

UC San Diego

UC San Diego Previously Published Works

Title

Polar Coding for the Binary Erasure Channel With Deletions

Permalink

<https://escholarship.org/uc/item/68p823s6>

Journal

IEEE Communications Letters, 21(4)

ISSN

1089-7798

Authors

Thomas, Eldho K
Tan, Vincent YF
Vardy, Alexander
et al.

Publication Date

2017

DOI

10.1109/lcomm.2017.2650918

Peer reviewed

Polar Coding for the Binary Erasure Channel With Deletions

Eldho K. Thomas, Vincent Y. F. Tan, *Senior Member, IEEE*, Alexander Vardy, *Fellow, IEEE*,
and Mehul Motani, *Senior Member, IEEE*

Abstract—We study the application of polar codes in deletion channels by analyzing the cascade of a binary erasure channel (BEC) and a deletion channel. We show how polar codes can be used effectively on a BEC with a single deletion, and propose a list decoding algorithm with a cyclic redundancy check for this case. The decoding complexity is $O(N^2 \log N)$, where N is the blocklength of the code. An important contribution is an optimization of the amount of redundancy added to minimize the overall error probability. Our theoretical results are corroborated by numerical simulations, which show that the list size can be reduced to one and the original message can be recovered with high probability as the length of the code grows.

Index Terms—Polar codes, deletions, binary erasure channel, cascade, list decoding, cyclic redundancy check, candidate set.

I. INTRODUCTION

POLAR CODES, invented by Arıkan [1], are the first provably capacity-achieving codes with low encoding and decoding complexity. Arıkan’s presentation of polar codes includes a successive cancellation decoding algorithm, which generally does not perform as well as the state-of-the-art error-correcting codes at finite block lengths [2]. To improve the performance of polar codes, Tal and Vardy [3] devised a list decoding algorithm. The initial work of Arıkan considers binary symmetric memoryless channels. There have been attempts to study polar codes for other channels, e.g., the AWGN channel [4]. However, there are not many constructions of polar codes for channels with memory. See [5] and references therein.

The deletion channel is a canonical example of a non-stationary, non-ergodic channel with memory. It deletes symbols arbitrarily and the positions of the deletions are unknown to the receiver. A survey by Mitzenmacher [6] discusses the major developments in the understanding of deletion channels in greater detail. To date, the Shannon capacity of deletion channels, in general, remains unknown. However, there have been attempts to find upper and lower bounds on the capacity of deletion channels [7], [8].

Our motivation is partly the work of Dolecek and Anantharam [9], in which the run length properties of

Reed-Muller (RM) codes were exploited to correct a certain number of substitutions together with a *single* deletion; our work involves correcting *erasures* rather than substitutions. RM codes and polar codes have similar algebraic structures and therefore polar codes are also potential candidates for correcting single deletions. However, they cannot be used directly on deletion channels since the polarization of a channel with memory has not been well-studied. Developing polarization techniques for deletion channels is beyond the scope of this study. Instead, motivated by decoders that are possibly defective and delete symbols arbitrarily, we consider polar codes over a binary erasure channel (BEC) and an adversarial version of the deletion channel with one deletion, and provide a list decoding algorithm to successfully recover the original message with high probability¹ (w.h.p.). Unlike RM codes, polar codes do not have rich run length properties. Instead, we use the successive cancellation algorithm [1] for decoding. In addition, we provide a detailed analysis of the error probability, which was lacking in [9]. Channel cascades were studied previously in [10] but our model has not been previously considered in the literature. We argue that the capacity of the cascade can be achieved; in contrast, [9] does not discuss capacity issues.

II. PRELIMINARIES

A. Polar Codes

We consider polar codes of length $N = 2^n$ constructed recursively from the kernel $G_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$. Given an information vector (message) $u_1^N = (u_1, \dots, u_N)$ where $u_i \in \mathbb{F}_2$, a codeword x_1^N is generated using the relation $x_1^N = u_1^N B_N G_2^{\otimes n}$ where $G_2^{\otimes n}$ is the n -th Kronecker product of G_2 and B_N is a bit-reversal permutation matrix, defined explicitly in [1]. The vector x_1^N is transmitted through N independent copies of a binary discrete memoryless channel (BDMC) $W : \mathbb{F}_2 \rightarrow \mathcal{Y}$ with transition probabilities $\{W(y|x) : x \in \mathbb{F}_2, y \in \mathcal{Y}\}$ and capacity $C(W)$. As n grows, the individual channels start polarizing. That is, a subset of the channels tend to noise-free channels and others tend to completely noisy channels. The fraction of noise-free channels tends to the capacity $C(W)$. The polarization behavior suggests using the noise-free channels to transmit information bits, while setting the inputs to the noisy channels to values that are known *a priori* to the decoder (i.e., the frozen bits). That is, a message vector u_1^N consists of information bits and frozen bits (often set to zero) where $I \subset \{1, \dots, N\} = \mathcal{I}$ of size k is the information set and \bar{I} is the set of frozen bits. This scheme achieves capacity [1]. Denote the channel output by $y_1^N = (y_1, \dots, y_N)$ and the

¹In this letter, we use the term w.h.p. to mean with probability tending to 1 as the blocklength of the code N tends to infinity.

Manuscript received November 7, 2016; revised December 6, 2016 and December 23, 2016; accepted January 6, 2017. Date of publication January 10, 2017; date of current version April 7, 2017. This work is partially funded by a Singapore Ministry of Education (MoE) Tier 2 grant (R-263-000-B61-112). The associate editor coordinating the review of this letter and approving it for publication was C. Jego.

E. K. Thomas is with the Institute of Computer Science, University of Tartu, 51014 Tartu Estonia (e-mail: eldhokt@gmail.com).

V. Y. F. Tan and M. Motani are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: vtan@nus.edu.sg; motani@nus.edu.sg).

A. Vardy is with Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA, and the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371 (e-mail: avardy@ucsd.edu).

Digital Object Identifier 10.1109/LCOMM.2017.2650918

i -th synthesized subchannel with input u_i and output (y_1^N, u_1^{i-1}) by $W_N^{(i)}$ for $i = 1, \dots, N$. The transition probability matrix $W_N^{(i)}$ is defined as

$$W_N^{(i)}(y_1^N, u_1^{i-1} | u_i) := \sum_{u_{i+1}^N \in \mathbb{F}_2^{N-i}} \frac{1}{2^{N-i}} W_N(y_1^N | u_1^N),$$

where $W_N(y_1^N | u_1^N) := \prod_{i=1}^N W(y_i | x_i)$ and $x_1^N = u_1^N B_N G_2^{\otimes n}$ is the codeword corresponding to the message u_1^N . The encoding complexity of polar coding is $O(N \log N)$ [1].

B. Successive Cancellation Decoding

Arikan [1] proposed a successive cancellation (SC) decoding scheme for polar codes. Given y_1^N and the estimates \hat{u}_1^{i-1} of u_1^{i-1} , the SC algorithm estimates u_i . The following logarithmic likelihood ratios (LLR) are used to estimate each u_i for $i = 1, \dots, N$:

$$L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) = \log \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i = 0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i = 1)}.$$

The estimate of an unfrozen bit u_i is determined by the signs of the LLRs, i.e., $\hat{u}_i = 0$ if $L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 0$ and $\hat{u}_i = 1$ otherwise. It is known that polar codes with SC decoding achieve capacity with decoding complexity of $O(N \log N)$ [1].

C. Adversarial Deletion Channel

We suppose that N bits are sent over a channel and exactly d bits are deleted. We call this a d -deletion channel. That is, for N bits sent, the decoder only receives $N - d$ bits after d deletions and the positions of deletions are not known to the receiver. Note that this is not the probabilistic deletion channel in which each symbol is independently deleted with some fixed probability $q \in (0, 1)$ [8].

III. PROBLEM SETTING AND MODEL

Consider the 1-deletion channel ($d = 1$ in the definition in Section II-C), where exactly one bit is deleted. We suppose that $N = 2^n$ where $n \in \mathbb{N}$. A message vector u_1^N is encoded using the polar encoder and is sent across N uses of a BEC $W_1^N = \mathbf{W}_1$, each with erasure probability $p \in (0, 1)$. The output vector is passed through a 1-deletion channel \mathbf{W}_2 . We denote this cascade of \mathbf{W}_1 and \mathbf{W}_2 as \mathbf{W} and call this a *BEC-1-Deletion Cascade*. This model is shown in Fig. 1. The output of \mathbf{W} is denoted as \tilde{y}_1^{N-1} . Note that \mathbf{W} permits erasures and a single deletion. That is, a message u_1^N is sent across \mathbf{W} and a vector \tilde{y}_1^{N-1} is received. A decoder is designed in such a way that w.h.p., a list \mathcal{L} (of linear size in N) containing an estimate \hat{u}_1^N of the original message u_1^N is returned.

IV. CODING FOR THE BEC-1-DELETION CASCADE

A. Reconstruction of the BEC Output

A message u_1^N is sent over a BEC-1-Deletion cascade using a polar encoder described in Section II-A and \tilde{y}_1^{N-1} is received. In order to decode \tilde{y}_1^{N-1} , we use the SC algorithm (refer to Section II-B). Since the position of the deletion is unknown, we first identify a set of vectors, called the *candidate set*, which contains \tilde{y}_1^{N-1} as a sub-sequence. A naïve algorithm

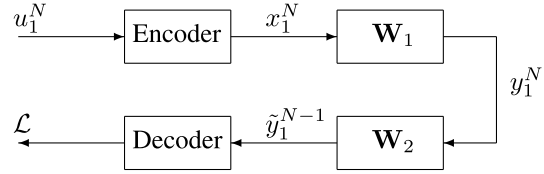


Fig. 1. BEC-1-Deletion Cascade. $\mathbf{W}_1 = \text{BEC}(p)^N$ is the length- N BEC, \mathbf{W}_2 is the 1-deletion channel, and \mathcal{L} is the list of possible messages.

to construct the candidate set would be to insert 0, 1, e in the N locations before and after each symbol of \tilde{y}_1^{N-1} . We then apply the SC algorithm to each vector in the candidate set.

For example, suppose $N = 4$ and the received vector is $\tilde{y}_1^3 = 01e$. Then the following set \mathcal{S} includes all vectors which contain the subsequence 01e:

$$\mathcal{S} = \{001e, 101e, e01e, 011e, 0e1e, 010e, 01ee, 01e0, 01e1\}$$

The size of this set can be further reduced if we notice that inserting e at N positions is enough to identify all possible messages those can output \tilde{y}_1^{N-1} after a single deletion. This is because of the following: Suppose the i -th symbol is deleted from y_1^N . Instead of inserting 0 or 1 at position i , we insert an erasure symbol e. Since a polar code correcting $\alpha \approx Np'$ (where $p' < p$) erasures also corrects $\alpha + 1$ erasures w.h.p., under the SC decoding algorithm, this new length- N vector decodes to the correct message w.h.p. no matter which symbol was at position i . We state this observation formally:

Proposition 1: Suppose u_1^N is sent over a BEC-1-Deletion cascade \mathbf{W} . (See Fig. 1.) The size of the candidate set \mathcal{A} (constructed above) is $N - \alpha$ where α is the number of erasures present in the received string \tilde{y}_1^{N-1} .

Proof: The candidate set is

$$\mathcal{A} = \{(\tilde{y}_1^{i-1}, e, \tilde{y}_i^{N-1}) : i = 1, 2, \dots, N\} \subset \{0, 1, e\}^N$$

where \tilde{y}_1^{N-1} is the received string. Suppose that the j -th symbol of \tilde{y}_1^{N-1} is e. Inserting another e before the j -th symbol e forms vector $\tilde{y}_1^{j-1} e e \tilde{y}_{j+1}^{N-1}$. This vector repeats if we insert e again after the the j -th symbol e. Therefore, considering non-erasure bits of \tilde{y}_1^{N-1} and inserting exactly one erasure symbol e at positions before and after these non-erasure bits produces unique vectors in the candidate set \mathcal{A} . Since the number of erasure symbols is α , the total number of vectors in \mathcal{A} is $N - \alpha$. ■

We remark that as $N \rightarrow \infty$, by the law of large numbers $\frac{\alpha}{N} \rightarrow p$ and hence $|\mathcal{A}| \approx N - Np$ where $p \in (0, 1)$ is the erasure probability of the BEC.

B. List Decoding

After the construction of the set \mathcal{A} , the problem reduces to the decoding of each vector in \mathcal{A} using the SC algorithm. Since $|\mathcal{A}| = N - \alpha$, we get a list of messages of size at most $N - \alpha$ at the end of the whole decoding procedure.

Let $\text{SC}(y_1^N)$ denote the SC decoding of y_1^N , and define

$$\mathcal{L} = \{u_1^k : u_1^k = \hat{u}_1^N |_I, \hat{u}_1^N = \text{SC}(y_1^N), y_1^N \in \mathcal{A}\}, \quad (1)$$

as the list of messages returned by the set \mathcal{A} where I is the information set.

Since we insert the erasure symbol e at each of the N possible positions (including the deleted position), the original message sent belongs to \mathcal{L} w.h.p. Arıkan [1] proved that the probability of error $P_e^{(N)}$ vanishes asymptotically for polar codes over any BDMC. A more precise estimate was provided by Arıkan and Telatar [11] who showed that for any $\beta \in (0, 1/2)$, $P_e^{(N)} \leq 2^{-N^\beta}$ for sufficiently large block lengths N . Therefore, under SC decoding, vectors in \mathcal{A} return all possible messages that can produce the string \tilde{y}_1^{N-1} under a single (adversarial) deletion.

C. Recovering the Correct Message From the List via Cyclic Redundancy Check (CRC)

Naturally, there can be multiple $u_1^k \in \mathcal{M}$ that belong to the list \mathcal{L} and it may not be easy to single out the original message. However, by applying a simple pre-coding technique using an r -bit CRC (or a code having an $r \times k$ parity check matrix) [3], [12], the original message can be detected from the list, albeit with some additional probability of error. We describe how to recover the correct message w.h.p. here.

Recall that we have $N - k$ frozen bits that we usually set to zero. Instead of setting all of them to zero, we set $N - k - r$ frozen bits to zero, where r is a small number we optimize in Section IV-D. These r bits will contain the r -bit CRC value of the k unfrozen bits (or simply the parity bits). To generate a r -bit CRC, we select a polynomial of degree r , called a *CRC polynomial*, having $r + 1$ coefficients. We then divide the message (by treating it as a binary polynomial) by this CRC polynomial to generate a remainder of degree at most $r - 1$, with total number of coefficients r . We append these r coefficients at the end of the k -bit message to generate a $(k + r)$ -bit vector. To verify that the correct message is received, we perform the polynomial division again to check if the remainder is zero. For more details on the choice of CRC polynomials, please refer to [13]. We send these $k + r$ bits across the cascade. This new encoding is a slight variation the original polar coding scheme [1]. Also, note that the original information rate $R = \frac{k}{N}$ is preserved. However, the rate of the polar code is slightly increased to $R_{\text{polar}} = \frac{k+r}{N}$.

To summarize, we encode the message u_1^k of length k into a length $k + r$ vector $u_1^{k+r} \in \mathcal{C}'$ having redundancy r where $|\mathcal{C}'| = 2^k$. Then we apply the polar coding scheme for the codebook \mathcal{C}' . This will result in a polar code \mathcal{C} of length N and size 2^{k+r} where only the subset $\mathcal{C}' \subset \mathcal{C}$ carries information that we wish to transmit. The codeword $x_1^N \in \mathcal{C}$ corresponding to the original message u_1^k is then passed through the BEC-1-Deletion channel and outputs a vector \hat{y}_1^{N-1} . After constructing the set \mathcal{A} by inserting e at each possible N positions, we apply the SC algorithm on \mathcal{A} . However, not all of these resulting vectors in \mathcal{C} carry information. We can check this using the initial r -bit CRC (or the parity check matrix). All vectors which fail under the CRC check are removed and we then select the message with the maximum likelihood from the list.

D. Analysis and Optimization of the Overall Error Probability

Suppose H denotes the $r \times (k + r)$ parity check matrix with rows $\{h_i : i = 1, \dots, r\}$ that is being used for adding parity to

the k bit message. Then the set of messages that carries any information can be identified as

$$\widehat{\mathcal{M}} := \{u_1^k : u_1^{k+r} H^T = 0, u_1^{k+r} \in \widehat{\mathcal{L}}\},$$

where $\widehat{\mathcal{L}}$ is the modified version of (1) according to the new polar coding scheme defined as

$$\widehat{\mathcal{L}} := \{u_1^{k+r} : u_1^{k+r} = \hat{u}_1^N|_{\mathcal{I} \cup \mathcal{P}}, \hat{u}_1^N = \text{SC}(y_1^N), y_1^N \in \mathcal{A}\},$$

and where $\mathcal{P} \subset \bar{\mathcal{I}}$ is the set of parity bits ($\bar{\mathcal{I}}$ is the set of frozen bits). If the rows of H are chosen uniformly and independently from $\{0, 1\}^{k+r}$, the probability that a vector u_1^k is in $\widehat{\mathcal{M}}$ is

$$\Pr(u_1^k \in \widehat{\mathcal{M}}) = \Pr\left(\langle h_i, u_1^{k+r} \rangle = 0, \forall i = 1, \dots, r\right) = \frac{1}{2^r}$$

where $u_1^{k+r} \in \widehat{\mathcal{L}}$. That is, a message in $\widehat{\mathcal{L}}$ is wrongly identified as the original message with probability $1/2^r$. However, the true message sent satisfies the parity-check condition $u_1^{k+r} H^T = 0$. Therefore, by the union bound, the total probability that an incorrect message is returned is upper bounded as

$$P_{\text{TotErr}}^{(N)} \leq \frac{|\widehat{\mathcal{L}}|}{2^r} + |\mathcal{A}| P_e^{(N)}, \quad (2)$$

where $P_e^{(N)}$ is the probability of error of the SC decoding algorithm and $|\widehat{\mathcal{L}}| \leq |\mathcal{A}| \approx N(1 - p)$ for a single deletion.

To maintain that $R_{\text{polar}} \approx R$ (that is, as the block length N grows, R_{polar} converges to R) and the upper bound on $P_{\text{TotErr}}^{(N)}$ in (2) is minimized, we have to choose r carefully.

For a single deletion, the size of the candidate set $|\mathcal{A}| \approx N(1 - p)$ and hence $|\widehat{\mathcal{L}}| \leq N(1 - p)$ w.h.p. From Hassani *et al.* [14], the rate-dependent error probability of the polar code for the BEC with rate R_{polar} is

$$P_e^{(N)} = 2^{-2^{\frac{n}{2} + \frac{\sqrt{n}}{2}} Q^{-1}\left(\frac{R_{\text{polar}}}{C(\mathbf{W})}\right) + o(\sqrt{n})}$$

where $N = 2^n$, $Q(x) := \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-\frac{t^2}{2}) dt$ is the complementary Gaussian cumulative distribution function, and $C(\mathbf{W})$ is the capacity of the channel cascade.

From (2),

$$P_{\text{TotErr}}^{(N)} \leq |\mathcal{A}| \left[2^{-r} + 2^{-2^{\frac{n}{2} + \frac{\sqrt{n}}{2}} Q^{-1}\left(\frac{R_{\text{polar}}}{C(\mathbf{W})}\right) + o(\sqrt{n})} \right]. \quad (3)$$

It can be verified easily that the first term in the square parentheses in (3) is decreasing and the second term with $R_{\text{polar}} = \frac{k+r}{N}$ is increasing in r . To optimize the upper bound in (3), we set the exponents of two terms to be equal (neglecting the insignificant $o(\sqrt{n})$ term), i.e.,

$$r = 2^{\frac{n}{2} + \frac{\sqrt{n}}{2}} Q^{-1}\left(\frac{k+r}{NC(\mathbf{W})}\right) = \sqrt{N} 2^{\frac{\sqrt{\log_2 N}}{2}} Q^{-1}\left(\frac{k+r}{NC(\mathbf{W})}\right),$$

where we used the fact that $N = 2^n$.

Now we find an expression for r in terms of the backoff from capacity. To transmit the code at a rate close to the capacity, for a small constant $\delta > 0$, assume that $R = (1 - \delta)(1 - p)$ where $C(\mathbf{W}) = 1 - p$ since a polar code over the BEC 1-deletion cascade achieves the capacity of the BEC; this is a simple consequence of [15, Problem 3.14] and the fact that the list size is polynomial. Then the rate

$R_{\text{polar}} = R + \frac{r}{N}(1-p) \geq (1 - \frac{\delta}{2})(1-p)$ for N large enough. Therefore,

$$r = \sqrt{N} \cdot 2^{\frac{\sqrt{\log_2 N}}{2}} Q^{-1}\left(1 - \frac{\delta}{2}\right).$$

Let $z = Q^{-1}\left(1 - \frac{\delta}{2}\right)$. Since $\frac{\delta}{2} \approx 0$, $z \ll 0$. Then $Q(z) = 1 - \frac{\delta}{2}$ and hence $\frac{\delta}{2} = Q(-z)$. Since $Q(-z)$ decays as $e^{-z^2/2}$ as $z \rightarrow -\infty$, $z^2 = 2 \ln \frac{2}{\delta}$. Then $z = -\sqrt{2 \ln \frac{2}{\delta}}$. Therefore, the optimal value of the number of parity bits r is

$$r = \sqrt{N} \cdot 2^{-\sqrt{\frac{(\log_2 N)(\ln \frac{2}{\delta})}{2}}} = \Theta(\sqrt{N}).$$

This is a rate-dependent choice of r (through δ) that simultaneously ensures that $R_{\text{polar}} \rightarrow R$ and the upper bound on $P_{\text{TotErr}}^{(N)}$ in (2) is minimized.

E. Finite Number of Deletions

Now consider the cascade of a BEC and a d -deletion channel where $d \in \mathbb{N}$ is finite. This model can be analyzed using the same techniques presented here. The only difference is the size of the candidate set \mathcal{A} . By using the same arguments as in the 1-deletion case, we construct \mathcal{A} by inserting erasure symbols at d positions and $|\mathcal{A}| = \binom{N}{d} - \alpha$. Therefore, the list size $|\hat{\mathcal{L}}| \leq \binom{N}{d} - \alpha$. Since the models are similar, a CRC construction and error probability analysis for the BEC- d -Deletion cascade similar to that presented in Sections IV-C and IV-D respectively can be performed. In addition, we see that even if the list size is $d = o\left(\frac{N}{\log N}\right)$, the capacity of the BEC is achieved because $|\hat{\mathcal{L}}| \leq N^d$ is still subexponential.

F. Complexity of the Decoding Algorithm

The encoding complexity of the BEC-1-Deletion cascade is same as that for standard polar codes, i.e., $O(N \log N)$. However, the SC decoding algorithm has to be applied to all vectors in the candidate set \mathcal{A} of size $N - \alpha$ (cf. Prop. 1). Thus, the complexity of the decoding algorithm of the BEC-1-Deletion cascade is $O(N^2 \log N)$ and that for the BEC- d -Deletion cascade is $O(N^{d+1} \log N)$. Although the complexity of the decoding algorithm increases by $O(N)$ for each additional deletion, it can still be performed in polynomial time.

V. SIMULATION RESULTS

In this section, we demonstrate the utility of the proposed algorithm by performing numerical simulations. The simulations are carried out in MATLAB using code provided in [16] with the following parameters. Let $n = \log_2 N$ vary from 6 to 11. The erasure probability of the BEC is $p = 0.3$. Thus, the capacity of the cascade is $C(\mathbf{W}) = 0.7$. We consider three different code rates: $R = 0.50, 0.55$ and 0.60 . We fix $r = \lceil 0.7\sqrt{N} \rceil$ and the r -bit CRC polynomial is chosen according to [13]. The error probability is computed by averaging over 1000 independent runs.

We encode a random length- $\lceil RN \rceil$ message using a r -bit CRC polynomial so that the input of the encoder is a $k + r$ length input vector and the output is an N -bit vector. This vector is then transmitted through a BEC-1-deletion cascade

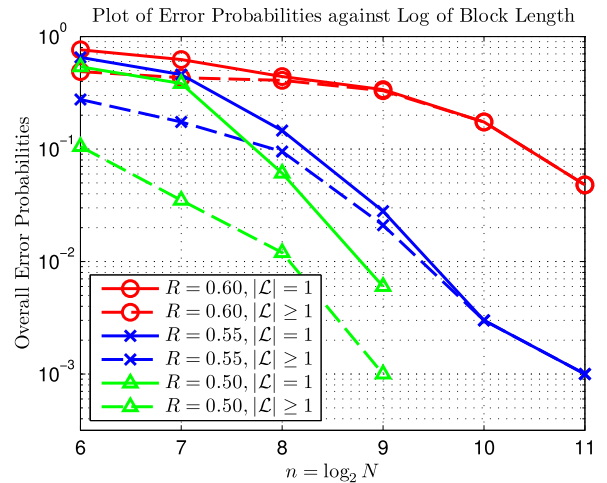


Fig. 2. Plot of error probabilities against $n = \log_2 N \in \{6, \dots, 11\}$. The solid line is the error probability of obtaining a list of size 1, which is exactly the original message. The broken line is the error probability of obtaining a list of size at least 1 containing the original message. The list size after using CRC is small even though it is not exactly 1. Data points that are not available indicate that the simulated error probability over 1000 runs is exactly 0.

and received a length- $(N - 1)$ vector. The CRC list decoder then computes a list of possible messages given the channel output. Fig. 2 shows that, with a suitable choice of the number of CRC bits r and CRC polynomials, as N grows, the list is of size 1 and contains only the original message w.h.p.

REFERENCES

- [1] E. Arkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.
- [2] S. H. Hassani, K. Alishahi, and R. L. Urbanke, "Finite-length scaling for polar codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5875–5898, Oct. 2014.
- [3] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.
- [4] E. Abbe and A. Barron, "Polar coding schemes for the AWGN channel," in *Proc. ISIT*, Jul. 2011, pp. 194–198.
- [5] R. Wang, J. Honda, H. Yamamoto, Y. Hou, and R. Liu, "Construction of polar codes for channels with memory," in *Proc. ITW*, Jeju Island, South Korea, Oct. 2015, pp. 187–191.
- [6] M. Mitzenmacher, "A survey of results for deletion channels and related synchronization channels," *Probab. Surv.*, vol. 6, pp. 1–33, 2009.
- [7] R. Venkataramanan, S. Tatikonda, and K. Ramchandran, "Achievable rates for channels with deletions and insertions," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 6990–7013, Nov. 2013.
- [8] S. Diggavi, M. Mitzenmacher, and H. D. Pfister, "Capacity upper bounds for the deletion channel," in *Proc. ISIT*, 2007, pp. 1716–1720.
- [9] L. Dolecek and V. Anantharam, "Using Reed–Müller $RM(1, m)$ codes over channels with synchronization and substitution errors," *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1430–1443, Apr. 2007.
- [10] A. B. Kiely and J. T. Coffey, "On the capacity of a cascade of channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1310–1321, Jul. 1993.
- [11] E. Arkan and I. E. Telatar, "On the rate of channel polarization," in *Proc. ISIT*, Jun. 2009, pp. 1493–1495.
- [12] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, Oct. 2012.
- [13] P. Koopman and T. Chakravarty, "Cyclic redundancy code (CRC) polynomial selection for embedded networks," in *Proc. Int. Conf. Dependable Syst. Netw.*, Jun. 2004, pp. 145–154.
- [14] S. H. Hassani, R. Mori, T. Tanaka, and R. L. Urbanke, "Rate-dependent analysis of the asymptotic behavior of channel polarization," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2267–2276, Apr. 2013.
- [15] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [16] H. Vangala, Y. Hong, and E. Viterbo, "Efficient algorithms for systematic polar encoding," *IEEE Commun. Lett.*, vol. 20, no. 1, pp. 17–20, Jan. 2016.