y_1^N, u_1^{i-1}|u_i\right) &= \sum_{u_{i+1}^N} \frac{1}{2^{N-1}} W_N\left(y_1^N|u_1^N\right) \\ &= \sum_{u_{i+1}^N} \frac{1}{2^{N-1}} W_N\left(a_1^N G_N \cdot y_1^N | u_1^N \oplus a_1^N\right) \\ &= W_N\left(a_1^N G_N \cdot y_1^N, u_1^{i-1} \oplus a_1^{i-1} | u_i \oplus a_i\right) \end{aligned}$$

where we used the fact that the sum over $u_{i+1}^N \in \mathcal{X}^{N-i}$ can be replaced with a sum over $u_{i+1}^N \oplus a_{i+1}^N$ for any fixed a_1^N since $\{u_{i+1}^N \oplus a_{i+1}^N : u_{i+1}^N \in \mathcal{X}^{N-i}\} = X^{N-i}$. □

B. Proof of Theorem 4

We return to the analysis in Section V and consider a code ensemble $(N, K, \mathcal{A})$ under SC decoding, only this time assuming that W is a symmetric channel. We first show that the error events $\{\mathcal{E}_i\}$ defined by (53) have a symmetry property.

Proposition 14: For a symmetric B-DMC W, the event $\mathcal{E}_i$ has the property that

$$\left(u_1^N, y_1^N\right) \in \mathcal{E}_i \quad \text{iff} \quad \left(a_1^N \oplus u_1^N, a_1^N G_N \cdot y_1^N\right) \in \mathcal{E}_i \tag{60}$$

for each $1 \leq i \leq N$, $(u_1^N, y_1^N) \in \mathcal{X}^N \times \mathcal{Y}^N$, $a_1^N \in \mathcal{X}^N$.

Proof: This follows directly from the definition of $\mathcal{E}_i$ by using the symmetry property (59) of the channel $W_N^{(i)}$. □

Now, consider the transmission of a particular source vector $u_{\mathcal{A}}$ and a frozen vector $u_{\mathcal{A}^c}$, jointly forming an input vector u_1^N for the channel W_N. This event is denoted below as $\{U_1^N = u_1^N\}$ instead of the more formal $\{u_1^N\} \times \mathcal{Y}^N$.

Corollary 1: For a symmetric B-DMC W, for each $1 \leq i \leq N$ and $u_1^N \in \mathcal{X}^N$, the events $\mathcal{E}_i$ and $\{U_1^N = u_1^N\}$ are independent; hence, $P(\mathcal{E}_i) = P(\mathcal{E}_i|\{U_1^N = u_1^N\})$.

Proof: For $(u_1^N, y_1^N) \in \mathcal{X}^N \times \mathcal{Y}^N$ and $x_1^N = u_1^N G_N$, we have

$$\begin{aligned} &P\left(\mathcal{E}_i | \left\{U_1^N = u_1^N\right\}\right) \\ &\quad = \sum_{y_1^N} W_N\left(y_1^N | u_1^N\right) 1_{\mathcal{E}_i}\left(u_1^N, y_1^N\right) \\ &\quad = \sum_{y_1^N} W_N\left(x_1^N \cdot y_1^N | 0_1^N\right) 1_{\mathcal{E}_i}\left(0_1^N, x_1^N \cdot y_1^N\right) \qquad (61) \\ &\quad = P\left(\mathcal{E}_i | \left\{U_1^N = 0_1^N\right\}\right). \qquad (62) \end{aligned}$$

Equality follows in (61) from (58) and (60) by taking $a_1^N = u_1^N$, and in (62) from the fact that $\{x_1^N \cdot y_1^N : y_1^N \in \mathcal{Y}^N\} = \mathcal{Y}^N$ for any fixed $x_1^N \in \mathcal{X}^N$. The rest of the proof is immediate. □

Now, by (54), we have, for all $u_1^N \in \mathcal{X}^N$

$$P\left(\mathcal{E}_i | \left\{U_1^N = u_1^N\right\}\right) \leq Z\left(W_N^{(i)}\right) \tag{63}$$

and, since $\mathcal{E} \subset \cup_{i\in\mathcal{A}} \mathcal{E}_i$, we obtain

$$P\left(\mathcal{E} | \left\{U_1^N = u_1^N\right\}\right) \leq \sum_{i\in\mathcal{A}} Z\left(W_N^{(i)}\right). \tag{64}$$

This implies that, for every symmetric B-DMC W and every $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ code

$$P_e(N, K, \mathcal{A}, u_{\mathcal{A}^c}) = \sum_{u_{\mathcal{A}} \in \mathcal{X}^K} \frac{1}{2^K} P\left(\mathcal{E} \middle| \{U_1^N = u_1^N\}\right) \leq \sum_{i \in \mathcal{A}} Z\left(W_N^{(i)}\right). \quad (65)$$

This bound on $P_e(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ is independent of the frozen vector $u_{\mathcal{A}^c}$. Theorem 4 is now obtained by combining Theorem 2 with Proposition 2, as in the proof of Theorem 3.

Note that although we have given a bound on $P(\mathcal{E}|\{U_1^N = u_1^N\})$ that is independent of u_1^N, we stopped short of claiming that the error event $\mathcal{E}$ is independent of U_1^N because our decision functions $\{h_i\}$ break ties always in favor of $\hat{u}_i = 0$. If this bias were removed by randomization, then $\mathcal{E}$ would become independent of U_1^N.

C. Further Symmetries of the Channel $W_N^{(i)}$

We may use the degrees of freedom in the choice of a_1^N in (59) to explore the symmetries inherent in the channel $W_N^{(i)}$. For a given (y_1^N, u_1^i), we may select a_1^N with $a_1^i = u_1^i$ to obtain

$$W_N^{(i)}\left(y_1^N, u_1^{i-1} | u_i\right) = W_N^{(i)}\left(a_1^N G_N \cdot y_1^N, 0_1^{i-1} | 0\right). \quad (66)$$

So, if we were to prepare a lookup table for the transition probabilities $\{W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) : y_1^N \in \mathcal{Y}^N, u_1^i \in \mathcal{X}^i\}$, it would suffice to store only the subset of probabilities $\{W_N^{(i)}(y_1^N, 0_1^{i-1}|0) : y_1^N \in \mathcal{Y}^N\}$.

The size of the lookup table can be reduced further by using the remaining degrees of freedom in the choice of a_{i+1}^N. Let $\mathcal{X}_{i+1}^N \triangleq \{a_1^N \in \mathcal{X}^N : a_1^i = 0_1^i\}$, $1 \leq i \leq N$. Then, for any $1 \leq i \leq N$, $a_1^N \in \mathcal{X}_{i+1}^N$, and $y_1^N \in \mathcal{Y}^N$, we have

$$W_N^{(i)}\left(y_1^N, 0^{i-1} | 0\right) = W_N^{(i)}\left(a_1^N G_N \cdot y_1^N, 0_1^{i-1} | 0\right) \quad (67)$$

which follows from (66) by taking $u_1^i = 0_1^i$ on the left hand side.

To explore this symmetry further, let $\mathcal{X}_{i+1}^N \cdot y_1^N \triangleq \{a_1^N G_N \cdot y_1^N : a_1^N \in \mathcal{X}_{i+1}^N\}$. The set $\mathcal{X}_{i+1}^N \cdot y_1^N$ is the *orbit* of y_1^N under the *action group* $\mathcal{X}_{i+1}^N$. The orbits $\mathcal{X}_{i+1}^N \cdot y_1^N$ over variation of y_1^N partition the space $\mathcal{Y}^N$ into equivalence classes. Let $\mathcal{Y}_{i+1}^N$ be a set formed by taking one representative from each equivalence class. The output alphabet of the channel $W_N^{(i)}$ can be represented effectively by the set $\mathcal{Y}_{i+1}^N$.

For example, suppose W is a BSC with $\mathcal{Y} = \{0, 1\}$. Each orbit $\mathcal{X}_{i+1}^N \cdot y_1^N$ has 2^{N-i} elements and there are 2^i orbits. In particular, the channel $W_N^{(1)}$ has effectively two outputs, and being symmetric, it has to be a BSC. This is a great simplification since $W_N^{(1)}$ has an apparent output alphabet size of 2^N. Likewise, while $W_N^{(i)}$ has an apparent output alphabet size of 2^{N+i-1}, due to symmetry, the size shrinks to 2^i.

Further output alphabet size reductions may be possible by exploiting other properties specific to certain B-DMCs. For example, if W is a BEC, the channels $\{W_N^{(i)}\}$ are known to be BECs, each with an effective output alphabet size of three.

The symmetry properties of $\{W_N^{(i)}\}$ help simplify the computation of the channel parameters.

Proposition 15: For any symmetric B-DMC W, the parameters $\{Z(W_N^{(i)})\}$ given by (7) can be calculated by the simplified formula

$$Z\left(W_N^{(i)}\right) = 2^{i-1} \sum_{y_1^N \in \mathcal{Y}_{i+1}^N} \left|\mathcal{X}_{i+1}^N \cdot y_1^N\right| \cdot \sqrt{W_N^{(i)}\left(y_1^N, 0_1^{i-1} | 0\right) W_N^{(i)}\left(y_1^N, 0_1^{i-1} | 1\right)}.$$

We omit the proof of this result.

For the important example of a BSC, this formula becomes

$$Z\left(W_N^{(i)}\right) = 2^{N-1} \sum_{y_1^N \in \mathcal{Y}_{i+1}^N} \sqrt{W_N^{(i)}\left(y_1^N, 0_1^{i-1} | 0\right) W_N^{(i)}\left(y_1^N, 0_1^{i-1} | 1\right)}.$$

This sum for $Z(W_N^{(i)})$ has 2^i terms, as compared to 2^{N+i-1} terms in (7).

VII. Encoding

In this section, we will consider the encoding of polar codes and prove the part of Theorem 5 about encoding complexity. We begin by giving explicit algebraic expressions for G_N, the generator matrix for polar coding, which so far has been defined only in a schematic form by Fig. 3. The algebraic forms of G_N naturally point at efficient implementations of the encoding operation $x_1^N = u_1^N G_N$. In analyzing the encoding operation G_N, we exploit its relation to fast transform methods in signal processing; in particular, we use the bit-indexing idea of [11] to interpret the various permutation operations that are part of G_N.

A. Formulas for G_N

In the following, assume $N = 2^n$ for some $n \geq 0$. Let I_k denote the k-dimensional identity matrix for any $k \geq 1$. We begin by translating the recursive definition of G_N as given by Fig. 3 into an algebraic form

$$G_N = (I_{N/2} \otimes F) R_N (I_2 \otimes G_{N/2}), \quad \text{for } N \geq 2$$

with $G_1 = I_1$.

Either by verifying algebraically that $(I_{N/2} \otimes F) R_N = R_N (F \otimes I_{N/2})$ or by observing that channel combining operation in Fig. 3 can be redrawn equivalently as in Fig. 8, we obtain a second recursive formula

$$G_N = R_N (F \otimes I_{N/2})(I_2 \otimes G_{N/2}) = R_N (F \otimes G_{N/2}) \quad (68)$$

valid for $N \geq 2$. This form appears more suitable to derive a recursive relationship. We substitute $G_{N/2} = R_{N/2}(F \otimes G_{N/4})$ back into (68) to obtain

$$G_N = R_N (F \otimes (R_{N/2}(F \otimes G_{N/4}))) = R_N (I_2 \otimes R_{N/2})(F^{\otimes 2} \otimes G_{N/4}) \quad (69)$$

where (69) is obtained by using the identity $(AC) \otimes (BD) = (A \otimes B)(C \otimes D)$ with $A = I_2$, $B = R_{N/2}$, $C = F$, $D = F \otimes G_{N/4}$. Repeating this, we obtain

$$G_N = B_N F^{\otimes n} \quad (70)$$

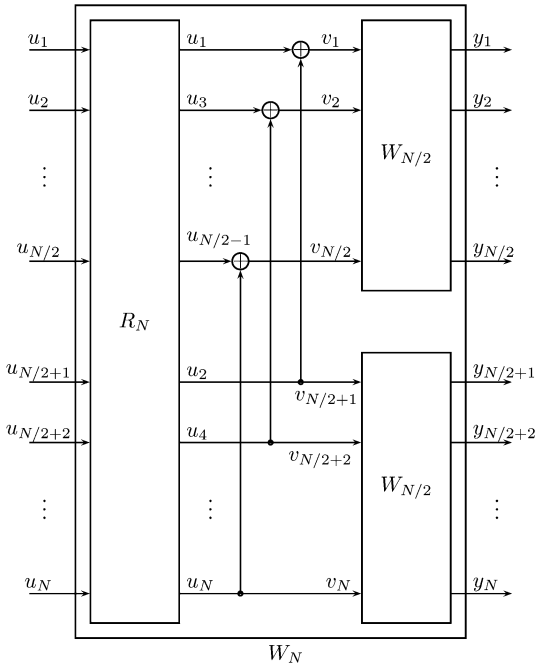

Fig. 8. An alternative realization of the recursive construction for W_N.

where $B_N \triangleq R_N(I_2 \otimes R_{N/2})(I_4 \otimes R_{N/4}) \cdots (I_{N/2} \otimes R_2)$. It can seen by simple manipulations that

$$B_N = R_N(I_2 \otimes B_{N/2}). \tag{71}$$

We can see that B_N is a permutation matrix by the following induction argument. Assume that $B_{N/2}$ is a permutation matrix for some $N \geq 4$; this is true for $N = 4$ since $B_2 = I_2$. Then, B_N is a permutation matrix because it is the product of two permutation matrices, R_N and $I_2 \otimes B_{N/2}$.

In the following, we will say more about the nature of B_N as a permutation.

B. Analysis by Bit-Indexing

To analyze the encoding operation further, it will be convenient to index vectors and matrices with bit sequences. Given a vector a_1^N with length $N = 2^n$ for some $n \geq 0$, we denote its ith element, a_i, $1 \leq i \leq N$, alternatively as $a_{b_1 \cdots b_n}$ where $b_1 \cdots b_n$ is the binary expansion of the integer $i - 1$ in the sense that $i = 1 + \sum_{j=1}^{n} b_j 2^{n-j}$. Likewise, the element A_{ij} of an N-by-N matrix A is denoted alternatively as $A_{b_1 \cdots b_n, b'_1 \cdots b'_n}$ where $b_1 \cdots b_n$ and $b'_1 \cdots b'_n$ are the binary representations of $i-1$ and $j-1$, respectively. Using this convention, it can be readily verified that the product $C = A \otimes B$ of a 2^n-by-2^n matrix A and a 2^m-by-2^m matrix B has elements $C_{b_1 \cdots b_{n+m}, b'_1 \cdots b'_{n+m}} = A_{b_1 \cdots b_n, b'_1 \cdots b'_n} B_{b_{n+1} \cdots b_{n+m}, b'_{n+1} \cdots b'_{n+m}}$.

We now consider the encoding operation under bit-indexing. First, we observe that the elements of F in bit-indexed form are given by $F_{b,b'} = 1 \oplus b' \oplus bb'$ for all $b, b' \in \{0,1\}$. Thus, $F^{\otimes n}$ has elements

$$\begin{aligned} F^{\otimes n}_{b_1 \cdots b_n, b'_1 \cdots b'_n} &= \prod_{i=1}^{n} F_{b_i, b'_i} \\ &= \prod_{i=1}^{n} (1 \oplus b'_i \oplus b_i b'_i). \end{aligned} \tag{72}$$

Second, the reverse shuffle operator R_N acts on a row vector u_1^N to replace the element in bit-indexed position $b_1 \cdots b_n$ with the element in position $b_2 \cdots b_n b_1$; that is, if $v_1^N = u_1^N R_N$, then $v_{b_1 \cdots b_n} = u_{b_2 \cdots b_n b_1}$ for all $b_1, \ldots, b_n \in \{0,1\}$. In other words, R_N cyclically rotates the bit-indexes of the elements of a left operand u_1^N to the right by one place.

Third, the matrix B_N in (70) can be interpreted as the *bit-reversal* operator: if $v_1^N = u_1^N B_N$, then $v_{b_1 \cdots b_n} = u_{b_n \cdots b_1}$ for all $b_1, \ldots, b_n \in \{0,1\}$. This statement can be proved by induction using the recursive formula (71). We give the idea of such a proof by an example. Let us assume that B_4 is a bit-reversal operator and show that the same is true for B_8. Let u_1^8 be any vector over GF(2). Using bit-indexing, it can be written as $(u_{000}, u_{001}, u_{010}, u_{011}, u_{100}, u_{101}, u_{110}, u_{111})$. Since $u_1^8 B_8 = u_1^8 R_8 (I_2 \otimes B_4)$, let us first consider the action of R_8 on u_1^8. The reverse shuffle R_8 rearranges the elements of u_1^8 with respect to odd–even parity of their indices, so $u_1^8 R_8$ equals $(u_{000}, u_{010}, u_{100}, u_{110}, u_{001}, u_{011}, u_{101}, u_{111})$. This has two halves, $c_1^4 \triangleq (u_{000}, u_{010}, u_{100}, u_{110})$ and $d_1^4 \triangleq (u_{001}, u_{011}, u_{101}, u_{111})$, corresponding to odd–even index classes. Notice that $c_{b_1 b_2} = u_{b_1 b_2 0}$ and $d_{b_1 b_2} = u_{b_1 b_2 1}$ for all $b_1, b_2 \in \{0,1\}$. This is to be expected since the reverse shuffle rearranges the indices in increasing order within each odd–even index class. Next, consider the action of $I_2 \otimes B_4$ on (c_1^4, d_1^4). The result is $(c_1^4 B_4, d_1^4 B_4)$. By assumption, B_4 is a bit-reversal operation, so $c_1^4 B_4 = (c_{00}, c_{10}, c_{01}, c_{11})$, which in turn equals $(u_{000}, u_{100}, u_{010}, u_{110})$. Likewise, the result of $d_1^4 B_4$ equals $(u_{001}, u_{101}, u_{011}, u_{111})$. Hence, the overall operation B_8 is a bit-reversal operation.

Given the bit-reversal interpretation of B_N, it is clear that B_N is a symmetric matrix, so $B_N^T = B_N$. Since B_N is a permutation, it follows from symmetry that $B_N^{-1} = B_N$.

It is now easy to see that, for any N-by-N matrix A, the product $C = B_N^T A B_N$ has elements $C_{b_1 \cdots b_n, b'_1 \cdots b'_n} = A_{b_n \cdots b_1, b'_n \cdots b'_1}$. It follows that if A is invariant under bit-reversal, i.e., if $A_{b_1 \cdots b_n, b'_1 \cdots b'_n} = A_{b_n \cdots b_1, b'_n \cdots b'_1}$ for every $b_1, \ldots, b_n, b'_1, \ldots, b'_n \in \{0,1\}$, then $A = B_N^T A B_N$. Since $B_N^T = B_N^{-1}$, this is equivalent to $B_N A = A B_T$. Thus, bit-reversal-invariant matrices commute with the bit-reversal operator.

Proposition 16: For any $N = 2^n$, $n \geq 1$, the generator matrix G_N is given by $G_N = B_N F^{\otimes n}$ and $G_N = F^{\otimes n} B_N$ where B_N is the bit-reversal permutation. G_N is a bit-reversal invariant matrix with

$$(G_N)_{b_1 \cdots b_n, b'_1 \cdots b'_n} = \prod_{i=1}^{n} (1 \oplus b'_i \oplus b_{n-i} b'_i). \tag{73}$$

Proof: $F^{\otimes n}$ commutes with B_N because it is invariant under bit-reversal, which is immediate from (72). The statement $G_N = B_N F^{\otimes n}$ was established before; by proving that $F^{\otimes n}$ commutes with B_N, we have established the other statement: $G_N = F^{\otimes n} B_N$. The bit-indexed form (73) follows by applying bit-reversal to (72). □

Finally, we give a fact that will be useful in Section X.

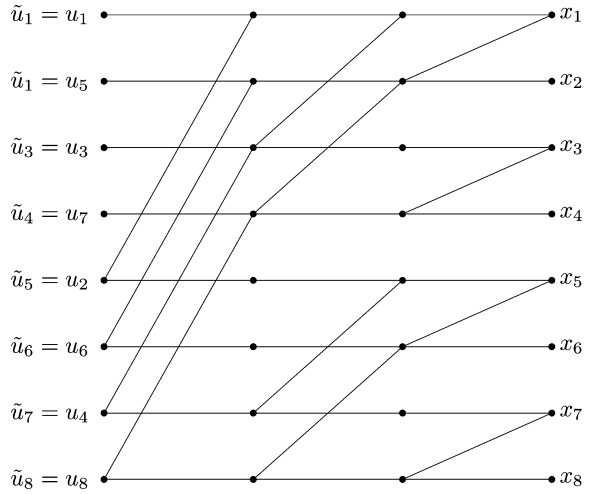

Fig. 9. A circuit for implementing the transformation $F^{\otimes 3}$. Signals flow from left to right. Each edge carries a signal 0 or 1. Each node adds (mod-2) the signals on all incoming edges from the left and sends the result out on all edges to the right. (Edges carrying the signals u_i and x_i are not shown.)

Proposition 17: For any $N = 2^n$, $n \geq 0$, $b_1, \ldots, b_n \in \{0,1\}$, the rows of G_N and $F^{\otimes n}$ with index $b_1 \cdots b_n$ have the same Hamming weight given by $2^{w_H(b_1,\ldots,b_n)}$, where

$$w_H(b_1, \ldots, b_n) \triangleq \sum_{i=1}^{n} b_i \tag{74}$$

is the Hamming weight of $(b_1, \ldots, b_n)$.

Proof: For fixed $b_1, \ldots, b_n$, the sum of the terms $(G_N)_{b_1 \ldots b_n, b'_1 \ldots b'_n}$ (as integers) over all $b'_1, \ldots, b'_n \in \{0,1\}$ gives the Hamming weight of the row of G_N with index $b_1 \cdots b_n$. From the preceding formula for $(G_N)_{b_1 \ldots b_n, b'_1 \ldots b'_n}$, this sum is easily seen to be $2^{w_H(b_1,\ldots,b_n)}$. The proof for $F^{\otimes n}$ is similar □

C. Encoding Complexity

For complexity estimation, our computational model will be a single-processor machine with a random-access memory. The complexities expressed will be time complexities. The discussion will be given for an arbitrary G_N-coset code with parameters $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$.

Let $\chi_E(N)$ denote the worst case encoding complexity over all $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ codes with a given block length N. If we take the complexity of a scalar mod-2 addition as one unit and the complexity of the reverse shuffle operation R_N as N units, we see from Fig. 3 that $\chi_E(N) \leq N/2 + N + 2\chi_E(N/2)$. Starting with an initial value $\chi_E(2) = 3$ (a generous figure), we obtain by induction that $\chi_E(N) \leq \frac{3}{2} N \log N$ for all $N = 2^n$, $n \geq 1$. Thus, the encoding complexity is $O(N \log N)$.

A specific implementation of the encoder using the form $G_N = B_N F^{\otimes n}$ is shown in Fig. 9 for $N = 8$. The input to the circuit is the bit-reversed version of u_1^8, i.e., $\tilde{u}_1^8 = u_1^8 B_8$. The output is given by $x_1^8 = \tilde{u}_1^8 F^{\otimes 3} = u_1^8 G_8$. In general, the complexity of this implementation is $O(N \log N)$ with $O(N)$ for B_N and $O(N \log N)$ for $F^{\otimes n}$.

An alternative implementation of the encoder would be to apply u_1^8 in natural index order at the input of the circuit in Fig. 9. Then, we would obtain $\tilde{x}_1^8 = u_1^8 F^{\otimes 3}$ at the output. Encoding could be completed by a post bit-reversal operation: $x_1^8 = \tilde{x}_1^8 B_8 = u_1^8 G_8$.

The encoding circuit of Fig. 9 suggests many parallel implementation alternatives for $F^{\otimes n}$: for example, with N processors, one may do a "column-by-column" implementation, and reduce the total latency to $\log N$. Various other tradeoffs are possible between latency and hardware complexity.

In an actual implementation of polar codes, it may be preferable to use $F^{\otimes n}$ in place of $B_N F^{\otimes n}$ as the encoder mapping in order to simplify the implementation. In that case, the SC decoder should compensate for this by decoding the elements of the source vector u_1^N in bit-reversed index order. We have included B_N as part of the encoder in this paper in order to have an SC decoder that decodes u_1^N in the natural index order, which simplified the notation.

VIII. Decoding

In this section, we consider the computational complexity of the SC decoding algorithm. As in the previous section, our computational model will be a single processor machine with a random-access memory and the complexities expressed will be time complexities. Let $\chi_D(N)$ denote the worst case complexity of SC decoding over all G_N-coset codes with a given block length N. We will show that $\chi_D(N) = O(N \log N)$.

A. A First Decoding Algorithm

Consider SC decoding for an arbitrary G_N-coset code with parameter $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$. Recall that the source vector u_1^N consists of a random part $u_{\mathcal{A}}$ and a frozen part $u_{\mathcal{A}^c}$. This vector is transmitted across W_N and a channel output y_1^N is obtained with probability $W_N(y_1^N | u_1^N)$. The SC decoder observes $(y_1^N, u_{\mathcal{A}^c})$ and generates an estimate $\hat{u}_1^N$ of u_1^N. We may visualize the decoder as consisting of N decision elements (DEs), one for each source element u_i; the DEs are activated in the order 1 to N. If $i \in \mathcal{A}^c$, the element u_i is known; so, the ith DE, when its turn comes, simply sets $\hat{u}_i = u_i$ and sends this result to all succeeding DEs. If $i \in \mathcal{A}$, the ith DE waits until it has received the previous decisions $\hat{u}_1^{i-1}$, and upon receiving them, computes the likelihood ratio (LR)

$$L_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1}\right) \triangleq \frac{W_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1} | 0\right)}{W_N^{(i)}\left(y_1^N, \hat{u}_1^{i-1} | 1\right)}$$

and generates its decision as

$$\hat{u}_i = \begin{cases} 0, & \text{if } L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 1 \\ 1, & \text{otherwise} \end{cases}$$

which is then sent to all succeeding DEs. This is a single-pass algorithm, with no revision of estimates. The complexity of this algorithm is determined essentially by the complexity of computing the LRs.

A straightforward calculation using the recursive formulas (22) and (23) gives

$$L_N^{(2i-1)}\left(y_1^N, \hat{u}_1^{2i-2}\right) = \frac{L_{N/2}^{(i)}\left(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) L_{N/2}^{(i)}\left(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2}\right) + 1}{L_{N/2}^{(i)}\left(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) + L_{N/2}^{(i)}\left(y_{N/2+1}^{N}, \hat{u}_{1,e}^{2i-2}\right)} \tag{75}$$

and

$$L_N^{(2i)}\left(y_1^N, \hat{u}_1^{2i-1}\right) = \left[L_{N/2}^{(i)}\left(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right)\right]^{1-2\hat{u}_{2i-1}} \cdot L_{N/2}^{(i)}\left(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}\right). \quad (76)$$

Thus, the calculation of an LR at length N is reduced to the calculation of two LRs at length $N/2$. This recursion can be continued down to block length 1, at which point the LRs have the form $L_1^{(1)}(y_i) = W(y_i|0)/W(y_i|1)$ and can be computed directly.

To estimate the complexity of LR calculations, let $\chi_L(k)$, $k \in \{N, N/2, N/4, \ldots, 1\}$, denote the worst case complexity of computing $L_k^{(i)}(y_1^k, v_1^{i-1})$ over $i \in [1,k]$ and $(y_1^k, v_1^{i-1}) \in \mathcal{Y}^k \times \mathcal{X}^{i-1}$. From the recursive LR formulas, we have the complexity bound

$$\chi_L(k) \leq 2\chi_L(k/2) + \alpha \quad (77)$$

where α is the worst case complexity of assembling two LRs at length $k/2$ into an LR at length k. Taking $\chi_L(1)$ as one unit, we obtain the bound

$$\chi_L(N) \leq (1+\alpha)N = O(N). \quad (78)$$

The overall decoder complexity can now be bounded as $\chi_D(N) \leq K\chi_L(N) \leq N\chi_L(N) = O(N^2)$. This complexity corresponds to a decoder whose DEs do their LR calculations privately, without sharing any partial results with each other. It turns out, if the DEs pool their scratch-pad results, a more efficient decoder implementation is possible with overall complexity $O(N \log N)$, as we will show next.

B. Refinement of the Decoding Algorithm

We now consider a decoder that computes the full set of LRs, $\{L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) : 1 \leq i \leq N\}$. The previous decoder could skip the calculation of $L_N^{(i)}(y_1^N, \hat{u}_1^{i-1})$ for $i \in \mathcal{A}^c$; but now we do not allow this. The decisions $\{\hat{u}_i : 1 \leq i \leq N\}$ are made in exactly the same manner as before; in particular, if $i \in \mathcal{A}^c$, the decision $\hat{u}_i$ is set to the known frozen value u_i, regardless of $L_N^{(i)}(y_1^N, \hat{u}_1^{i-1})$.

To see where the computational savings will come from, we inspect (75) and (76) and note that each LR value in the pair

$$\left(L_N^{(2i-1)}\left(y_1^N, \hat{u}_1^{2i-2}\right), L_N^{(2i)}\left(y_1^N, \hat{u}_1^{2i-1}\right)\right)$$

is assembled from the same pair of LRs

$$\left(L_{N/2}^{(i)}\left(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right), L_{N/2}^{(i)}\left(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}\right)\right).$$

Thus, the calculation of all N LRs at length N requires exactly N LR calculations at length $N/2$.[3] Let us split the N LRs at length $N/2$ into two classes, namely

$$\left\{L_{N/2}^{(i)}\left(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) : 1 \leq i \leq N/2\right\}$$
$$\left\{L_{N/2}^{(i)}\left(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}\right) : 1 \leq i \leq N/2\right\}. \quad (79)$$

[3]Actually, some LR calculations at length $N/2$ may be avoided if, by chance, some duplications occur, but we will disregard this.

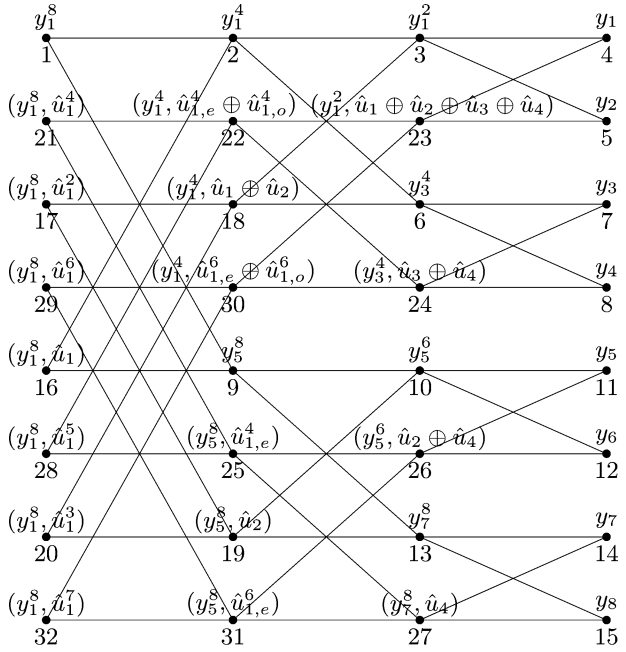

Fig. 10. An implementation of the successive cancellation decoder for polar coding at block-length $N = 8$.

Let us suppose that we carry out the calculations in each class independently, without trying to exploit any further savings that may come from the sharing of LR values between the two classes. Then, we have two problems of the same type as the original but at half the size. Each class in (79) generates a set of $N/2$ LR calculation requests at length $N/4$, for a total of N requests. For example, if we let $\hat{v}_1^{N/2} \triangleq \hat{u}_{1,o}^{N/2} \oplus \hat{u}_{1,e}^{N/2}$, the requests arising from the first class are

$$\left\{L_{N/4}^{(i)}\left(y_1^{N/4}, \hat{v}_{1,o}^{2i-2} \oplus \hat{v}_{1,e}^{2i-2}\right) : 1 \leq i \leq N/4\right\}$$
$$\left\{L_{N/4}^{(i)}\left(y_{N/4+1}^{N/2}, \hat{v}_{1,e}^{2i-2}\right) : 1 \leq i \leq N/4\right\}.$$

Using this reasoning inductively across the set of all lengths $\{N, N/2, \ldots, 1\}$, we conclude that the total number of LRs that need to be calculated is $N(1+\log N)$.

So far, we have not paid attention to the exact order in which the LR calculations at various block lengths are carried out. Although this gave us an accurate count of the total number of LR calculations, for a full description of the algorithm, we need to specify an order. There are many possibilities for such an order, but to be specific we will use a depth-first algorithm, which is easily described by a small example.

We consider a decoder for a code with parameter $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ chosen as $(8, 5, \{3, 5, 6, 7, 8\}, (0, 0, 0))$. The computation for the decoder is laid out in a graph as shown in Fig. 10. There are $N(1+\log N) = 32$ nodes in the graph, each responsible for computing an LR request that arises during the course of the algorithm. Starting from the left side, the first column of nodes correspond to LR requests at length 8 (decision level), the second column of nodes to requests at length 4, the third at length 2, and the fourth at length 1 (channel level).

Each node in the graph carries two labels. For example, the third node from the bottom in the third column has the labels $(y_5^6, \hat{u}_2 \oplus \hat{u}_4)$ and 26; the first label indicates that the LR value to

be calculated at this node is $L_8^{(2)}(y_5^6, \hat{u}_2 \oplus \hat{u}_4)$ while the second label indicates that this node will be the 26th node to be activated. The numeric labels, 1 through 32, will be used as quick identifiers in referring to nodes in the graph.

The decoder is visualized as consisting of N DEs situated at the leftmost side of the decoder graph. The node with label $(y_1^8, \hat{u}_1^{i-1})$ is associated with the ith DE, $1 \leq i \leq 8$. The positioning of the DEs in the leftmost column follows the bit-reversed index order, as in Fig. 9.

Decoding begins with DE 1 activating node 1 for the calculation of $L_8^{(1)}(y_1^8)$. Node 1 in turn activates node 2 for $L_4^{(1)}(y_1^4)$. At this point, program control passes to node 2, and node 1 will wait until node 2 delivers the requested LR. The process continues. Node 2 activates node 3, which activates node 4. Node 4 is a node at the channel level; so it computes $L_1^{(1)}(y_1)$ and passes it to nodes 3 and 23, its left-side neighbors. In general, a node will send its computational result to all its left-side neighbors (although this will not be stated explicitly below). Program control will be passed back to the left neighbor from which it was received.

Node 3 still needs data from the right side and activates node 5, which delivers $L_1^{(1)}(y_2)$. Node 3 assembles $L_2^{(1)}(y_1^2)$ from the messages it has received from nodes 4 and 5 and sends it to node 2. Next, node 2 activates node 6, which activates nodes 7 and 8, and returns its result to node 2. Node 2 compiles its response $L_4^{(1)}(y_1^4)$ and sends it to node 1. Node 1 activates node 9 which calculates $L_4^{(1)}(y_5^8)$ in the same manner as node 2 calculated $L_4^{(1)}(y_1^4)$, and returns the result to node 1. Node 1 now assembles $L_8^{(1)}(y_1^8)$ and sends it to DE 1. Since u_1 is a frozen node, DE 1 ignores the received LR, declares $\hat{u}_1 = 0$, and passes control to DE 2, located next to node 16.

DE 2 activates node 16 for $L_8^{(2)}(y_1^8, \hat{u}_1)$. Node 16 assembles $L_8^{(2)}(y_1^8, \hat{u}_1)$ from the already-received LRs $L_4^{(1)}(y_1^4)$ and $L_4^{(1)}(y_5^8)$, and returns its response without activating any node. DE 2 ignores the returned LR since u_2 is frozen, announces $\hat{u}_2 = 0$, and passes control to DE 3.

DE 3 activates node 17 for $L_8^{(3)}(y_1^8, \hat{u}_1^2)$. This triggers LR requests at nodes 18 and 19, but no further. The bit u_3 is not frozen; so, the decision $\hat{u}_3$ is made in accordance with $L_8^{(3)}(y_1^8, \hat{u}_1^2)$, and control is passed to DE 4. DE 4 activates node 20 for $L_8^{(4)}(y_1^8, \hat{u}_1^3)$, which is readily assembled and returned. The algorithm continues in this manner until finally DE 8 receives $L_8^{(7)}(y_1^8, \hat{u}_1^7)$ and decides $\hat{u}_8$.

There are a number of observations that can be made by looking at this example that should provide further insight into the general decoding algorithm. First, notice that the computation of $L_8^{(1)}(y_1^8)$ is carried out in a subtree rooted at node 1, consisting of paths going from left to right, and spanning all nodes at the channel level. This subtree splits into two disjoint subtrees, namely, the subtree rooted at node 2 for the calculation of $L_4^{(1)}(y_1^4)$ and the subtree rooted at node 9 for the calculation of $L_4^{(1)}(y_5^8)$. Since the two subtrees are disjoint, the corresponding calculations can be carried out independently (even in parallel if there are multiple processors). This splitting of computational subtrees into disjoint subtrees holds for all nodes in the graph (except those at the channel level), making it possible to implement the decoder with a high degree of parallelism.

Second, we notice that the decoder graph consists of *butterflies* (2-by-2 complete bipartite graphs) that tie together adjacent levels of the graph. For example, nodes 9, 19, 10, and 13 form a butterfly. The computational subtrees rooted at nodes 9 and 19 split into a single pair of computational subtrees, one rooted at node 10, the other at node 13. Also note that among the four nodes of a butterfly, the upper-left node is always the first node to be activated by the above depth-first algorithm and the lower-left node always the last one. The upper-right and lower-right nodes are activated by the upper-left node and they may be activated in any order or even in parallel. The algorithm we specified always activated the upper-right node first, but this choice was arbitrary. When the lower-left node is activated, it finds the LRs from its right neighbors ready for assembly. The upper-left node assembles the LRs it receives from the right side as in formula (75), the lower-left node as in (76). These formulas show that the butterfly patterns impose a constraint on the completion time of LR calculations: in any given butterfly, the lower-left node needs to wait for the result of the upper-left node which in turn needs to wait for the results of the right-side nodes.

Variants of the decoder are possible in which the nodal computations are scheduled differently. In the "left-to-right" implementation given above, nodes waited to be activated. However, it is possible to have a "right-to-left" implementation in which each node starts its computation autonomously as soon as its right-side neighbors finish their calculations; this allows exploiting parallelism in computations to the maximum possible extent.

For example, in such a fully parallel implementation for the case in Fig. 10, all eight nodes at the channel-level start calculating their respective LRs in the first time slot following the availability of the channel output vector y_1^8. In the second time slot, nodes 3, 6, 10, and 13 do their LR calculations in parallel. Note that this is the maximum degree of parallelism possible in the second time slot. Node 23, for example, cannot calculate $L_N^{(2)}(y_1^2, \hat{u}_1 \oplus \hat{u}_2 \oplus \hat{u}_3 \oplus \hat{u}_4)$ in this slot, because $\hat{u}_1 \oplus \hat{u}_2 \oplus \hat{u}_3 \oplus \hat{u}_4$ is not yet available; it has to wait until decisions $\hat{u}_1$, $\hat{u}_2$, $\hat{u}_3$, $\hat{u}_4$ are announced by the corresponding DEs. In the third time slot, nodes 2 and 9 do their calculations. In time slot 4, the first decision $\hat{u}_1$ is made at node 1 and broadcast to all nodes across the graph (or at least to those that need it). In slot 5, node 16 calculates $\hat{u}_2$ and broadcasts it. In slot 6, nodes 18 and 19 do their calculations. This process continues until time slot 15 when node 32 decides $\hat{u}_8$. It can be shown that, in general, this fully parallel decoder implementation has a latency of $2N - 1$ time slots for a code of block-length N.

IX. Code Construction

The input to a polar code construction algorithm is a triple (W, N, K) where W is the B-DMC on which the code will be used, N is the code block length, and K is the dimensionality of the code. The output of the algorithm is an information set $\mathcal{A} \subset \{1, \ldots, N\}$ of size K such that $\sum_{i \in \mathcal{A}} Z(W_N^{(i)})$ is as small as possible. We exclude the search for a good frozen vector $u_{\mathcal{A}^c}$ from the code construction problem because the problem is already difficult enough. Recall that, for symmetric channels, the code performance is not affected by the choice of $u_{\mathcal{A}^c}$.

In principle, the code construction problem can be solved by computing all the parameters $\{Z(W_N^{(i)}) : 1 \leq i \leq N\}$ and sorting them; unfortunately, we do not have an efficient algorithm for doing this. For symmetric channels, some computational shortcuts are available, as we showed by Proposition 15, but these shortcuts have not yielded an efficient algorithm, either. One exception to all this is the BEC for which the parameters $\{Z(W_N^{(i)})\}$ can all be calculated in time $O(N)$ thanks to the recursive formulas (38).

Since exact code construction appears too complex, it makes sense to look for approximate constructions based on estimates of the parameters $\{Z(W_N^{(i)})\}$. To that end, it is preferable to pose the exact code construction problem as a decision problem: Given a threshold $\gamma \in [0,1]$ and an index $i \in \{1,\ldots,N\}$, decide whether $i \in \mathcal{A}_\gamma$ where

$$\mathcal{A}_\gamma \triangleq \left\{ i \in \{1,\ldots,N\} : Z\left(W_N^{(i)}\right) < \gamma \right\}.$$

Any algorithm for solving this decision problem can be used to solve the code construction problem. We can simply run the algorithm with various settings for γ until we obtain an information set $\mathcal{A}_\gamma$ of the desired size K.

Approximate code construction algorithms can be proposed based on statistically reliable and efficient methods for estimating whether $i \in \mathcal{A}_\gamma$ for any given pair (i,γ). The estimation problem can be approached by noting that, as we have implicitly shown in (54), the parameter $Z(W_N^{(i)})$ is the expectation of the RV

$$\sqrt{\frac{W_N^{(i)}\left(Y_1^N, U_1^{i-1} | U_i \oplus 1\right)}{W_N^{(i)}\left(Y_1^N, U_1^{i-1} | U_i\right)}} \tag{80}$$

where (U_1^N, Y_1^N) is sampled from the joint probability assignment $P_{U_1^N, Y_1^N}(u_1^N, y_1^N) \triangleq 2^{-N} W_N(y_1^N | u_1^N)$. A Monte Carlo approach can be taken, where samples of (U_1^N, Y_1^N) are generated from the given distribution and the empirical means $\{\hat{Z}(W_N^{(i)})\}$ are calculated. Given a sample (u_1^N, y_1^N) of (U_1^N, Y_1^N), the sample values of the RVs (80) can all be computed in complexity $O(N \log N)$. An SC decoder may be used for this computation since the sample values of (80) are just the square roots of the decision statistics that the DEs in an SC decoder ordinarily compute. (In applying an SC decoder for this task, the information set $\mathcal{A}$ should be taken as the null set.)

Statistical algorithms are helped by the polarization phenomenon: for any fixed γ and as N grows, it becomes easier to resolve whether $Z(W_N^{(i)}) < \gamma$, because an ever-growing fraction of the parameters $\{Z(W_N^{(i)})\}$ tend to cluster around 0 or 1.

It is conceivable that, in an operational system, the estimation of the parameters $\{Z(W_N^{(i)})\}$ is made part of an SC decoding procedure, with continual update of the information set as more reliable estimates become available.

X. A Note on the RM Rule

In this part, we return to the claim made in Section I-D that the RM rule for information set selection leads to asymptotically unreliable codes under SC decoding.

Recall that, for a given (N,K), the RM rule constructs a G_N-coset code with parameter $(N, K, \mathcal{A}, u_{\mathcal{A}^c})$ by prioritizing each index $i \in \{1,\ldots,N\}$ for inclusion in the information set $\mathcal{A}$ w.r.t. the Hamming weight of the ith row of G_N. The RM rule sets the frozen bits $u_{\mathcal{A}^c}$ to zero. In light of Proposition 17, the RM rule can be restated in bit-indexed terminology as follows.

RM Rule: For a given (N,K), with $N = 2^n$, $n \geq 0$, $0 \leq K \leq N$, choose $\mathcal{A}$ as follows: i) Determine the integer r such that

$$\sum_{k=r}^{n} \binom{n}{k} \leq K < \sum_{k=r-1}^{n} \binom{n}{k}. \tag{81}$$

ii) Put each index $b_1 \cdots b_n$ with $w_H(b_1,\ldots,b_n) \geq r$ into $\mathcal{A}$. iii) Put sufficiently many additional indices $b_1 \cdots b_n$ with $w_H(b_1,\ldots,b_n) = r-1$ into $\mathcal{A}$ to complete its size to K.

We observe that this rule will select the index

$$0^{n-r}1^r \triangleq \overbrace{0\cdots0}^{n-r}\overbrace{1\cdots1}^{r}$$

for inclusion in $\mathcal{A}$. This index turns out to be a particularly poor choice, at least for the class of BECs, as we show in the remaining part of this section.

Let us assume that the code constructed by the RM rule is used on a BEC W with some erasure probability $\epsilon > 0$. We will show that the symmetric capacity $I(W_{0^{n-r}1^r})$ converges to zero for any fixed positive coding rate as the block length is increased. For this, we recall the relations (6), which, in bit-indexed channel notation of Section IV, can be written as follows. For any $\ell \geq 1$, $b_1,\ldots,b_\ell \in \{0,1\}$

$$\begin{aligned} I(W_{b_1\cdots b_\ell 0}) &= I(W_{b_1\cdots b_\ell})^2 \\ I(W_{b_1\cdots b_\ell 1}) &= 2I(W_{b_1\cdots b_\ell}) - I(W_{b_1\cdots b_\ell})^2 \\ &\leq 2I(W_{b_1\cdots b_\ell}) \end{aligned}$$

with initial values $I(W_0) = I^2(W)$ and $I(W_1) = 2I(W) - I^2(W)$. These give the bound

$$I(W_{0^{n-r}1^r}) \leq 2^r (1-\epsilon)^{2^{n-r}}. \tag{82}$$

Now, consider a sequence of RM codes with a fixed rate $0 < R < 1$, N increasing to infinity, and $K = \lfloor NR \rfloor$. Let $r(N)$ denote the parameter r in (81) for the code with block length N in this sequence. Let $n = \log_2(N)$. A simple asymptotic analysis shows that the ratio $r(N)/n$ must go to $1/2$ as N is increased. This in turn implies by (82) that $I(W_{0^{n-r}1^r})$ must go to zero.

Suppose that this sequence of RM codes is decoded using an SC decoder as in Section I-C.2 where the decision metric ignores knowledge of frozen bits and instead uses randomization over all possible choices. Then, as N goes to infinity, the SC decoder decision element with index $0^{n-r}1^r$ sees a channel whose capacity goes to zero, while the corresponding element of the input vector u_1^N is assigned 1 bit of information by the RM rule. This means that the RM code sequence is asymptotically unreliable under this type of SC decoding.

We should emphasize that the above result does not say that RM codes are asymptotically bad under *any* SC decoder, nor does it make a claim about the performance of RM codes under other decoding algorithms. (It is interesting that the possibility of RM codes being capacity-achieving codes under ML decoding seems to have received no attention in the literature.)

XI. Concluding Remarks

In this section, we go through the paper to discuss some results further, point out some generalizations, and state some open problems.

A. Rate of Polarization

A major open problem suggested by this paper is to determine how fast a channel polarizes as a function of the block-length parameter N. In recent work [12], the following result has been obtained in this direction.

Proposition 18: Let W be a B-DMC. For any fixed rate $R < I(W)$ and constant $\beta < \frac{1}{2}$, there exists a sequence of sets $\{\mathcal{A}_N\}$ such that $\mathcal{A}_N \subset \{1, \ldots, N\}$, $|\mathcal{A}_N| \geq NR$, and

$$\sum_{i \in \mathcal{A}_N} Z\left(W_N^{(i)}\right) = o(2^{-N^\beta}). \tag{83}$$

Conversely, if $R > 0$ and $\beta > \frac{1}{2}$, then for any sequence of sets $\{\mathcal{A}_N\}$ with $\mathcal{A}_N \subset \{1, \ldots, N\}$, $|\mathcal{A}_N| \geq NR$, we have

$$\max\left\{Z\left(W_N^{(i)}\right) : i \in \mathcal{A}_N\right\} = \omega(2^{-N^\beta}). \tag{84}$$

As a corollary, Theorem 3 is strengthened as follows.

Proposition 19: For polar coding on a B-DMC W at any fixed rate $R < I(W)$, and any fixed $\beta < \frac{1}{2}$

$$P_e(N, R) = o(2^{-N^\beta}). \tag{85}$$

This is a vast improvement over the $O(N^{-\frac{1}{4}})$ bound proved in this paper. Note that the bound still does not depend on the rate R as long as $R < I(W)$. A problem of theoretical interest is to obtain sharper bounds on $P_e(N, R)$ that show a more explicit dependence on R.

Another problem of interest related to polarization is *robustness* against channel parameter variations. A finding in this regard is the following result [13]: If a polar code is designed for a B-DMC W but used on some other B-DMC W', then the code will perform at least as well as it would perform on W provided W is a degraded version of W' in the sense of Shannon [14]. This result gives reason to expect a graceful degradation of polar-coding performance due to errors in channel modeling.

B. Generalizations

The polarization scheme considered in this paper can be generalized as shown in Fig. 11. In this general form, the channel input alphabet is assumed q-ary, $\mathcal{X} = \{0, 1, \ldots, q-1\}$, for some $q \geq 2$. The construction begins by combining m independent copies of a DMC $W : \mathcal{X} \to \mathcal{Y}$ to obtain W_m, where $m \geq 2$ is a fixed parameter of the construction. The general step

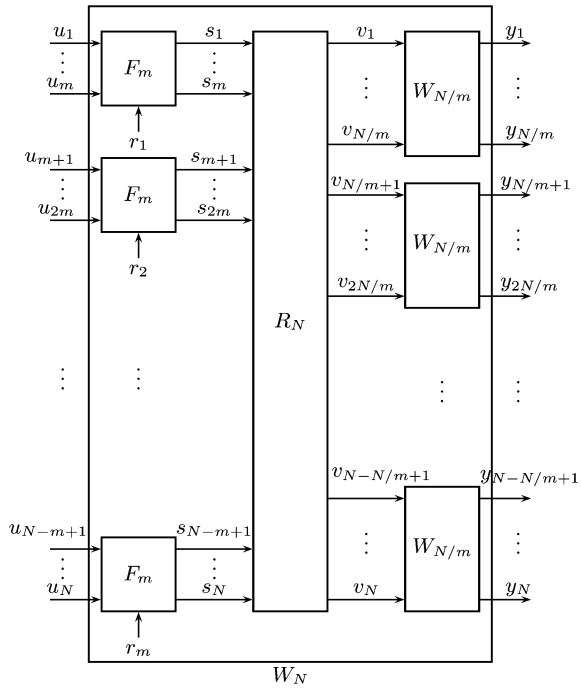

Fig. 11. General form of channel combining.

combines m independent copies of the channel $W_{N/m}$ from the previous step to obtain W_N. In general, the size of the construction is $N = m^n$ after n steps. The construction is characterized by a *kernel* $F_m : \mathcal{X}^m \times \mathcal{R} \to \mathcal{X}^m$ where $\mathcal{R}$ is some finite set included in the mapping for *randomization*. The reason for introducing randomization will be discussed shortly.

The vectors $u_1^N \in \mathcal{X}^N$ and $y_1^N \in \mathcal{Y}^N$ in Fig. 11 denote the input and output vectors of W_N. The input vector is first transformed into a vector $s_1^N \in \mathcal{X}^N$ by breaking it into N consecutive subblocks of length m, namely, $u_1^m, \ldots, u_{N-m+1}^N$, and passing each subblock through the transform F_m. Then, a permutation R_N sorts the components of s_1^N w.r.t. mod-m residue classes of their indices. The sorter ensures that, for any $1 \leq k \leq m$, the kth copy of $W_{N/m}$, counting from the top of the figure, gets as input those components of s_1^N whose indices are congruent to k mod-m. For example, $v_1 = s_1$, $v_2 = s_{m+1}$, $v_{N/m} = s_{(N/m-1)m+1}$, $v_{N/m+1} = s_2$, $v_{N/m+2} = s_{m+2}$, and so on. The general formula is $v_{kN/m+j} = s_{k+(j-1)m+1}$ for all $0 \leq k \leq (m-1)$, $1 \leq j \leq N/m$.

We regard the randomization parameters $r_1, \ldots, r_m$ as being chosen at random at the time of code construction, but fixed throughout the operation of the system; the decoder operates with full knowledge of them. For the binary case considered in this paper, we did not employ any randomization. Here, randomization has been introduced as part of the general construction because preliminary studies show that it greatly simplifies the analysis of generalized polarization schemes. This subject will be explored further in future work.

Certain additional constraints need to be placed on the kernel F_m to ensure that a polar code can be defined that is suitable for SC decoding in the natural order u_1 to u_N. To that end, it is sufficient to restrict F_m to *unidirectional* functions, namely, invertible functions of the form $F_m : (u_1^m, r) \in \mathcal{X}^m \times \mathcal{R} \mapsto$

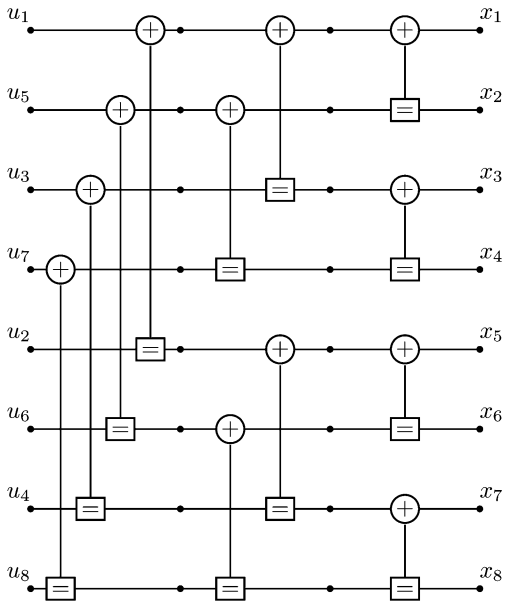

Fig. 12. The factor graph representation for the transformation $F^{\otimes 3}$.

$x_1^m \in \mathcal{X}^m$ such that $x_i = f_i(u_i^m, r)$, for a given set of coordinate functions $f_i : \mathcal{X}^{m-i+1} \times \mathcal{R} \to \mathcal{X}, i = 1, \ldots, m$. For a unidirectional F_m, the combined channel W_N can be split to channels $\{W_N^{(i)}\}$ in much the same way as in this paper. The encoding and SC decoding complexities of such a code are both $O(N \log N)$.

Polar coding can be generalized further in order to overcome the restriction of the block length N to powers of a given number m by using a sequence of kernels F_{m_i}, $i = 1, \ldots, n$, in the code construction. Kernel F_{m_1} combines m_1 copies of a given DMC W to create a channel W_{m_1}. Kernel F_{m_2} combines m_2 copies of W_{m_1} to create a channel $W_{m_1 m_2}$, etc., for an overall block-length of $N = \prod_{i=1}^n m_i$. If all kernels are unidirectional, the combined channel W_N can still be split into channels $W_N^{(i)}$ whose transition probabilities can be expressed by recursive formulas and $O(N \log N)$ encoding and decoding complexities are maintained.

So far we have considered only combining copies of one DMC W. Another direction for generalization of the method is to combine copies of two or more distinct DMCs. For example, the kernel F considered in this paper can be used to combine copies of any two B-DMCs W, W'. The investigation of coding advantages that may result from such variations on the basic code construction method is an area for further research.

It is easy to propose variants and generalizations of the basic channel polarization scheme, as we did above; however, it is not clear if we obtain channel polarization under each such variant. We conjecture that channel polarization is a common phenomenon, which is almost impossible to avoid as long as channels are combined with a sufficient density and mix of connections, whether chosen recursively or at random, provided the coordinate-wise splitting of the synthesized vector channel is done according to a suitable SC decoding order. The study of channel polarization in such generality is an interesting theoretical problem.

C. Iterative Decoding of Polar Codes

We have seen that polar coding under SC decoding can achieve symmetric channel capacity; however, one needs to use codes with impractically large block lengths. A question of interest is whether polar coding performance can improve significantly under more powerful decoding algorithms. The sparseness of the graph representation of $F^{\otimes n}$ makes Gallager's belief propagation (BP) decoding algorithm [15] applicable to polar codes. A highly relevant work in this connection is [16] which proposes BP decoding for RM codes using a factor-graph of $F^{\otimes n}$, as shown in Fig. 12 for $N = 8$. We carried out experimental studies to assess the performance of polar codes under BP decoding, using RM codes under BP decoding as a benchmark [17]. The results showed significantly better performance for polar codes. Also, the performance of polar codes under BP decoding was significantly better than their performance under SC decoding. However, more work needs to be done to assess the potential of polar coding for practical applications.

APPENDIX

A. Proof of Proposition 1

The RHS of (1) equals the channel parameter $E_0(1, Q)$ as defined in Gallager [10, Sec. 5.6] with Q taken as the uniform input distribution. (This is the *symmetric cutoff rate* of the channel.) It is well known (and shown in the same section of [10]) that $I(W) \geq E_0(1, Q)$. This proves (1).

To prove (2), for any B-DMC $W : \mathcal{X} \to \mathcal{Y}$, define

$$d(W) \triangleq \frac{1}{2} \sum_{y \in \mathcal{Y}} |W(y|0) - W(y|1)|.$$

This is the variational distance between the two distributions $W(y|0)$ and $W(y|1)$ over $y \in \mathcal{Y}$.

Lemma 2: For any B-DMC W, $I(W) \leq d(W)$.

Proof: Let W be an arbitrary B-DMC with output alphabet $\mathcal{Y} = \{1, \ldots, n\}$ and put $P_i = W(i|0)$, $Q_i = W(i|1)$, $i = 1, \ldots, n$. By definition

$$I(W) = \sum_{i=1}^n \frac{1}{2} \left[P_i \log \frac{P_i}{\frac{1}{2}P_i + \frac{1}{2}Q_i} + Q_i \log \frac{Q_i}{\frac{1}{2}P_i + \frac{1}{2}Q_i} \right].$$

The ith bracketed term under the summation is given by

$$f(x) \triangleq x \log \frac{x}{x+\delta} + (x + 2\delta) \log \frac{x + 2\delta}{x + \delta}$$

where $x = \min\{P_i, Q_i\}$ and $\delta = \frac{1}{2}|P_i - Q_i|$. We now consider maximizing $f(x)$ over $0 \leq x \leq 1 - 2\delta$. We compute

$$\frac{df}{dx} = \frac{1}{2} \log \frac{\sqrt{x(x+2\delta)}}{(x+\delta)}$$

and recognize that $\sqrt{x(x+2\delta)}$ and $(x+\delta)$ are, respectively, the geometric and arithmetic means of the numbers x and $(x+2\delta)$. So, $df/dx \leq 0$ and $f(x)$ is maximized at $x = 0$, giving the

inequality $f(x) \leq 2\delta$. Using this in the expression for $I(W)$, we obtain the claim of the lemma

$$I(W) \leq \sum_{i=1} \frac{1}{2} |P_i - Q_i| = d(W). \qquad \square$$

Lemma 3: For any B-DMC W, $d(W) \leq \sqrt{1 - Z(W)^2}$.

Proof: Let W be an arbitrary B-DMC with output alphabet $\mathcal{Y} = \{1, \ldots, n\}$ and put $P_i = W(i|0)$, $Q_i = W(i|1)$, $i = 1, \ldots, n$. Let $\delta_i \triangleq \frac{1}{2}|P_i - Q_i|$, $\delta \triangleq d(W) = \sum_{i=1}^n \delta_i$, and $R_i \triangleq (P_i + Q_i)/2$. Then, we have $Z(W) = \sum_{i=1}^n \sqrt{(R_i - \delta_i)(R_i + \delta_i)}$. Clearly, $Z(W)$ is upper-bounded by the maximum of $\sum_{i=1}^n \sqrt{R_i^2 - \delta_i^2}$ over $\{\delta_i\}$ subject to the constraints that $0 \leq \delta_i \leq R_i$, $i = 1, \ldots, n$, and $\sum_{i=1}^n \delta_i = \delta$. To carry out this maximization, we compute the partial derivatives of $Z(W)$ with respect to δ_i

$$\frac{\partial Z}{\partial \delta_i} = -\frac{\delta_i}{\sqrt{R_i^2 - \delta_i^2}}$$
$$\frac{\partial^2 Z}{\partial \delta_i^2} = -\frac{R_i^2}{(R_i^2 - \delta_i^2)^{\frac{3}{2}}}$$

and observe that $Z(W)$ is a decreasing, concave function of δ_i for each i, within the range $0 \leq \delta_i \leq R_i$. The maximum occurs at the solution of the set of equations $\partial Z/\partial \delta_i = k$, all i, where k is a constant, i.e., at $\delta_i = R_i\sqrt{k^2/(1+k^2)}$. Using the constraint $\sum_i \delta_i = \delta$ and the fact that $\sum_{i=1}^n R_i = 1$, we find $\sqrt{k^2/(1+k^2)} = \delta$. So, the maximum occurs at $\delta_i = \delta R_i$ and has the value $\sum_{i=1}^n \sqrt{R_i^2 - \delta^2 R_i^2} = \sqrt{1 - \delta^2}$. We have thus shown that $Z(W) \leq \sqrt{1 - d(W)^2}$, which is equivalent to $d(W) \leq \sqrt{1 - Z(W)^2}$. $\square$

From the above two lemmas, the proof of (2) is immediate.

B. Proof of Proposition 3

To prove (22), we write

$$\begin{aligned} W_{2N}^{(2i-1)} &\left(y_1^{2N}, u_1^{2i-2} | u_{2i-1}\right) \\ &= \sum_{u_{2i}^{2N}} \frac{1}{2^{2N-1}} W_{2N}\left(y_1^{2N} | u_1^{2N}\right) \\ &= \sum_{u_{2i,o}^{2N}, u_{2i,e}^{2N}} \frac{1}{2^{2N-1}} W_N\left(y_1^N | u_{1,o}^{2N} \oplus u_{1,e}^{2N}\right) W_N\left(y_{N+1}^{2N} | u_{1,e}^{2N}\right) \\ &= \sum_{u_{2i}} \frac{1}{2} \sum_{u_{2i+1,e}^{2N}} \frac{1}{2^{N-1}} W_N\left(y_{N+1}^{2N} | u_{1,e}^{2N}\right) \\ &\quad \cdot \sum_{u_{2i+1,o}^{2N}} \frac{1}{2^{N-1}} W_N\left(y_1^N | u_{1,o}^{2N} \oplus u_{1,e}^{2N}\right). \end{aligned} \tag{86}$$

By definition (5), the sum over $u_{2i+1,o}^{2N}$ for any fixed $u_{1,e}^{2N}$ equals

$$W_N^{(i)}\left(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2} | u_{2i-1} \oplus u_{2i}\right)$$

because, as $u_{2i+1,o}^{2N}$ ranges over $\mathcal{X}^{N-i}$, $u_{2i+1,o}^{2N} \oplus u_{2i+1,e}^{2N}$ ranges also over $\mathcal{X}^{N-i}$. We now factor this term out of the middle sum in (86) and use (5) again to obtain (22). For the proof of (23), we write

$$\begin{aligned} W_{2N}^{(2i)}\left(y_1^{2N}, u_1^{2i-1} | u_{2i}\right) &= \sum_{u_{2i+1}^{2N}} \frac{1}{2^{2N-1}} W_{2N}\left(y_1^{2N} | u_1^{2N}\right) \\ &= \frac{1}{2} \sum_{u_{2i+1,e}^{2N}} \frac{1}{2^{N-1}} W_N\left(y_{N+1}^{2N} | u_{1,e}^{2N}\right) \\ &\quad \cdot \sum_{u_{2i+1,o}^{2N}} \frac{1}{2^{N-1}} W_N\left(y_1^N | u_{1,o}^{2N} \oplus u_{1,e}^{2N}\right). \end{aligned}$$

By carrying out the inner and outer sums in the same manner as in the proof of (22), we obtain (23).

C. Proof of Proposition 4

Let us specify the channels as follows: $W : \mathcal{X} \to \mathcal{Y}$, $W' : \mathcal{X} \to \tilde{Y}$, and $W'' : \mathcal{X} \to \tilde{Y} \times \mathcal{X}$. By hypothesis there is a one-to-one function $f : \mathcal{Y} \to \tilde{\mathcal{Y}}$ such that (17) and (18) are satisfied. For the proof it is helpful to define an ensemble of RVs $(U_1, U_2, X_1, X_2, Y_1, Y_2, \tilde{Y})$ so that the pair (U_1, U_2) is uniformly distributed over $\mathcal{X}^2$, $(X_1, X_2) = (U_1 \oplus U_2, U_2)$, $P_{Y_1,Y_2|X_1,X_2}(y_1, y_2|x_1, x_2) = W(y_1|x_1)W(y_2|x_2)$, and $\tilde{Y} = f(Y_1, Y_2)$. We now have

$$W'(\tilde{y}|u_1) = P_{\tilde{Y}|U_1}(\tilde{y}|u_1)$$
$$W''(\tilde{y}, u_1|u_2) = P_{\tilde{Y}U_1|U_2}(\tilde{y}, u_1|u_2).$$

From these and the fact that $(Y_1, Y_2) \mapsto \tilde{Y}$ is invertible, we get

$$I(W') = I(U_1; \tilde{Y}) = I(U_1; Y_1Y_2)$$
$$I(W'') = I(U_2; \tilde{Y}U_1) = I(U_2; Y_1Y_2U_1).$$

Since U_1 and U_2 are independent, $I(U_2; Y_1Y_2U_1)$ equals $I(U_2; Y_1Y_2|U_1)$. So, by the chain rule, we have

$$I(W') + I(W'') = I(U_1U_2; Y_1Y_2) = I(X_1X_2; Y_1Y_2)$$

where the second equality is due to the one-to-one relationship between (X_1, X_2) and (U_1, U_2). The proof of (24) is completed by noting that $I(X_1X_2; Y_1Y_2)$ equals $I(X_1; Y_1) + I(X_2; Y_2)$ which in turn equals $2I(W)$.

To prove (25), we begin by noting that

$$\begin{aligned} I(W'') &= I(U_2; Y_1Y_2U_1) \\ &= I(U_2; Y_2) + I(U_2; Y_1U_1|Y_2) \\ &= I(W) + I(U_2; Y_1U_1|Y_2). \end{aligned}$$

This shows that $I(W'') \geq I(W)$. This and (24) give (25). The above proof shows that equality holds in (25) iff $I(U_2; Y_1U_1|Y_2) = 0$, which is equivalent to having

$$\begin{aligned} P_{U_1,U_2,Y_1|Y_2}(u_1, u_2, y_1|y_2) \\ = P_{U_1,Y_1|Y_2}(u_1, y_1|y_2) P_{U_2|Y_2}(u_2|y_2) \end{aligned}$$

for all (u_1, u_2, y_1, y_2) such that $P_{Y_2}(y_2) > 0$, or equivalently

$$\begin{aligned} P_{Y_1,Y_2|U_1,U_2}(y_1, y_2|u_1, u_2) P_{Y_2}(y_2) \\ = P_{Y_1,Y_2|U_1}(y_1, y_2|u_1) P_{Y_2|U_2}(y_2|u_2) \end{aligned} \tag{87}$$

for all (u_1, u_2, y_1, y_2). Since $P_{Y_1,Y_2|U_1,U_2}(y_1, y_2|u_1, u_2) = W(y_1|u_1 \oplus u_2)W(y_2|u_2)$, (87) can be written as

$$W(y_2|u_2)[W(y_1|u_1 \oplus u_2)P_{Y_2}(y_2) - P_{Y_1,Y_2}(y_1, y_2|u_1)] = 0. \tag{88}$$

Substituting $P_{Y_2}(y_2) = \frac{1}{2}W(y_2|u_2) + \frac{1}{2}W(y_2|u_2 \oplus 1)$ and

$$P_{Y_1,Y_2|U_1}(y_1, y_2|u_1) = \frac{1}{2}W(y_1|u_1 \oplus u_2)W(y_2|u_2) + \frac{1}{2}W(y_1|u_1 \oplus u_2 \oplus 1)W(y_2|u_2 \oplus 1)$$

into (88) and simplifying, we obtain

$$W(y_2|u_2)W(y_2|u_2 \oplus 1) \cdot [W(y_1|u_1 \oplus u_2) - W(y_1|u_1 \oplus u_2 \oplus 1)] = 0$$

which for all four possible values of (u_1, u_2) is equivalent to

$$W(y_2|0)W(y_2|1)[W(y_1|0) - W(y_1|1)] = 0.$$

Thus, either there exists no y_2 such that $W(y_2|0)W(y_2|1) > 0$, in which case $I(W) = 1$, or for all y_1 we have $W(y_1|0) = W(y_1|1)$, which implies $I(W) = 0$.

D. Proof of Proposition 5

Proof of (26) is straightforward.

$$\begin{aligned} Z(W'') &= \sum_{y_1^2,u_1} \sqrt{W''(f(y_1,y_2),u_1|0)} \cdot \sqrt{W''(f(y_1,y_2),u_1|1)} \\ &= \sum_{y_1^2,u_1} \frac{1}{2}\sqrt{W(y_1|u_1)W(y_2|0)} \cdot \sqrt{W(y_1|u_1 \oplus 1)W(y_2|1)} \\ &= \sum_{y_2} \sqrt{W(y_2|0)W(y_2|1)} \cdot \sum_{u_1} \frac{1}{2} \sum_{y_1} \sqrt{W(y_1|u_1)W(y_1|u_1 \oplus 1)} \\ &= Z(W)^2. \end{aligned}$$

To prove (27), we use shorthand notation $\alpha(y_1) = W(y_1|0)$, $\delta(y_1) = W(y_1|1)$, $\beta(y_2) = W(y_2|0)$, and $\gamma(y_2) = W(y_2|1)$, and write

$$\begin{aligned} Z(W') &= \sum_{y_1^2} \sqrt{W'(f(y_1,y_2)|0)W'(f(y_1,y_2)|1)} \\ &= \sum_{y_1^2} \frac{1}{2}\sqrt{\alpha(y_1)\beta(y_2) + \delta(y_1)\gamma(y_2)} \cdot \sqrt{\alpha(y_1)\gamma(y_2) + \delta(y_1)\beta(y_2)} \\ &\le \sum_{y_1^2} \frac{1}{2}[\sqrt{\alpha(y_1)\beta(y_2)} + \sqrt{\delta(y_1)\gamma(y_2)}] \cdot [\sqrt{\alpha(y_1)\gamma(y_2)} + \sqrt{\delta(y_1)\beta(y_2)}] \\ &\quad - \sum_{y_1^2} \sqrt{\alpha(y_1)\beta(y_2)\delta(y_1)\gamma(y_2)} \end{aligned}$$

where the inequality follows from the identity

$$\begin{aligned} &[\sqrt{(\alpha\beta + \delta\gamma)(\alpha\gamma + \delta\beta)}]^2 \\ &\quad + 2\sqrt{\alpha\beta\delta\gamma}(\sqrt{\alpha} - \sqrt{\delta})^2(\sqrt{\beta} - \sqrt{\gamma})^2 \\ &\quad = [(\sqrt{\alpha\beta} + \sqrt{\delta\gamma})(\sqrt{\alpha\gamma} + \sqrt{\delta\beta}) - 2\sqrt{\alpha\beta\delta\gamma}]^2. \end{aligned}$$

Next, we note that

$$\sum_{y_1^2} \alpha(y_1)\sqrt{\beta(y_2)\gamma(y_2)} = Z(W).$$

Likewise, each term obtained by expanding $(\sqrt{\alpha(y_1)\beta(y_2)} + \sqrt{\delta(y_1)\gamma(y_2)})(\sqrt{\alpha(y_1)\gamma(y_2)} + \sqrt{\delta(y_1)\beta(y_2)})$ gives $Z(W)$ when summed over y_1^2. Also, $\sqrt{\alpha(y_1)\beta(y_2)\delta(y_1)\gamma(y_2)}$ summed over y_1^2 equals $Z(W)^2$. Combining these, we obtain the claim (27). Equality holds in (27) iff, for any choice of y_1^2, one of the following is true: $\alpha(y_1)\beta(y_2)\gamma(y_2)\delta(y_1) = 0$ or $\alpha(y_1) = \delta(y_1)$ or $\beta(y_2) = \gamma(y_2)$. This is satisfied if W is a BEC. Conversely, if we take $y_1 = y_2$, we see that for equality in (27), we must have, for any choice of y_1, either $\alpha(y_1)\delta(y_1) = 0$ or $\alpha(y_1) = \delta(y_1)$; this is equivalent to saying that W is a BEC.

To prove (28), we need the following result which states that the parameter $Z(W)$ is a convex function of the channel transition probabilities.

Lemma 4: Given any collection of B-DMCs $W_j : \mathcal{X} \to \mathcal{Y}$, $j \in \mathcal{J}$, and a probability distribution Q on $\mathcal{J}$, define $W : \mathcal{X} \to \mathcal{Y}$ as the channel $W(y|x) = \sum_{j\in\mathcal{J}} Q(j)W_j(y|x)$. Then

$$\sum_{j\in\mathcal{J}} Q(j)Z(W_j) \le Z(W). \tag{89}$$

Proof: This follows by first rewriting $Z(W)$ in a different form and then applying Minkowsky's inequality [10, p. 524, inequality (h)]

$$\begin{aligned} Z(W) &= \sum_y \sqrt{W(y|0)W(y|1)} \\ &= -1 + \frac{1}{2}\sum_y \left[\sum_x \sqrt{W(y|x)}\right]^2 \\ &\ge -1 + \frac{1}{2}\sum_y \sum_{j\in\mathcal{J}} Q(j) \left[\sum_x \sqrt{W_j(y|x)}\right]^2 \\ &= \sum_{j\in\mathcal{J}} Q(j)\, Z(W_j). \end{aligned}$$

We now write W' as the mixture

$$W'(f(y_1,y_2)|u_1) = \frac{1}{2}\left[W_0\left(y_1^2|u_1\right) + W_1\left(y_1^2|u_1\right)\right]$$

where

$$\begin{aligned} W_0(y_1^2|u_1) &= W(y_1|u_1)W(y_2|0) \\ W_1(y_1^2|u_1) &= W(y_1|u_1 \oplus 1)W(y_2|1) \end{aligned}$$

and apply Lemma 4 to obtain the claimed inequality

$$Z(W') \ge \frac{1}{2}[Z(W_0) + Z(W_1)] = Z(W).$$

Since $0 \leq Z(W) \leq 1$ and $Z(W'') = Z(W)^2$, we have $Z(W) \geq Z(W'')$, with equality iff $Z(W)$ equals 0 or 1. Since $Z(W') \geq Z(W)$, this also shows that $Z(W') = Z(W'')$ iff $Z(W)$ equals 0 or 1. So, by Proposition 1, $Z(W') = Z(W'')$ iff $I(W)$ equals 1 or 0.

E. Proof of Proposition 6

From (17), we have the identities

$$\begin{aligned} &W'(f(y_1,y_2)|0)W'(f(y_1,y_2)|1) \\ &\quad = \frac{1}{4}[W(y_1|0)^2 + W(y_1|1)^2]W(y_2|0)W(y_2|1) \\ &\qquad + \frac{1}{4}[W(y_2|0)^2 + W(y_2|1)^2]W(y_1|0)W(y_1|1) \qquad (90) \\ &W'(f(y_1,y_2)|0) - W'(f(y_1,y_2)|1) \\ &\quad = \frac{1}{2}[W(y_1|0) - W(y_1|1)][W(y_2|0) - W(y_2|1)]. \qquad (91) \end{aligned}$$

Suppose W is a BEC, but W' is not. Then, there exists (y_1, y_2) such that the left sides of (90) and (91) are both different from zero. From (91), we infer that neither y_1 nor y_2 is an erasure symbol for W. But then the RHS of (90) must be zero, which is a contradiction. Thus, W' must be a BEC. From (91), we conclude that $f(y_1, y_2)$ is an erasure symbol for W' iff either y_1 or y_2 is an erasure symbol for W. This shows that the erasure probability for W' is $2\epsilon - \epsilon^2$, where ϵ is the erasure probability of W.

Conversely, suppose W' is a BEC but W is not. Then, there exists y_1 such that $W(y_1|0)W(y_1|1) > 0$ and $W(y_1|0) - W(y_1|1) \neq 0$. By taking $y_2 = y_1$, we see that the RHSs of (90) and (91) can both be made nonzero, which contradicts the assumption that W' is a BEC.

The other claims follow from the identities

$$\begin{aligned} &W''(f(y_1,y_2),u_1|0)W''(f(y_1,y_2),u_1|1) \\ &\quad = \frac{1}{4}W(y_1|u_1)W(y_1|u_1 \oplus 1)W(y_2|0)W(y_2|1) \\ &W''(f(y_1,y_2),u_1|0) - W''(f(y_1,y_2),u_1|1) \\ &\quad = \frac{1}{2}[W(y_1|u_1)W(y_2|0) - W(y_1|u_1 \oplus 1)W(y_2|1)]. \end{aligned}$$

The arguments are similar to the ones already given and we omit the details, other than noting that $(f(y_1, y_2), u_1)$ is an erasure symbol for W'' iff both y_1 and y_2 are erasure symbols for W.

F. Proof of Lemma 1

The proof follows that of a similar result from Chung [9, Theorem 4.1.1]. Fix $\zeta > 0$. Let $\Omega_0 \triangleq \{\omega \in \Omega : \lim_{n\to\infty} Z_n(\omega) = 0\}$. By Proposition 10, $P(\Omega_0) = I_0$. Fix $\omega \in \Omega_0$. $Z_n(\omega) \to 0$ implies that there exists $n_0(\omega, \zeta)$ such that $n \geq n_0(\omega, \zeta) \Rightarrow Z_n(\omega) \leq \zeta$. Thus, $\omega \in \mathcal{T}_m(\zeta)$ for some m. So, $\Omega_0 \subset \bigcup_{m=1}^{\infty} \mathcal{T}_m(\zeta)$. Therefore, $P(\bigcup_{m=1}^{\infty} \mathcal{T}_m(\zeta)) \geq P(\Omega_0)$. Since $\mathcal{T}_m(\zeta) \uparrow \bigcup_{m=1}^{\infty} \mathcal{T}_m(\zeta)$, by the monotone convergence property of a measure, $\lim_{m\to\infty} P[\mathcal{T}_m(\zeta)] = P[\bigcup_{m=1}^{\infty} \mathcal{T}_m(\zeta)]$. So, $\lim_{m\to\infty} P[\mathcal{T}_m(\zeta)] \geq I_0$. It follows that, for any $\zeta > 0$, $\delta > 0$, there exists a finite $m_0 = m_0(\zeta, \delta)$ such that, for all $m \geq m_0$, $P[\mathcal{T}_m(\zeta)] \geq I_0 - \delta/2$. This completes the proof.

REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, Jul.–Oct. 1948.

[2] E. Arıkan, "Channel combining and splitting for cutoff rate improvement," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 628–639, Feb. 2006.

[3] D. E. Muller, "Application of Boolean algebra to switching circuit design and to error correction," *IRE Trans. Electron. Comput.*, vol. EC-3, no. 9, pp. 6–12, Sep. 1954.

[4] I. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *IRE Trans. Inf. Theory*, vol. IT-4, no. 3, pp. 39–44, Sep. 1954.

[5] M. Plotkin, "Binary codes with specified minimum distance," *IRE Trans. Inf. Theory*, vol. IT-6, no. 3, pp. 445–450, Sep. 1960.

[6] S. Lin and D. J. Costello, Jr., *Error Control Coding*, 2nd ed. Upper Saddle River, N.J.: Pearson, 2004.

[7] R. E. Blahut, *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley, 1983.

[8] G. D. Forney, Jr., MIT 6.451 Lecture Notes, Spring, 2005, unpublished.

[9] K. L. Chung, *A Course in Probability Theory*, 2nd ed. New York: Academic, 1974.

[10] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[11] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, 1965.

[12] E. Arıkan and E. Telatar, "On the Rate of Channel Polarization," Aug. 2008, arXiv:0807.3806v2 [cs.IT].

[13] A. Sahai, P. Glover, and E. Telatar, private communication, Oct. 2008.

[14] C. E. Shannon, "A note on partial ordering for communication channels," *Inf. Contr.*, vol. 1, pp. 390–397, 1958.

[15] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. IT-8, no. 1, pp. 21–28, Jan. 1962.

[16] G. D. Forney, Jr., "Codes on graphs: Normal realizations," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.

[17] E. Arıkan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447–449, Jun. 2008.

Erdal Arıkan (S'84–M'79–SM'94) was born in Ankara, Turkey, in 1958. He received the B.S. degree from the California Institute of Technology, Pasadena, in 1981 and the S.M. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, in 1982 and 1985, respectively, all in electrical engineering.

Since 1987 he has been with the Electrical-Electronics Engineering Department of Bilkent University, Ankara, Turkey, where he is presently a Professor.