Parallelism versus Latency in Simplified Successive-Cancellation Decoding of Polar Codes

Seyyed Ali Hashemi, Marco Mondelli, Arman Fazeli, Alexander Vardy,
John Cioffi, and Andrea Goldsmith

Abstract

This paper characterizes the latency of the simplified successive-cancellation (SSC) decoding scheme for polar codes under hardware resource constraints. In particular, when the number of processing elements P that can perform SSC decoding operations in parallel is limited, as is the case in practice, the latency of SSC decoding is $O\left(N^{1-1/\mu} + \frac{N}{P}\log_2\log_2\frac{N}{P}\right)$, where N is the block length of the code and μ is the scaling exponent of the channel. Three direct consequences of this bound are presented. First, in a fully-parallel implementation where $P = \frac{N}{2}$, the latency of SSC decoding is $O\left(N^{1-1/\mu}\right)$, which is sublinear in the block length. This recovers a result from our earlier work. Second, in a fully-serial implementation where $P = 1$, the latency of SSC decoding scales as $O\left(N\log_2\log_2 N\right)$. The multiplicative constant is also calculated: we show that the latency of SSC decoding when $P = 1$ is given by $(2 + o(1))\, N\log_2\log_2 N$. Third, in a semi-parallel implementation, the smallest P that gives the same latency as that of the fully-parallel implementation is $P = N^{1/\mu}$. The tightness of our bound on SSC decoding latency and the applicability of the foregoing results is validated through extensive simulations.

I. Introduction

Polar codes [1] have been adopted as the coding scheme for control and physical broadcast channels of the enhanced mobile broadband (eMBB) mode and the ultra-reliable low latency communications (URLLC) mode in the fifth generation (5G) wireless communications standard

S. A. Hashemi and J. Cioffi are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA (email: ahashemi@stanford.edu, cioffi@stanford.edu). M. Mondelli is with the Institute of Science and Technology (IST) Austria, Klosterneuburg, Austria (email: marco.mondelli@ist.ac.at). A. Fazeli and A. Vardy are with the Department of Electrical and Computer Engineering, UC San Diego, La Jolla, CA 92093, USA (email: afazelic@ucsd.edu, avardy@ucsd.edu). A. Goldsmith is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA (email: goldsmith@princeton.edu).

[2], [3]. For a polar code of block length N, the encoding and successive-cancellation (SC) decoding complexity for any binary memoryless symmetric (BMS) channel is $O\left(N \log_2 N\right)$. Polar codes can be constructed with complexity that is sublinear in N [4], and the error probability under SC decoding scales with the block length roughly as $2^{-\sqrt{N}}$ [5]. The gap to capacity scales with the block length roughly as

$$I(W) - R \ \sim \ N^{-1/\mu} \ , \tag{1}$$

where W is the BMS transmission channel, $I(W)$ is its capacity, R is the rate of the code, and μ is called the *scaling exponent* (see [6], [7], [8], [9], [10], [11], [12]). In general, the scaling exponent μ depends on the transmission channel W. It is known [6], [7] that for conventional polar codes, $3.579 \leq \mu \leq 4.714$ for any BMS channel W. Furthermore, $\mu \approx 3.63$ when W is a binary erasure channel (BEC), as shown in [7], $\mu \approx 4$ when W is a binary additive white Gaussian noise channel (BAWGNC), as shown in [13], and it is conjectured that $\mu \approx 4.2$ when W is a binary symmetric channel (BSC). It is possible to approach the optimal scaling exponent $\mu = 2$ for any BMS channel by using large polarization kernels [11], [12], [14]. The moderate deviations regime, in which both the error probability and the gap to capacity jointly vanish as the block length grows, has also been a subject of recent investigation [7], [15], [16], [17].

For practical block lengths, polar codes' error-correction performance under SC decoding is not satisfactory. Therefore, an SC list (SCL) decoder with time complexity $O\left(LN \log_2 N\right)$ and space complexity $O\left(LN\right)$ is used [18], where L is the size of the list. SCL decoding runs L coupled SC decoders in parallel and maintains a list of the most likely codewords. The SCL decoder's empirical performance is close to that of the optimal MAP decoder with practical list-size L. Furthermore, by adding some extra bits of cyclic redundancy check (CRC) precoding, the performance is comparable to state-of-the-art low-density parity-check (LDPC) codes.

SC-based decoding algorithms, such as SC and SCL decoding, suffer from high latency. This is due to the fact that SC decoding is inherently a serial algorithm: it proceeds sequentially bit by bit. In order to mitigate this issue, a *simplified SC* (SSC) decoder was proposed in [19]. The SSC decoder identifies two specific constituent codes in the polar code, namely, constituent codes of rate 0 (Rate-0) and rate 1 (Rate-1). The bits within each constituent code can be decoded in parallel; thus, these constituent codes are decoded in one shot. Consequently, the latency is reduced without increasing the error probability. In [20], [21], [22], more constituent codes were identified and low-complexity parallel decoders were designed, increasing the throughput and

reducing the latency even further. These results were extended to SCL decoders in [23], [24], [25]. Recently, it was shown in [26] that the latency of the SSC decoder proposed in [19] is $O\left(N^{1-1/\mu}\right)$. Thus the latency of SSC decoding is *sublinear in* N, in contrast to the $O(N)$ latency of standard SC decoding [1]. However, these results are based on the assumption that the hardware resources are unlimited, and thus a fully-parallel architecture can be implemented. In a practical application, this assumption is no longer valid and a specific number of processing elements (PEs) P are allocated to perform the operations in SC-based decoding algorithms [27]. In the extreme case where $P = 1$ (a fully-serial architecture), the latency of SC decoding grows from $O(N)$ to $O(N \log_2 N)$.

This paper quantifies the latency of the SSC decoder proposed in [19] as a function of hardware resource constraints. Our main result is that the latency of SSC decoding scales as

$$O\left(N^{1-1/\mu} + \frac{N}{P}\log_2\log_2\frac{N}{P}\right) \tag{2}$$

with the block length N. Several consequences of the bound in (2) are as follows. In a fully-parallel implementation, where $P = \frac{N}{2}$, this bound reduces to $O\left(N^{1-1/\mu}\right)$, thereby recovering the main result of [26]. In a fully-serial implementation, where $P = 1$, the bound in (2) reduces to $O(N \log_2\log_2 N)$. This aligns with the results of [28], wherein a variant of polar codes with log-logarithmic complexity per information bit has been introduced. However, this paper's analysis is for *conventional* polar codes rather than a variant thereof. Moreover, for the case where $P = 1$, we determine the multiplicative constant in our bound and further refine it to $(2 + o(1))\, N \log_2\log_2 N$. Finally, it is shown that $P = N^{\frac{1}{\mu}}$ is the smallest number of processing elements that, asymptotically, provides the same latency as that of the fully-parallel decoder. The applicability of the foregoing results is validated through extensive simulations. Our numerical results confirm the presented bounds' tightness.

The rest of this paper is organized as follows: Section II explains polar codes and discusses SC and SSC decoding algorithms with limited number of PEs; Section III states and proves that in an implementation of the SSC decoder with P processing elements, the latency is upper bounded by $O\left(N^{1-1/\mu} + \frac{N}{P}\log_2\log_2\frac{N}{P}\right)$; numerical results are presented in Section IV to verify the proposed bounds; and conclusions are drawn in Section V.

II. Polar Coding Preliminaries

A. Polar Codes

Consider a BMS channel $W : \mathcal{X} \to \mathcal{Y}$ defined by transition probabilities $\{W(y \mid x) : x \in \mathcal{X}, y \in \mathcal{Y}\}$, where $\mathcal{X} = \{0, 1\}$ is the input alphabet and $\mathcal{Y}$ is an arbitrary output alphabet. The reliability of the channel W can be measured by its Bhattacharyya parameter $Z(W) = \sum_{y\in\mathcal{Y}} \sqrt{W(y \mid 0)W(y \mid 1)}$. Channel polarization [1] is the process of mapping two copies of the channel W into two synthetic channels $W^0 : \mathcal{X} \to \mathcal{Y}^2$ and $W^1 : \mathcal{X} \to \mathcal{X} \times \mathcal{Y}^2$ as

$$\begin{aligned} W^0(y_1, y_2 \mid x_1) &= \sum_{x_2\in\mathcal{X}} \frac{1}{2} W(y_1 \mid x_1 \oplus x_2)W(y_2 \mid x_2), \\ W^1(y_1, y_2, x_1 \mid x_2) &= \frac{1}{2} W(y_1 \mid x_1 \oplus x_2)W(y_2 \mid x_2), \end{aligned} \tag{3}$$

where W^0 is a *worse* channel and W^1 is a *better* channel than W because [1], [29]

$$Z(W)\sqrt{2 - Z(W)^2} \leq Z(W^0) \leq 2Z(W) - Z(W)^2, \tag{4}$$

$$Z(W^1) = Z(W)^2. \tag{5}$$

By recursively performing the operation in (3) n times, 2^n copies of W are transformed into 2^n synthetic channels $W_n^{(i)} = (((W^{b_1^{(i)}})^{b_2^{(i)}})^{\cdots})^{b_n^{(i)}}$, where $1 \leq i \leq 2^n$ and $(b_1^{(i)}, \ldots, b_n^{(i)})$ is the binary representation of the integer $i-1$ over n bits. Consider a random sequence of channels, defined recursively as

$$W_n = \begin{cases} W_{n-1}^0, & \text{w.p. } 1/2, \\ W_{n-1}^1, & \text{w.p. } 1/2, \end{cases} \tag{6}$$

where $W_0 = W$. Using (4) and (5), the random process that tracks the Bhattacharyya parameter of W_n can be represented as

$$Z_n \begin{cases} \in \left[Z_{n-1}\sqrt{2 - Z_{n-1}^2},\, 2Z_{n-1} - Z_{n-1}^2\right], & \text{w.p. } 1/2, \\ = Z_{n-1}^2, & \text{w.p. } 1/2, \end{cases} \tag{7}$$

where $Z_n = Z(W_n)$ and $n \geq 1$.

The construction of polar codes comprises the assigning of information bits to the set of positions with the best Bhattacharyya parameters, as stated in the following definition.

Definition 1 (Polar code construction)**:** For a given block length $N = 2^n$, BMS channel W, and probability of error $p_e \in (0, 1)$, the polar code $\mathcal{C}_{\text{polar}}(p_e, W, N)$ is constructed by assigning the information bits to the positions corresponding to all the synthetic channels whose Bhattacharyya

parameter is less than p_e/N and by assigning a predefined (frozen) value to the remaining positions.

With the construction rule of Definition 1, the error probability under SC decoding is guaranteed to be *at most* p_e. Moreover, this construction rule ensures that the rate R of the code tends to capacity at a speed that is captured by the *scaling exponent* of the channel.

Definition 2 (Upper bound on scaling exponent)**:** We say that μ is an upper bound on the scaling exponent if there exists a function $h(x) : [0, 1] \to [0, 1]$ such that $h(0) = h(1) = 0$, $h(x) > 0$ for any $x \in (0, 1)$, and

$$\sup_{\substack{x\in(0,1)\\ y\in[x\sqrt{2-x^2},2x-x^2]}} \frac{h(x^2)+h(y)}{2h(x)} < 2^{-1/\mu}. \tag{8}$$

By defining the scaling exponent as in Definition 2, the gap to capacity $I(W) - R$ scales as $O(N^{-1/\mu})$, see Theorem 1 of [7]. Note that $\mu \approx 4$ for BAWGNC as shown in [13], and it is conjectured that $\mu \approx 4.2$ for BSC. For the BEC, the condition (8) can be relaxed to

$$\sup_{x\in(0,1)} \frac{h(x^2)+h(2x-x^2)}{2h(x)} < 2^{-1/\mu}, \tag{9}$$

which gives a numerical value $\mu \approx 3.63$.

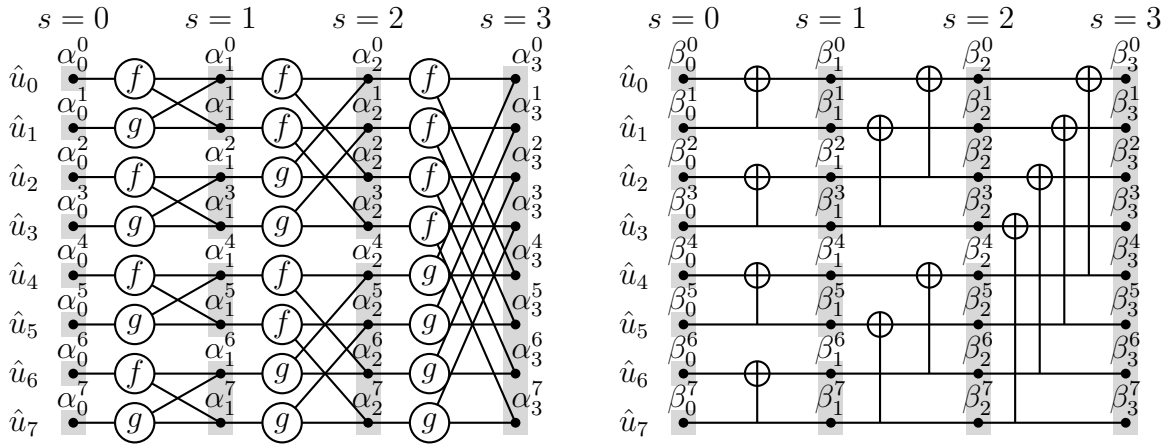
B. Successive-Cancellation Decoding

SC decoding is a message passing algorithm on the factor graph of polar codes, as shown in Fig. 1 for a polar code of length $N = 8$. At level n of the factor graph, the LLR values $\boldsymbol{\alpha}_n^{0:N-1} = \{\alpha_n^0, \alpha_n^1, \ldots, \alpha_n^{N-1}\}$, that are calculated from the received channel-output vector, are fed to the decoder. Fig. 1a shows how the vector of internal LLR values, $\boldsymbol{\alpha}_s^{0:N-1} = \{\alpha_s^0, \alpha_s^1, \ldots, \alpha_s^{N-1}\}$, which is composed of $\frac{N}{2^s}$ vectors of 2^s LLR values $\boldsymbol{\alpha}_s^{i2^s:(i+1)2^s-1} = \{\alpha_s^{i2^s}, \alpha_s^{i2^s+1}, \ldots, \alpha_s^{(i+1)2^s-1}\}$, is generated. Specifically, at each level s, we have:

$$\alpha_s^i = \begin{cases} f(\alpha_{s+1}^i, \alpha_{s+1}^{i+2^s}) & \text{if } \lfloor \frac{i}{2^s} \rfloor \bmod 2 = 0, \\ g(\alpha_{s+1}^i, \alpha_{s+1}^{i-2^s}, \beta_s^{i-2^s}) & \text{if } \lfloor \frac{i}{2^s} \rfloor \bmod 2 = 1, \end{cases} \tag{10}$$

where $f(a, b) = 2\operatorname{arctanh}\left(\tanh\left(\frac{a}{2}\right)\tanh\left(\frac{b}{2}\right)\right)$, $g(a, b, c) = a + (1 - 2c)b$, and β_s^i is the i-th bit estimate at level s of the factor graph. As shown in Fig. 1b, the bit estimates $\boldsymbol{\beta}_s = \{\beta_s^0, \beta_s^1, \ldots, \beta_s^{N-1}\}$ are calculated as

$$\beta_s^i = \begin{cases} \beta_{s-1}^i \oplus \beta_{s-1}^{i+2^s} & \text{if } \lfloor \frac{i}{2^s} \rfloor \bmod 2 = 0, \\ \beta_{s-1}^i & \text{if } \lfloor \frac{i}{2^s} \rfloor \bmod 2 = 1, \end{cases} \tag{11}$$



(a) Generation of LLR values.

(b) Generation of bit estimates.

Fig. 1: SC decoding on the factor graph representation of polar codes with $N = 8$. Each gray area represents one node in the binary tree representation of SC decoding.

where $\oplus$ is the bit-wise XOR operation. All frozen bits are assumed to be zero. Hence at level $s = 0$, the i-th bit $\hat{u}_i$ is estimated as

$$\hat{u}_i = \beta_0^i = \begin{cases} 0 & \text{if } u_i \text{ is a frozen bit or } \alpha_0^i > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{12}$$

By combining all the operations in (10) that can be performed in parallel, SC decoding can be represented as on Fig. 2's binary tree. Fig. 2's root node at decoding level n is fed with the LLR values, and the results of operations in (10) and (11) are passed on the branches of the decoding tree. SC decoding has a sequential structure in the sense that the decoding of each bit depends on the decoding of its previous bits. More formally, on the one hand, when $\mod(\frac{i}{2^s}, 2) = 0$, the calculation of α_s^i at level s is only dependent on the LLR values that are received from a node at level $s + 1$. On the other hand, when $\mod(\frac{i}{2^s}, 2) = 1$, the calculation of α_s^i also depends on a hard bit estimation $\beta_s^{i-2^s}$ that is a result of estimating the previous bits (see (10)). Consequently, SC decoding proceeds by traversing the binary tree such that the nodes at level $s = 0$ are visited from left to right.

All operations at a specific SC-decoding-tree node can be in principle performed in parallel. However, when the SC-decoder hardware implementation is considered, the number of PEs that perform the calculations in (10) is constrained to a specific value P, which can improve the

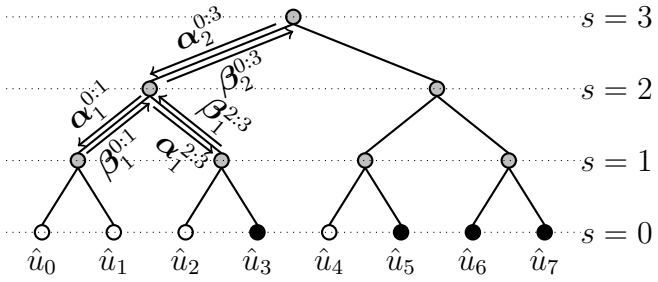


Fig. 2: Binary tree representation of SC decoding for a polar code with $N = 8$ and $R = 1/2$. The white nodes represent frozen bits and the black nodes represent information bits.

trade-off between chip area and latency [27]. As shown in [27], if the channel LLR values are readily available, then the latency of SC decoding is

$$\mathcal{L} = 2N + \frac{N}{P} \log_2 \left(\frac{N}{4P} \right). \tag{13}$$

For different values of P, Fig. 3 shows the resulting LLR values at each time step in a length $N = 8$ polar code. When $P = \frac{N}{2}$, the decoder can perform all the parallelizable operations in one time step, thus the implementation is *fully-parallel* (see Fig. 3a). When $P = 1$, only one operation can be performed at each time step, thus the implementation is *fully-serial* (see Fig. 3c). Any P in the interval $(1, \frac{N}{2})$ results in a *semi-parallel* implementation (see Fig. 3b).

The latency of SC decoding can be represented on a binary tree by assigning *decoding weights* to each edge based on the value of P, as illustrated in Fig. 4. At each edge of the decoding tree that connects a node at level $s + 1$ to a node at level s, the decoding weight is calculated as $\lceil \frac{2^s}{P} \rceil$, where P is assumed to be a positive integer. In Fig. 4a's fully-parallel implementation, all the edges have a decoding weight of 1 since all the parallelizable operations are performed in parallel. However, in a fully-serial implementation of Fig. 4c, the edges at the top of the SC decoding tree consume more time steps, thus their decoding weights are larger. Using the binary tree representation, the latency of SC decoding can be calculated by adding the decoding weights on all the edges. Note that in a fully-parallel implementation, $\mathcal{L} = 2N - 2$, and in a fully-serial implementation, $\mathcal{L} = N \log_2 N$. The latency in a fully-serial implementation is also the decoding complexity.

time	1	2	3	4	5	6	7	8	9	10	11	12	13	14
PE$_1$	α_2^0	α_1^0	α_0^0	α_0^1	α_1^2	α_0^2	α_0^3	α_2^4	α_1^4	α_0^4	α_0^5	α_1^6	α_0^6	α_0^7
PE$_2$	α_2^1	α_1^1			α_1^3			α_2^5	α_1^5			α_1^7		
PE$_3$	α_2^2							α_2^6						
PE$_4$	α_2^3							α_2^7						
output			$\hat{u}_0$	$\hat{u}_1$		$\hat{u}_2$	$\hat{u}_3$			$\hat{u}_4$	$\hat{u}_5$		$\hat{u}_6$	$\hat{u}_7$

(a) Fully-parallel ($P = 4$).

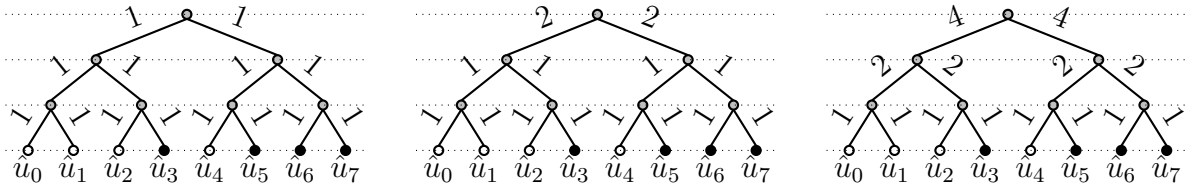
time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
PE$_1$	α_2^0	α_2^2	α_1^0	α_0^0	α_0^1	α_1^2	α_0^2	α_0^3	α_2^4	α_2^6	α_1^4	α_0^4	α_0^5	α_1^6	α_0^6	α_0^7
PE$_2$	α_2^1	α_2^3	α_1^1			α_1^3			α_2^5	α_2^7	α_1^5			α_1^7		
output				$\hat{u}_0$	$\hat{u}_1$		$\hat{u}_2$	$\hat{u}_3$				$\hat{u}_4$	$\hat{u}_5$		$\hat{u}_6$	$\hat{u}_7$

(b) Semi-parallel ($P = 2$).

time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
PE$_1$	α_2^0	α_2^1	α_2^2	α_2^3	α_1^0	α_1^1	α_0^0	α_0^1	α_1^2	α_1^3	α_0^2	α_0^3	α_2^4	α_2^5	α_2^6	α_2^7	α_1^4	α_1^5	α_0^4	α_0^5	α_1^6	α_1^7	α_0^6	α_0^7
output							$\hat{u}_0$	$\hat{u}_1$			$\hat{u}_2$	$\hat{u}_3$							$\hat{u}_4$	$\hat{u}_5$			$\hat{u}_6$	$\hat{u}_7$

(c) Fully-serial ($P = 1$).

Fig. 3: SC decoding schedule for a polar code with $N = 8$.



(a) Fully-parallel ($P = 4$). (b) Semi-parallel ($P = 2$). (c) Fully-serial ($P = 1$).

Fig. 4: Decoding weights on a SC decoding tree for a polar code with $N = 8$ and $R = 1/2$.

C. Simplified Successive-Cancellation Decoding

The SSC decoding algorithm [19] identifies two types of nodes in the SC decoding tree. The bits within each node can be decoded efficiently in one shot without traversing its descendent nodes. These two types of nodes are:

- *Rate-0 node*: A node at level s of the SC decoding tree all of whose leaf nodes at level 0 are frozen bits. For a Rate-0 node at level s, bit estimates can be directly calculated at the

level where the node is located as

$$\beta_s^i = 0. \tag{14}$$

- *Rate-1 node*: A node at level s of the SC decoding tree whose leaf nodes at level 0 are all information bits. For a Rate-1 node at level s, the bit estimations can be directly calculated at the level where the node is located as

$$\beta_s^i = \begin{cases} 0 & \text{if } \alpha_s^i > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{15}$$

This paper considers a non-systematic polar code, whose information bits appear at level 0. A non-systematic polar code requires hard decisions to calculate the information bits at level 0 from the estimated bits at an intermediate level where a Rate-0 or a Rate-1 node is located. Calculating the bit estimates in (11) and calculating the information bits at level 0 of the decoding tree from the estimated bits at an intermediate level where a Rate-0 or a Rate-1 node is located are bit-wise operations that are conducted in the same time step in which the LLR values are calculated [20], [21]. This is due to the fact that the time it takes to perform bit-wise calculations is negligible with respect to performing LLR calculations. In fact, these bit-wise calculations can be implemented efficiently using shift-registers [30], [31]. Moreover, if a systematic polar code [32] (whose information bits appear at level n) is considered, there is no need to calculate the bit values at the leaf nodes because the information is present in the root node of the decoding tree. In fact, SSC decoding can decode Rate-0 and Rate-1 nodes in a single time step. In a binary tree representation of SC decoding, this corresponds to pruning all the nodes that are the descendants of a Rate-0 node or a Rate-1 node, as illustrated in Fig. 5.

For practical code lengths, SSC decoding has a significantly lower latency than SC decoding [19]. This is due to the fact that the number of edges in the SSC decoding tree is significantly smaller than the number of edges in the SC decoding tree. Further, the latency of SSC decoding can be calculated by adding all the decoding weights in its (pruned) binary tree representation (as done in the case of SC decoding).

III. Latency of SSC Decoding with Limited Parallelism

Theorem 1 (Latency of SSC Decoder with Limited Parallelism)**:** Let W be a given BMS channel with symmetric capacity $I(W)$. Fix p_e and design a sequence of polar codes $\mathcal{C}_{\text{polar}}(p_e, W, N)$ of increasing block lengths with rates approaching $I(W)$, as per Definition 1. Then, for any

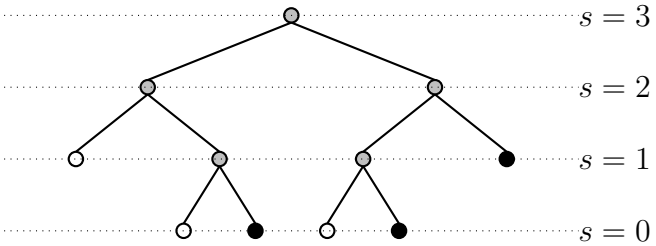


Fig. 5: Binary tree representation of SSC decoding for a polar code with $N = 8$ and $R = 1/2$. The white nodes represent Rate-0 nodes, the black nodes represent Rate-1 nodes, and the gray nodes are neither Rate-0 nodes nor Rate-1 nodes.

$\epsilon > 0$, there exists $\bar{N}(\epsilon)$ such that, for any $N \geq \bar{N}(\epsilon)$, the latency of the SSC decoder with P processing elements is upper bounded by

$$c\, N^{1-1/\mu} + (2+\epsilon)\frac{N}{P}\log_2\log_2\frac{N}{P}, \tag{16}$$

where $c > 0$ is an absolute constant (independent of N, P, p_e, ϵ and W).

Some remarks are in order. First, note that, in a fully-serial implementation with $P = 1$, the upper bound (16) reduces to

$$(2 + o(1))N \log_2\log_2 N. \tag{17}$$

Furthermore, if $P = N^{1/\mu}$, then (16) is

$$\tilde{O}(N^{1-1/\mu}), \tag{18}$$

where the $\tilde{O}$ notation hides (log-)logarithmic factors. Recall that the latency of a fully-parallel implementation of the SSC decoder is $O(N^{1-1/\mu})$, see Theorem 1 of [26]. Thus, another immediate consequence of Theorem 1 is that $P \sim N^{1/\mu}$ suffices to get roughly the same latency as $P = \frac{N}{2}$, and this is the smallest such P.

The key idea of the proof is to look at various levels of the decoding tree and approximate the number of nodes whose corresponding bit-channels are already polarized beyond a certain threshold. Such nodes will be pruned, thus reducing the total weight of the tree. A similar approach appears in [26], which however considers only the fully-parallel setting where $P = \frac{N}{2}$. Here, in order to be able to handle values of P much smaller than $\frac{N}{2}$, we need to develop a different pruning strategy. The idea is to divide the decoding tree into two parts. The first part

contains nodes/edges at distance at most $\lceil \log_2 \frac{N}{P} \rceil$ from the root node, and we consider pruning at depths roughly $c_1 \log\log \frac{N}{P}$ and $c_2 \log\log \frac{N}{P}$, where c_1 is close to 2 and c_2 is a sufficiently large constant (independent of N and P). The second part of the decoding tree contains the rest of the nodes/edges, and we consider pruning at depths which are logarithmic in $\frac{N}{P}$ (as opposed to the doubly-logarithmic scaling of the depths for the first part of the tree).

In order to provide a rigorous bound on the performance of the aforementioned pruning strategy, we need a refined estimate on the number of un-polarized channels, which is contained in the intermediate lemma below. This result is a two-sided version of the bound on Z_n, as defined in (7), leading to Theorem 3 in [7]. Its proof appears in the Appendix.

Lemma 1 (Refined bound on number of un-polarized channels)**:** Let W be a BMS channel and let $Z_n = Z(W_n)$ be the random process that tracks the Bhattacharyya parameter of W_n. Let μ be an upper bound on the scaling exponent according to Definition 2. Fix $\gamma \in \left(\frac{1}{1+\mu}, 1\right)$. Then, for $n \geq 1$,

$$\mathbb{P}\left(Z_n \in \left[2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}, 1 - 2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}\right]\right) \leq c\, 2^{-n(1-\gamma)/\mu}, \tag{19}$$

where c is a numerical constant that does not depend on n, W, or γ, and $h_2^{(-1)}$ is the inverse of the binary entropy function $h_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$ for $x \in [0, 1/2]$.

We will also use the following intermediate result, which is stated as Lemma 2 in [26].

Lemma 2 (Sufficient condition for Rate-0 and Rate-1 nodes)**:** Let W be a BMS channel, $p_e \in (0,1)$, $N = 2^n$, and $M = 2^m$ with $m < n$. Consider the polar code $\mathcal{C}_{\text{polar}}(p_e/M, W, N/M)$ constructed according to Definition 1. Then, there exists an integer n_0, which depends on p_e, such that for all $n \geq n_0$, the following holds:

1) If $Z(W) \leq 1/N^3$, then the polar code $\mathcal{C}_{\text{polar}}(p_e/M, W, N/M)$ has rate 1.
2) If $Z(W) \geq 1 - 1/N^3$, then the polar code $\mathcal{C}_{\text{polar}}(p_e/M, W, N/M)$ has rate 0.

At this point, the proof of Theorem 1 is presented.

Proof of Theorem 1. The decoding tree is divided into two segments. The first part is called $\mathcal{F}_1$ and it consists of all nodes/edges at distance at most $\lceil \log_2 \frac{N}{P} \rceil$ from the root node. The second part is called $\mathcal{F}_2$ and it consists of the rest, which are all the nodes/edges in the bottom $\lfloor \log_2 P \rfloor$ levels. To analyze the latency, three cases are considered: (**Case A**) $N^{0.01} \leq P \leq N^{0.99}$

(moderate values of P), (**Case B**) $N^{0.99} \leq P$ (large values of P), and (**Case C**) $P \leq N^{0.01}$ (small values of P).

Case A: $N^{0.01} \leq P \leq N^{0.99}$. Let us first look at $\mathcal{F}_1$, and consider pruning at depths k_1 and $k_1 + k_2$, with

$$\begin{aligned} k_1 &= \left\lceil c_1 \log_2 \log_2 \frac{N}{P} \right\rceil, \\ k_2 &= \left\lceil c_2 \log_2 \log_2 \frac{N}{P} \right\rceil, \end{aligned} \tag{20}$$

where c_1 and c_2 are constants to be determined later. Further assume that

$$c_1 \gamma_1 h_2^{(-1)} \left(\frac{\gamma_1(\mu+1)-1}{\gamma_1 \mu} \right) > 1, \tag{21}$$

$$c_2 \gamma_2 h_2^{(-1)} \left(\frac{\gamma_2(\mu+1)-1}{\gamma_2 \mu} \right) > 1, \tag{22}$$

where the constants γ_1 and γ_2 will be also determined later. If (21) and (22) are true, then, as $P \leq N^{0.99}$, for sufficiently large values of N,

$$\begin{aligned} 2^{-2^{k_1 \gamma_1 h_2^{(-1)}\left(\frac{\gamma_1(\mu+1)-1}{\gamma_1 \mu}\right)}} &\leq \frac{1}{N^3}, \\ 2^{-2^{k_2 \gamma_2 h_2^{(-1)}\left(\frac{\gamma_2(\mu+1)-1}{\gamma_2 \mu}\right)}} &\leq \frac{1}{N^3}. \end{aligned} \tag{23}$$

Also,

$$\lim_{\gamma_1 \to 1} \gamma_1 h_2^{(-1)} \left(\frac{\gamma_1(\mu+1)-1}{\gamma_1 \mu} \right) = \frac{1}{2}. \tag{24}$$

We choose $c_1 = 2 + \epsilon$ for a positive ϵ. In view of (24), there exists $\delta > 0$ such that (21) is satisfied by taking $\gamma_1 = 1 - \delta$. Furthermore, we pick $\gamma_2 = 0.9$ and $c_2 = 100$. Selecting $\mu \geq 2$ ensures that (22) holds.

Now, the latency associated to $\mathcal{F}_1$ can be computed. To do so, $\mathcal{F}_1$ is partitioned into three parts: *(i)* nodes that appear above depth k_1, *(ii)* what remains between depth k_1 and the next k_2 levels after pruning the tree at depth k_1, and *(iii)* what remains of $\mathcal{F}_1$ after pruning at depth $k_1 + k_2$.

For part *(i)*, the total decoding weight sums up to

$$\sum_{i=1}^{k_1} 2^i \left\lceil \frac{N}{2^i P} \right\rceil \leq 2^{k_1+1} + k_1 \frac{N}{P}. \tag{25}$$

At depth k_1, there are a total of 2^{k_1} nodes prior to the pruning. By using Lemma 1 and the first inequality in (23), there are at most

$$a_1 \triangleq c2^{k_1(1-\frac{1-\gamma_1}{\mu})} \leq c2^{k_1} \tag{26}$$

nodes whose Bhattacharyya parameter is in the interval $[1/N^3, 1-1/N^3]$. Thus, by applying Lemma 2 with $M = 2^{k_1}$ and desired error probability set to $p_e\frac{2^{k_1}}{N}$, all but those a_1 nodes can be pruned. Hence, part *(ii)* of $\mathcal{F}_1$ consists of at most a_1 sub-trees with depth k_2. Consequently, the total decoding weight for part *(ii)* can be upper bounded by

$$a_1 \sum_{i=1}^{k_2} 2^i \left\lceil \frac{N}{2^{i+k_1}P} \right\rceil \leq a_1\, 2^{k_2+1} + a_1\, k_2 \frac{N}{P2^{k_1}}. \tag{27}$$

At depth k_2, each of the sub-trees has a total of 2^{k_2} nodes before pruning. By using Lemma 1 and the second inequality in (23), at most $c2^{k_2(1-\frac{1-\gamma_2}{\mu})}$ of these nodes have Bhattacharyya parameter in the interval $[1/N^3, 1-1/N^3]$. Let v denote one of these at most $c2^{k_2(1-\frac{1-\gamma_2}{\mu})}$ nodes, and consider the subtree rooted at v. If we descend k_1 levels in this subtree, there are a total of 2^{k_1} nodes in it prior to pruning. However, by Lemma 1 and the first inequality in (23), at most $c^{k_1(1-\frac{1-\gamma_1}{\mu})}$ of these 2^{k_1} nodes have Bhattacharyya parameter in the interval $[1/N^3, 1-1/N^3]$. Thus, by applying Lemma 2 with $M = 2^{k_1+k_2}$ and error probability set to $p_e\frac{2^{k_1+k_2}}{N}$, the number of remaining nodes after pruning at depth k_1+k_2 can be upper bounded by

$$a_2 \triangleq c^2 2^{k_1(1-\frac{1-\gamma_1}{\mu})} 2^{k_2(1-\frac{1-\gamma_2}{\mu})}. \tag{28}$$

Consequently, the total decoding weight for part *(iii)* can be upper bounded by

$$\begin{aligned} a_2 \sum_{i=1}^{\lceil \log_2 \frac{N}{P} \rceil - k_1 - k_2} 2^i \left\lceil \frac{N}{2^{i+k_1+k_2}P} \right\rceil &\leq a_2\, 2^{\lceil \log_2 \frac{N}{P} \rceil - k_1 - k_2 + 1} \\ &\quad + a_2 \left(\left\lceil \log_2 \left(\frac{N}{P} \right) \right\rceil - k_1 - k_2 \right) \frac{N}{P2^{k_1+k_2}}. \end{aligned} \tag{29}$$

As a result, the latency associated to $\mathcal{F}_1$ is upper bounded by the sum of the terms in (25), (27), and (29). By using the definitions of k_1 and k_2 in (20) and of a_1 and a_2 in (26) and (28),

after some algebraic manipulations,

$$\begin{aligned}
&2^{k_1+1} \leq 4\left(\log_2 \frac{N}{P}\right)^{c_1},\\
&a_1\, 2^{k_2+1} \leq c2^{k_1}2^{k_2+1} \leq 8c\left(\log_2 \frac{N}{P}\right)^{c_1+c_2},\\
&a_1\, k_2 \frac{N}{P2^{k_1}} = \frac{c\frac{N}{P}k_2}{2^{k_1\frac{1-\gamma_1}{\mu}}} \leq \frac{c\,\frac{N}{P}\left(c_2\,\log_2\log_2\frac{N}{P}+1\right)}{\left(\log_2\frac{N}{P}\right)^{c_1(1-\gamma_1)/\mu}},\\
&a_2\, 2^{\lceil\log_2\frac{N}{P}\rceil-k_1-k_2+1} = \frac{2c^2\,2^{\lceil\log_2\frac{N}{P}\rceil}}{2^{k_1\frac{1-\gamma_1}{\mu}}2^{k_2\frac{1-\gamma_2}{\mu}}} \leq \frac{4c^2\frac{N}{P}}{\left(\log_2\frac{N}{P}\right)^{\frac{c_1(1-\gamma_1)}{\mu}+\frac{c_2(1-\gamma_2)}{\mu}}},\\
&a_2\left(\left\lceil \log_2\left(\frac{N}{P}\right)\right\rceil - k_1 - k_2\right)\frac{N}{P2^{k_1+k_2}} = \frac{c^2\frac{N}{P}\left(\left\lceil\log_2\left(\frac{N}{P}\right)\right\rceil - k_1 - k_2\right)}{2^{k_1\frac{1-\gamma_1}{\mu}}2^{k_2\frac{1-\gamma_2}{\mu}}}\\
&\qquad\qquad \leq \frac{c^2\frac{N}{P}\log_2\frac{N}{P}}{\left(\log_2\frac{N}{P}\right)^{\frac{c_1(1-\gamma_1)}{\mu}+\frac{c_2(1-\gamma_2)}{\mu}}}.
\end{aligned} \tag{30}$$

Note that $\frac{c_1(1-\gamma_1)}{\mu} > 0$ and $\frac{c_2(1-\gamma_2)}{\mu} > 1$, while $\frac{N}{P} \geq N^{0.01}$. Thus, for large N, all the right hand sides of the expressions in (30) are $o\left(\frac{N}{P}\log_2\log_2\frac{N}{P}\right)$, and the term $\frac{N}{P}k_1$ is the dominant one in the computation of the latency associated to $\mathcal{F}_1$. As a result, for sufficiently large N, this latency is upper bounded by

$$(2+\epsilon)\frac{N}{P}\log_2\log_2\frac{N}{P}, \tag{31}$$

for any $\epsilon > 0$.

Let us now look at $\mathcal{F}_2$, where pruning starts at depth $k_3 = \lceil\log_2\frac{N}{P}\rceil$. By applying Lemma 1 of [26] at depth k_3, for any $\nu > 1$,

$$\mathbb{P}(Z_{k_3} \in [2^{-\nu k_3}, 1-2^{-\nu k_3}]) \leq c2^{-k_3/\mu}, \tag{32}$$

where the constant c depends solely on ν (and not on k_3 or W). Since $P \leq N^{0.99}$, $k_3 \geq 0.01\log_2 N$. Thus, by taking $\nu = 300$ in (32), at level k_3, the number of nodes whose Bhattacharyya parameter is in the interval $[1/N^3, 1-1/N^3]$ is at most

$$a_3 \triangleq c_3\, 2^{k_3(1-\frac{1}{\mu})}, \tag{33}$$

for some constant c_3. Thus, by applying Lemma 2 with $M = 2^{k_3}$ and error probability $\frac{p_e}{2^{n-k_3}}$, the number of remaining nodes after pruning at this level can be upper bounded by a_3. Consequently, $\mathcal{F}_2$ consists of at most a_3 sub-trees of depth $\lfloor\log_2 P\rfloor$. Given that all nodes in $\mathcal{F}_2$ have decoding weights of 1, the pruning strategy of [26] can be applied. Recall that $P \geq N^{0.01}$. Thus, by

following the same strategy as in the proof of Theorem 1 in [26] and by boosting the constants ν by a factor of 100, after pruning, each such sub-tree has a decoding weight of at most

$$c_4\, P^{1-\frac{1}{\mu}}, \tag{34}$$

for some constant c_4. Therefore, the decoding latency over $\mathcal{F}_2$ can be upper bounded by

$$a_3\, c_4\, P^{1-\frac{1}{\mu}} = c_5 N^{1-\frac{1}{\mu}}, \tag{35}$$

for some constant c_5. Combining the upper bounds in (31) and (35) concludes the proof for **Case A**.

Case B: $N^{0.99} \leq P$. There is no need to prune part $\mathcal{F}_1$ of the tree. In fact, without any pruning, its latency is upper bounded by

$$\frac{N}{P}\log_2\frac{N}{P} \leq 0.01\, N^{0.01}\log_2 N. \tag{36}$$

Part $\mathcal{F}_2$ starts at depth $k = \lceil \log_2 \frac{N}{P} \rceil \leq \lceil 0.01 \cdot \log_2 N \rceil$. Recall that the decoding weights over $\mathcal{F}_2$ are all equal to 1. Hence, the latency associated to $\mathcal{F}_2$ can be upper bounded by the decoding latency of the complete tree in a fully-parallel setup. This, in turn, is upper bounded by $cN^{1-\frac{1}{\mu}}$ for some universal constant $c > 0$, see Theorem 1 of [26]. To conclude, note that the right hand side of (36) is smaller than $N^{1-\frac{1}{\mu}}$ for all sufficiently large N. Thus, the result for **Case B** readily follows.

Case C: $P \leq N^{0.01}$. In this case, most of the latency is associated to $\mathcal{F}_1$. Recall that, when deriving the upper bound of the latency associated to $\mathcal{F}_1$ in **Case A**, the fact that $P \leq N^{0.99}$ is used, which is also satisfied in this case. Hence, by following the same argument as in **Case A**, for all sufficiently large N, the latency associated to $\mathcal{F}_1$ is upper bounded by

$$(2+\epsilon)\frac{N}{P}\log_2\log_2\frac{N}{P}, \tag{37}$$

for any $\epsilon > 0$. Let us now look at $\mathcal{F}_2$. The tree is pruned at depth $k = \lceil \log_2 \frac{N}{P} \rceil \geq 0.99 \log_2 N$. Thus, by applying (32) with $\nu = 4$, at depth k, the number of nodes whose Bhattacharyya parameter is in the interval $[1/N^3, 1 - 1/N^3]$ is at most $a \triangleq c(\frac{N}{P})^{1-1/\mu}$, for some constant c. Hence, by applying Lemma 2 with $M = 2^k$, the number of remaining nodes after pruning at this level can be upper bounded by a. Consequently, $\mathcal{F}_2$ consists of at most a many sub-trees of depth $\lfloor \log_2 P \rfloor$. Therefore, the latency associated to $\mathcal{F}_2$ is upper bounded by

$$2aP = o\left(\frac{N}{P}\right), \tag{38}$$

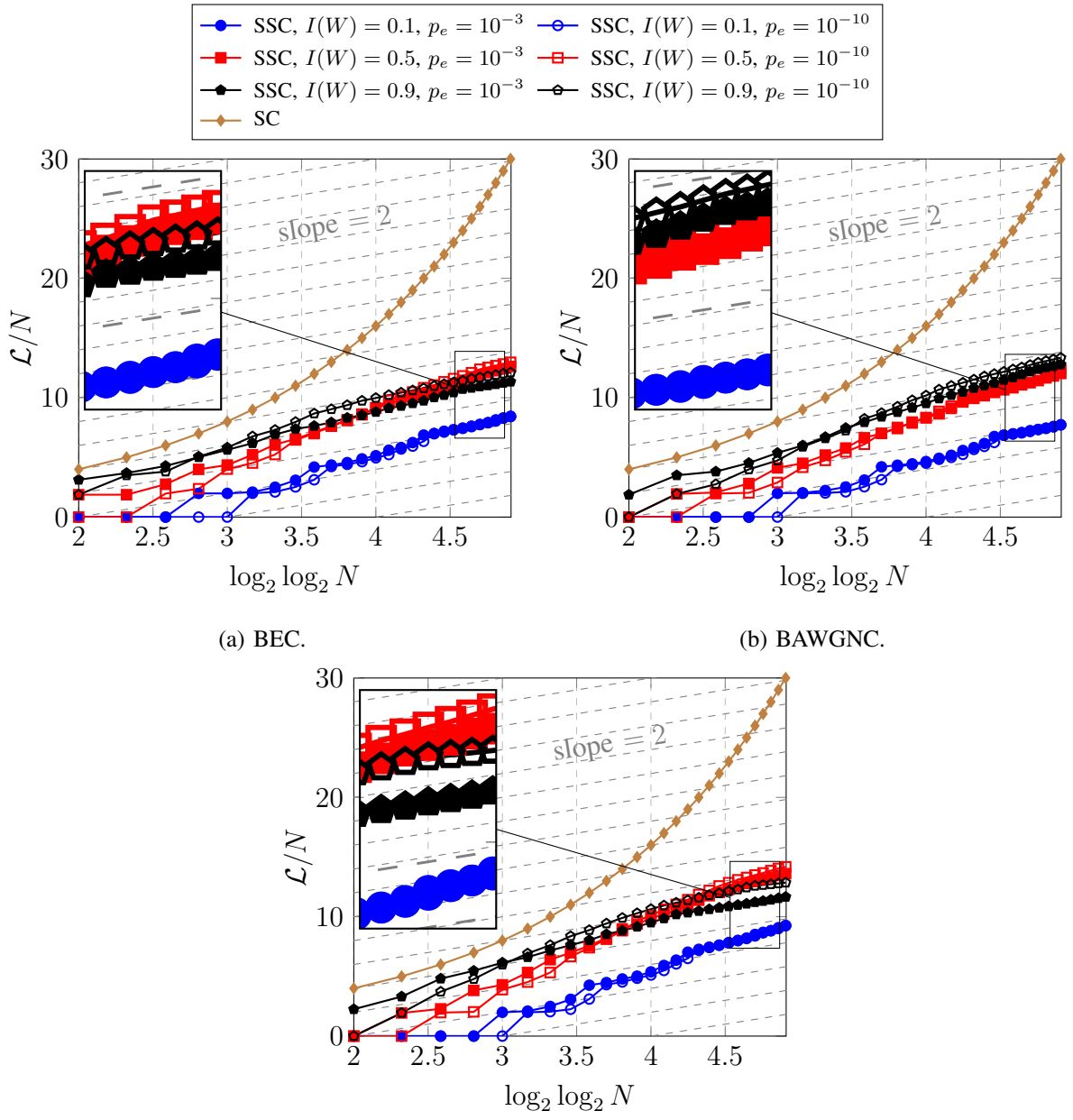
TABLE I: Slopes of the best linear fits for the normalized latency $\mathcal{L}/N$ of the fully-serial implementation of SSC decoding as a function of $\log_2 \log_2 N$ when $26 \leq \log_2 N \leq 30$ for different values of $I(W)$ and p_e. For comparison, note that the same line for SC decoding has a slope of 19.371.

$I(W)$	p_e	slope		
		BEC	BAWGNC	BSC
0.1	10^{-3}	3.322	2.353	3.706
0.1	10^{-10}	3.403	2.594	3.952
0.5	10^{-3}	3.287	3.751	3.195
0.5	10^{-10}	2.835	3.831	3.414
0.9	10^{-3}	1.944	2.792	2.168
0.9	10^{-10}	2.017	2.877	1.277

where in the last step $P \leq N^{0.01}$ and $\mu \in [2, 5]$ are considered. This establishes the fact that (37) is the dominant term in the computation of latency, which in turn completes the proof for **Case C**. □

IV. Numerical Results

This section numerically evaluates SSC-decoding latency for polar codes, constructed based on Definition 1 with $4 \leq \log_2 N \leq 30$, when a limited number of PEs are available. To illustrate SSC-decoding latency in a fully-serial implementation ($P = 1$), Fig. 6 plots the latency normalized with respect to the block length N, namely $\mathcal{L}/N$ (on the y-axis) versus $\log_2 \log_2 N$ (on the x-axis) when $I(W) \in \{0.1, 0.5, 0.9\}$ and $p_e \in \{10^{-3}, 10^{-10}\}$ for BEC (Fig. 6a), BAWGNC (Fig. 6b), and BSC (Fig. 6c). These figures show that SSC decoder's normalized decoding latency grows linearly with $\log_2 \log_2 N$, confirming Theorem 1's upper bound (see (17)). Moreover, the curves' slope approaches 2, as predicted by our theoretical result. The normalized latency of SC decoding grows exponentially in the $\log_2 \log_2 N$ domain because the SC decoder has a latency of $N \log_2 N$ when $P = 1$. Table I shows the slopes of the best linear fits for the last five points in Fig. 6. It can be seen that for all values of $I(W)$ and p_e and for different channels, the slopes of the best linear fits in the finite block length regime when $26 \leq \log_2 N \leq 30$ are quite close to 2. For comparison, let us point out that the same line for SC decoding has a slope of 19.371.



(a) BEC.

(b) BAWGNC.

(c) BSC.

Fig. 6: Normalized latency of SC and SSC decoding of polar codes in a fully-serial implementation ($P = 1$). As the code length N increases, the slope of the curves for SSC decoding tends to 2, confirming that the latency of the simplified decoder scales as $(2 + o(1))N \log_2 \log_2 N$.

Fig. 7 shows the SSC-decoding latency with $P \in \{1, N^{\frac{1}{8}}, N^{\frac{1}{\mu}}, N^{\frac{1}{2}}, \frac{N}{2}\}$. The polar codes are constructed for a BEC with $I(W) = 0.5$ and $p_e = 10^{-3}$. It can be seen that, as N increases, the slope of the curve with $P = N^{\frac{1}{\mu}}$ approaches $1 - \frac{1}{\mu}$, which is 0.72 for the BEC since $\mu \approx 3.63$ in

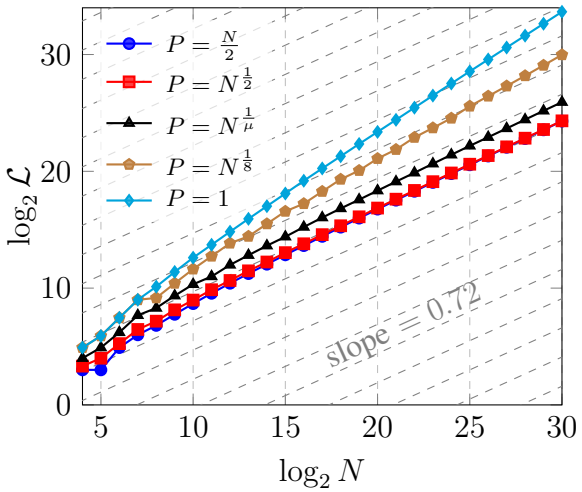


Fig. 7: Latency of SSC decoding of a polar code constructed for a BEC with $I(W) = 0.5$ and $p_e = 10^{-3}$ considering different values of P. The slope of the curve when $P = N^{\frac{1}{\mu}}$ is $1 - \frac{1}{\mu} = 0.72$ and is similar to the case where $P = \frac{N}{2}$.

TABLE II: Slopes of the best linear fits for the logarithm of the latency $\log_2 \mathcal{L}$ of the semi-parallel implementation of SSC decoding as a function of $\log_2 N$ when $26 \leq \log_2 N \leq 30$ for different values of P. Note that W is a BEC, $I(W) = 0.5$, and $p_e = 10^{-3}$.

P	slope
$\frac{N}{2}$	0.748
$N^{\frac{1}{2}}$	0.743
$N^{\frac{1}{\mu}}$	0.751
$N^{\frac{1}{8}}$	0.886
1	1.020

this case. This scaling is the same as the lowest achievable latency when $P = \frac{N}{2}$. Table II shows the slopes of the best linear fits for the last five points in Fig. 7. It can be seen that when $P \geq N^{\frac{1}{\mu}}$, the slopes of the best linear fits in the finite block length regime when $26 \leq \log_2 N \leq 30$ are close to 0.72, as predicted by our theoretical results.

Fig. 8 shows how P scales as N increases when SSC-decoder latency is only 1% higher than fully-parallel SSC decoding (i.e., the latency for $P = \frac{N}{2}$). The polar codes at different block lengths are constructed for a BEC with $I(W) = 0.5$ and $p_e = 10^{-3}$. Theorem 1 predicts that, if P scales as $N^{\frac{1}{\mu}}$, then the latency is close to that of the fully-parallel implementation, which

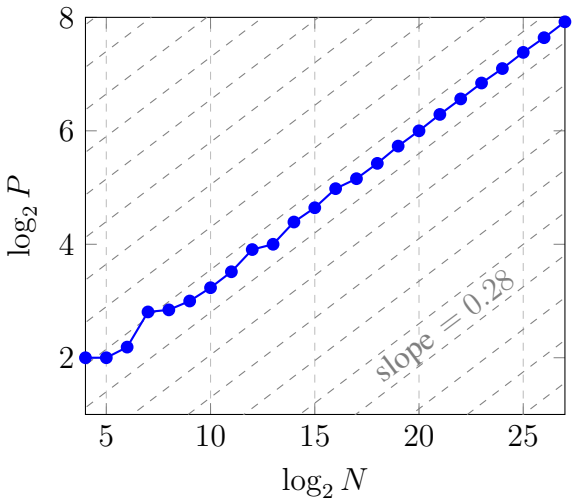


Fig. 8: Required value of P to achieve a latency for SSC decoding that is 1% more than the fully-parallel implementation ($P = \frac{N}{2}$). Polar codes are constructed for a BEC with $I(W) = 0.5$ and $p_e = 10^{-3}$. The slope of the curve is $\frac{1}{\mu} = 0.28$.

Fig. 8 confirms because the curve's slope is $\frac{1}{\mu} = 0.28$.

V. Summary

This paper characterizes the latency of simplified successive-cancellation (SSC) decoding when there is a limited number of processing elements available to implement the decoder. We show that for a polar code of block length N, when the number of processing elements P is limited, the latency of SSC decoding is $O(N^{1-1/\mu} + \frac{N}{P}\log_2\log_2\frac{N}{P})$, where μ is the scaling exponent of the channel. The bound resulted in three important implications. First, a fully-parallel implementation with $P = \frac{N}{2}$ results in a sublinear latency for SSC decoding, which recovers the result in [26]. Second, a fully-serial implementation with $P = 1$ results in a latency for SSC decoding that scales as $(2 + o(1))N\log_2\log_2 N$. Third, it is shown that $P = N^{1/\mu}$ in a semi-parallel implementation is the smallest P that results in the same latency as that of the fully-parallel implementation of SSC decoding.

Acknowledgments

S. A. Hashemi is supported by a Postdoctoral Fellowship from the Natural Sciences and Engineering Research Council of Canada (NSERC) and by Huawei. M. Mondelli is partially

supported by the 2019 Lopez-Loreta Prize. A. Fazeli and A. Vardy were supported in part by the National Science Foundation under Grant CCF-1764104.

REFERENCES

[1] E. Arıkan, "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, July 2009.

[2] "Final report of 3GPP TSG RAN WG1 #87 v1.0.0," Reno, USA, Nov. 2016.

[3] J. W. Won and J. M. Ahn, "3GPP URLLC patent analysis," *ICT Express*, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405959520302046

[4] M. Mondelli, S. H. Hassani, and R. Urbanke, "Construction of polar codes with sublinear complexity," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 2782–2791, May 2019.

[5] E. Arıkan and I. E. Telatar, "On the rate of channel polarization," in *Proc. of the IEEE Int. Symposium on Inf. Theory (ISIT)*, Seoul, South Korea, July 2009, pp. 1493–1495.

[6] S. H. Hassani, K. Alishahi, and R. Urbanke, "Finite-length scaling for polar codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5875–5898, Oct. 2014.

[7] M. Mondelli, S. H. Hassani, and R. Urbanke, "Unified scaling of polar codes: Error exponent, scaling exponent, moderate deviations, and error floors," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6698–6712, Dec. 2016.

[8] V. Guruswami and P. Xia, "Polar codes: Speed of polarization and polynomial gap to capacity," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 3–16, Jan. 2015.

[9] D. Goldin and D. Burshtein, "Improved bounds on the finite length scaling of polar codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6966–6978, Nov. 2014.

[10] M. Mondelli, S. H. Hassani, and R. Urbanke, "Scaling exponent of list decoders with applications to polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4838–4851, Sep. 2015.

[11] A. Fazeli, H. Hassani, M. Mondelli, and A. Vardy, "Binary linear codes with optimal scaling: Polar codes with large kernels," *IEEE Trans. Inf. Theory*, vol. 67, no. 9, pp. 5693–5710, Sep. 2021.

[12] V. Guruswami, A. Riazanov, and M. Ye, "Arıkan meets Shannon: Polar codes with near-optimal convergence to channel capacity," ser. STOC 2020. New York, NY, USA: Association for Computing Machinery, 2020.

[13] S. B. Korada, A. Montanari, E. Telatar, and R. Urbanke, "An empirical scaling law for polar codes," in *Proc. IEEE Int. Symp. on Inf. Theory (ISIT)*, Austin, TX, USA, Jun. 2010, pp. 884–888.

[14] H.-P. Wang and I. M. Duursma, "Polar codes' simplicity, random codes' durability," *IEEE Trans. Inf. Theory*, vol. 67, no. 3, pp. 1478–1508, Mar. 2021.

[15] S. Fong and V. Tan, "Scaling exponent and moderate deviations asymptotics of polar codes for the AWGN channel," *Entropy*, vol. 19, no. 7, p. 364, 2017.

[16] H.-P. Wang and I. Duursma, "Polar code moderate deviation: Recovering the scaling exponent," `arXiv:1806.02405`", June 2018.

[17] J. Błasiok, V. Guruswami, and M. Sudan, "Polar codes with exponentially small error at finite block length," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, no. 34, 2018, pp. 34:1–34:18.

[18] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[19] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Commun. Lett.*, vol. 15, no. 12, pp. 1378–1380, Dec. 2011.

[20] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 946–957, May 2014.

[21] M. Hanif and M. Ardakani, "Fast successive-cancellation decoding of polar codes: Identification and decoding of new nodes," *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2360–2363, Nov. 2017.

[22] C. Condo, V. Bioglio, and I. Land, "Generalized fast decoding of polar codes," in *IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[23] S. A. Hashemi, C. Condo, and W. J. Gross, "A fast polar code list decoder architecture based on sphere decoding," *IEEE Trans. Circuits Syst. I*, vol. 63, no. 12, pp. 2368–2380, Dec. 2016.

[24] S. A. Hashemi, C. Condo, and W. J. Gross, "Fast and flexible successive-cancellation list decoders for polar codes," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5756–5769, Nov. 2017.

[25] M. Hanif, M. H. Ardakani, and M. Ardakani, "Fast list decoding of polar codes: Decoders for additional nodes," in *IEEE Wireless Commun. and Networking Conf. Workshops (WCNCW)*, 2018, pp. 37–42.

[26] M. Mondelli, S. A. Hashemi, J. M. Cioffi, and A. Goldsmith, "Sublinear latency for simplified successive cancellation decoding of polar codes," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 18–27, Jan. 2021.

[27] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for polar codes," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 289–299, Jan. 2013.

[28] H. P. Wang and I. M. Duursma, "Log-logarithmic time pruned polar coding," *IEEE Trans. Inf. Theory*, pp. 1–1, 2020.

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

[30] T. Che and G. Choi, "An efficient partial sums generator for constituent code based successive cancellation decoding of polar codes," *arXiv preprint arXiv:1611.09452*, 2016.

[31] G. Berhault, C. Leroux, C. Jego, and D. Dallet, "Partial sums computation in polar codes decoding," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2015, pp. 826–829.

[32] E. Arıkan, "Systematic polar coding," *IEEE Commun. Lett.*, vol. 15, no. 8, pp. 860–862, 2011.

APPENDIX

Proof of Lemma 1. By applying Lemma 1 in [26], for $n_0 \geq 1$,

$$\mathbb{P}(Z_{n_0} \in [2^{-2n_0}, 1 - 2^{-2n_0}]) \leq c_1\, 2^{-n_0/\mu}, \tag{39}$$

where c_1 is a universal constant which does not depend on n_0, W. Let $\{B_n\}_{n\geq 1}$ be a sequence of i.i.d. random variables with distribution Bernoulli$(1/2)$. Then, by using (7), it is clear that, for $n \geq 1$,

$$Z_{n_0+n} \leq \begin{cases} Z_{n_0+n-1}^2, & \text{if } B_n = 1, \\ 2Z_{n_0+n-1}, & \text{if } B_n = 0. \end{cases}$$

Therefore, by applying Lemma 22 of [6], we obtain that, for $n_1 \geq 1$,

$$\mathbb{P}\left(Z_{n_0+n_1} \leq 2^{-2^{\sum_{i=1}^{n_1} B_i}} \mid Z_{n_0} = x\right) \geq 1 - c_2\, x(1 - \log_2 x), \tag{40}$$

with $c_2 = 2/(\sqrt{2}-1)^2$. Thus,

$$\begin{aligned}\mathbb{P}\left(Z_{n_0+n_1} \leq 2^{-2^{\sum_{i=1}^{n_1} B_i}} \mid Z_{n_0} \leq 2^{-n_0}\right) &\geq 1 - c_2 2^{-n_0}(1+n_0) \\ &\geq 1 - c_2 \frac{\sqrt{2}}{\ln 2} 2^{-n_0/\mu},\end{aligned} \tag{41}$$

where the first inequality uses the fact that $1 - c_2 x(1-\log_2 x)$ is decreasing in x for any $x \leq 2^{-n_0} \leq 1/2$, and the second inequality uses that $1 - c_2 2^{-n_0}(1+n_0) \geq 1 - c_2\sqrt{2}\cdot 2^{-n_0/2}/\ln 2$ for any $n_0 \in \mathbb{N}$ and that $\mu > 2$. Furthermore, by using the same passages of (54) in [7], we obtain that, for any $\epsilon \in (0, 1/2)$,

$$\mathbb{P}\left(2^{-2^{\sum_{i=1}^{n_1} B_i}} > 2^{-2^{n_1\epsilon}}\right) \leq 2^{-n_1(1-h_2(\epsilon))}, \tag{42}$$

where $h_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$ denotes the binary entropy function. By combining (41) and (42),

$$\mathbb{P}\left(Z_{n_0+n_1} \leq 2^{-2^{n_1\epsilon}} \mid Z_{n_0} \leq 2^{-n_0}\right) \geq 1 - c_2 \frac{\sqrt{2}}{\ln 2} 2^{-n_0/\mu} - 2^{-n_1(1-h_2(\epsilon))}. \tag{43}$$

Define $Y_n = 1 - Z_n$. Note that, if $Z_{n+1} = Z_n^2$, then

$$Y_{n+1} = 1 - (1-Y_n)^2 = 2Y_n - Y_n^2 \leq 2Y_n. \tag{44}$$

Furthermore, if $Z_{n+1} \geq Z_n\sqrt{2 - Z_n^2}$, then

$$Y_{n+1} \leq 1 - (1-Y_n)\sqrt{2-(1-Y_n)^2} \leq 2Y_n^2, \tag{45}$$

where in the last inequality the fact that $1 - t\sqrt{2-t^2} \leq 2(1-t)^2$ for any $t \in [0,1]$ is used. Thus, by using (7), for $n \geq 1$,

$$Y_{n_0+n} \leq \begin{cases} 2Y_{n_0+n-1}^2, & \text{if } B_n = 1, \\ 2Y_{n_0+n-1}, & \text{if } B_n = 0. \end{cases}$$

Define $\tilde{Y}_{n_0} = 2Y_{n_0}$ and

$$\tilde{Y}_{n_0+n} = \begin{cases} \tilde{Y}_{n_0+n-1}^2, & \text{if } B_n = 1, \\ 2\tilde{Y}_{n_0+n-1}, & \text{if } B_n = 0. \end{cases}$$

Then for any $n \geq 0$,

$$Y_{n_0+n} \leq \frac{1}{2}\tilde{Y}_{n_0+n} \leq \tilde{Y}_{n_0+n}. \tag{46}$$

By applying again Lemma 22 of [6] to the process $\tilde{Y}_n$, for $n_1 \geq 1$,

$$\mathbb{P}\left(\tilde{Y}_{n_0+n_1} \leq 2^{-2^{\sum_{i=1}^{n_1} B_i}} \mid \tilde{Y}_{n_0} \leq 2^{-n_0}\right) \geq 1 - c_2 \frac{\sqrt{2}}{\ln 2} 2^{-n_0/\mu}, \tag{47}$$

which, combined with (42), gives that, for any $\epsilon \in (0, 1/2)$,

$$\mathbb{P}\left(\tilde{Y}_{n_0+n_1} \leq 2^{-2^{n_1\epsilon}} \mid \tilde{Y}_{n_0} \leq 2^{-n_0}\right) \geq 1 - c_2 \frac{\sqrt{2}}{\ln 2} 2^{-n_0/\mu} - 2^{-n_1(1-h_2(\epsilon))}. \tag{48}$$

By using (46) and the fact that $\tilde{Y}_{n_0} = 2Y_{n_0}$, (48) implies that

$$\mathbb{P}\left(Y_{n_0+n_1} \leq 2^{-2^{n_1\epsilon}} \mid Y_{n_0} \leq 2^{-n_0-1}\right) \geq 1 - c_2 \frac{\sqrt{2}}{\ln 2} 2^{-n_0/\mu} - 2^{-n_1(1-h_2(\epsilon))}. \tag{49}$$

Let $n \geq 1$. Set $n_1 = \lceil \gamma n \rceil$, $n_0 = n - \lceil \gamma n \rceil$, and $\epsilon = h_2^{(-1)}\left((\gamma(\mu+1)-1)/(\gamma\mu)\right)$, where $h_2^{(-1)}(\cdot)$ is the inverse of $h_2(x)$ for any $x \in [0, 1/2]$. Note that if $\gamma \in (1/(1+\mu), 1)$, then $\epsilon \in (0, 1/2)$. Consequently, (43) implies that

$$\mathbb{P}\left(Z_n \leq 2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}} \mid Z_{n_0} \leq 2^{-n_0}\right) \geq 1 - c_3 2^{-n\frac{1-\gamma}{\mu}}, \tag{50}$$

where c_3 is a numerical constant. Similarly, by using that $Z_n = 1 - Y_n$, from (49),

$$\mathbb{P}\left(Z_n \geq 1 - 2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}} \mid Z_{n_0} \geq 1 - 2^{-n_0-1}\right) \geq 1 - c_3 2^{-n\frac{1-\gamma}{\mu}}. \tag{51}$$

The proof is concluded by the following chain of inequalities:

$$\begin{aligned}
&\mathbb{P}\left(Z_n \in \left[2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}, 1 - 2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}\right]\right) \\
&= 1 - \mathbb{P}\left(Z_n \leq 2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}\right) - \mathbb{P}\left(Z_n \geq 1 - 2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}\right) \\
&\leq 1 - \mathbb{P}\left(Z_n \leq 2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}, Z_{n_0} \leq 2^{-n_0}\right) \\
&\qquad - \mathbb{P}\left(Z_n \geq 1 - 2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}, Z_{n_0} \geq 1 - 2^{-n_0-1}\right) \\
&= 1 - \mathbb{P}\left(Z_n \leq 2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}} \mid Z_{n_0} \leq 2^{-n_0}\right) \mathbb{P}\left(Z_{n_0} \leq 2^{-n_0}\right) \\
&\qquad - \mathbb{P}\left(Z_n \geq 1 - 2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}} \mid Z_{n_0} \geq 1 - 2^{-n_0-1}\right) \mathbb{P}\left(Z_{n_0} \geq 1 - 2^{-n_0-1}\right) \\
&\overset{\text{(a)}}{\leq} 1 - \left(1 - c_3 2^{-n\frac{1-\gamma}{\mu}}\right)\left(1 - \mathbb{P}(Z_{n_0} \in [2^{-n_0}, 1 - 2^{-n_0-1}])\right) \\
&\overset{\text{(b)}}{\leq} 1 - \left(1 - c_3 2^{-n\frac{1-\gamma}{\mu}}\right)\left(1 - c_1 2^{-n\frac{1-\gamma}{\mu}}\right) \\
&\leq (c_3 + c_1) 2^{-n\frac{1-\gamma}{\mu}},
\end{aligned}$$

where (50) and (51) are used in (a), and (39) is used in (b). □