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

Seyyed Ali Hashemi*, Marco Mondelli†, Arman Fazeli‡, Alexander Vardy‡, John Cioffi*, Andrea Goldsmith§
*Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA, {ahashemi,cioffi}@stanford.edu
†Institute of Science and Technology (IST) Austria, Klosterneuburg, Austria, marco.mondelli@ist.ac.at
‡Department of Electrical and Computer Engineering, UC San Diego, La Jolla, CA 92093, USA, {afazelic,avardy}@ucsd.edu
§Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA, goldsmith@princeton.edu

***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 polar codes for 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 an 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 $\left(2+o(1)\right)N\log_2\log_2 N$. 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.**

A full version of this paper is accessible at: https://arxiv.org/pdf/2012.13378.pdf

I. Introduction

Polar codes achieve capacity for any binary memoryless symmetric (BMS) channel [1], and they 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 [2], [3]. For a polar code of block length N, the encoding and successive-cancellation (SC) decoding complexity for any 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 $I(W)-R \sim N^{-1/\mu}$, 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]–[12]). In general, the scaling exponent μ depends on the transmission channel W. It is known that $3.579 \leq \mu \leq 4.714$ for any BMS channel W [6], [7]. It is possible to approach the optimal scaling exponent $\mu=2$ by using large polarization kernels [11]–[13].

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 [14], where L is the size of the list. SC-based decoding algorithms suffer from high latency. This is due to the fact that SC decoding proceeds sequentially bit by bit. In order to mitigate this issue, a *simplified SC* (SSC) decoder was proposed in [15]. The SSC decoder identifies two specific constituent codes in the polar code whose bits can be decoded in parallel; thus, these constituent codes are decoded in one shot. Consequently, the latency is reduced without increasing the error probability. These results were extended to SCL decoders in [16], [17]. It was shown in [18] that the latency of the SSC decoder is $O\left(N^{1-1/\mu}\right)$. Thus the latency of SSC decoding is *sublinear in* N, in contrast to the $O\left(N\right)$ 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 is allocated to perform the operations in SC-based decoding algorithms [19]. In the extreme case where $P=1$ (a fully-serial architecture), the latency of SC decoding grows from $O\left(N\right)$ to $O\left(N\log_2 N\right)$.

This paper quantifies the latency of the SSC decoder proposed in [15] as a function of hardware resource constraints. Our main result is that the latency of SSC decoding scales as $O\left(N^{1-1/\mu}+\frac{N}{P}\log_2\log_2\frac{N}{P}\right)$ with the block length N. Several consequences of the bound 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 [18]. In a fully-serial implementation, where $P=1$, this bound reduces to $\left(2+o(1)\right)N\log_2\log_2 N$. 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.

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

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

where W^0 is a *worse* channel and W^1 is a *better* channel than W because [1], [20]

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

$$Z(W^1) = Z(W)^2. \tag{3}$$

By recursively performing the operation in (1) 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 $W_n = \begin{cases} W_{n-1}^0, & \text{w.p. } 1/2, \\ W_{n-1}^1, & \text{w.p. } 1/2, \end{cases}$ where $W_0 = W$. Using (2) and (3), the random process that tracks the Bhattacharyya parameter of W_n can be represented as

$$Z_n \begin{cases} \in \left[Z_{n-1}\sqrt{2 - Z_{n-1}^2},\, 2Z_{n-1} - Z_{n-1}^2\right], & \text{w.p. } 1/2, \\ = Z_{n-1}^2, & \text{w.p. } 1/2, \end{cases} \tag{4}$$

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.

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

$$\sup_{\substack{x\in(0,1) \\ y\in[x\sqrt{2-x^2},\,2x-x^2]}} \frac{h(x^2) + h(y)}{2h(x)} < 2^{-1/\mu}. \tag{5}$$

By defining the scaling exponent as in Definition 2, the gap to capacity $I(W) - R$ scales as $O(N^{-1/\mu})$ as N grows [7]. Note that $\mu \approx 3.63$ for BEC, $\mu \approx 4$ for BAWGNC [21], and it is conjectured that $\mu \approx 4.2$ for BSC.

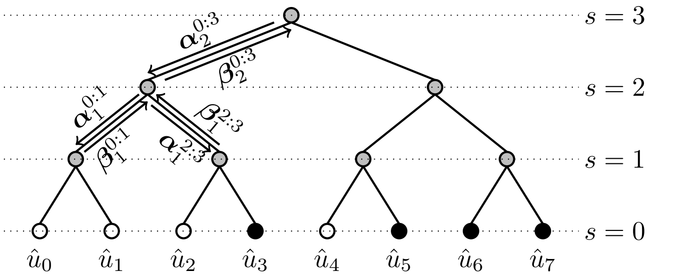

$s = 3$
$s = 2$
$s = 1$
$s = 0$
$\boldsymbol{\alpha}_2^{0:3}$
$\boldsymbol{\beta}_2^{0:3}$
$\boldsymbol{\alpha}_1^{0:1}$
$\boldsymbol{\beta}_1^{0:1}$
$\boldsymbol{\alpha}_1^{2:3}$
$\boldsymbol{\beta}_1^{2:3}$
$\hat{u}_0$
$\hat{u}_1$
$\hat{u}_2$
$\hat{u}_3$
$\hat{u}_4$
$\hat{u}_5$
$\hat{u}_6$
$\hat{u}_7$

Fig. 1: 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.

B. Successive-Cancellation Decoding

SC decoding is a message passing algorithm on the binary-tree representation of polar codes, as shown in Fig. 1 for a polar code of length $N = 8$. At stage n of the decoding tree, 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. 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 at each level s as

$$\alpha_s^i = \begin{cases} f(\alpha_{s+1}^i, \alpha_{s+1}^{i+2^s}) & \text{if } \lfloor \frac{i}{2^s} \rfloor \bmod 2 = 0, \\ g(\alpha_{s+1}^i, \alpha_{s+1}^{i-2^s}, \beta_s^{i-2^s}) & \text{if } \lfloor \frac{i}{2^s} \rfloor \bmod 2 = 1, \end{cases} \tag{6}$$

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 decoding tree. The bit estimates $\boldsymbol{\beta}_s = \{\beta_s^0, \beta_s^1, \ldots, \beta_s^{N-1}\}$ are calculated as

$$\beta_s^i = \begin{cases} \beta_{s-1}^i \oplus \beta_{s-1}^{i+2^s} & \text{if } \lfloor \frac{i}{2^s} \rfloor \bmod 2 = 0, \\ \beta_{s-1}^i & \text{if } \lfloor \frac{i}{2^s} \rfloor \bmod 2 = 1, \end{cases} \tag{7}$$

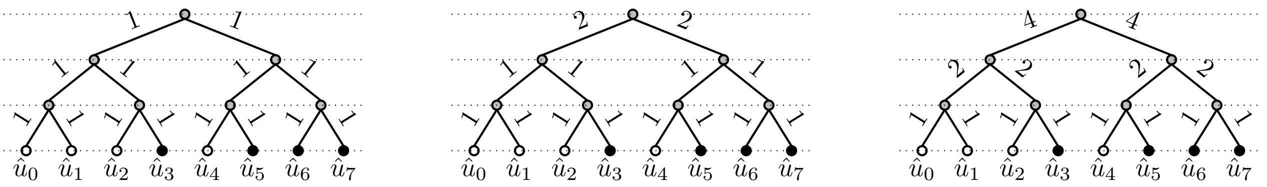
where $\oplus$ is the bit-wise XOR operation. All frozen bits are assumed to be zero. Hence at level $s = 0$, the i-th bit $\hat{u}_i$ is estimated as

$$\hat{u}_i = \beta_0^i = \begin{cases} 0 & \text{if } u_i \text{ is a frozen bit or } \alpha_0^i > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

SC decoding has a sequential structure in the sense that the decoding of each bit depends on the decoding of its previous bits. 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 (6) is constrained to a specific value P, which can improve the trade-off between chip area and latency [19]. As shown in [19], if the channel LLR values are readily available, then the latency of SC decoding is

$$\mathcal{L} = 2N + \frac{N}{P} \log_2\left(\frac{N}{4P}\right). \tag{9}$$


1 1
1 1 1 1
1 1 1 1 1 1 1 1
$\hat{u}_0$ $\hat{u}_1$ $\hat{u}_2$ $\hat{u}_3$ $\hat{u}_4$ $\hat{u}_5$ $\hat{u}_6$ $\hat{u}_7$
2 2
1 1 1 1
1 1 1 1 1 1 1 1
$\hat{u}_0$ $\hat{u}_1$ $\hat{u}_2$ $\hat{u}_3$ $\hat{u}_4$ $\hat{u}_5$ $\hat{u}_6$ $\hat{u}_7$
4 4
2 2 2 2
1 1 1 1 1 1 1 1
$\hat{u}_0$ $\hat{u}_1$ $\hat{u}_2$ $\hat{u}_3$ $\hat{u}_4$ $\hat{u}_5$ $\hat{u}_6$ $\hat{u}_7$

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

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

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

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

When $P = \frac{N}{2}$, the decoder can perform all the parallelizable operations in one time step, thus the implementation is *fully-parallel*. When $P = 1$, only one operation can be performed at each time step, thus the implementation is *fully-serial*. Any P in the interval $(1, \frac{N}{2})$ results in a *semi-parallel* implementation.

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. 2. 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. 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.

C. Simplified Successive-Cancellation Decoding

The SSC decoding algorithm [15] 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
$$\beta_s^i = 0. \tag{10}$$
- *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
$$\beta_s^i = \begin{cases} 0 & \text{if } \alpha_s^i > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{11}$$

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. For practical code lengths, SSC decoding has a significantly lower latency than SC decoding [15]. 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 $\epsilon > 0$, 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

$$c\, N^{1-1/\mu} + (2+\epsilon)\frac{N}{P}\log_2\log_2\frac{N}{P}, \tag{12}$$

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 (12) reduces to

$$(2 + o(1))N \log_2 \log_2 N. \tag{13}$$

Furthermore, if $P = N^{1/\mu}$, then (12) is

$$\tilde{O}(N^{1-1/\mu}), \tag{14}$$

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 [18]. Thus, another immediate consequence of Theorem 1 is that $P \sim N^{1/\mu}$ suffices to get roughly the same latency as $P = N/2$, 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 idea (though with a different pruning strategy) appears in [18]. However, the earlier work in [18] considers only the fully-parallel setting where $P = N/2$.

Before proceeding with the proof, two intermediate lemmas are required. The first one is a two-sided version of the bound on Z_n, as defined in (4), leading to Theorem 3 in [7]. Its proof appears in the extended version of this paper [22].

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 $n \geq 1$,

$$\mathbb{P}\left(Z_n \in \left[2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}, 1-2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}\right]\right) \leq c\, 2^{-n(1-\gamma)/\mu}, \tag{15}$$

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 $x \in [0, 1/2]$.

The second intermediate result is stated as Lemma 2 in [18].

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.

In the rest of this section, we give a sketch of the proof of Theorem 1. We will assume that $N^{0.01} \leq P \leq N^{0.99}$. The cases $N^{0.99} \leq P$ and $P \leq N^{0.01}$ are simpler, and they are handled in the extended version [22], which contains the full proof.

Sketch of the 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 $\lceil \log_2(N/P) \rceil$ 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 $\lfloor \log_2 P \rfloor$ layers.

Let us first look at $\mathcal{F}_1$, and only consider pruning at depths k_1 and $k_1 + k_2$. For $i \in \{1, 2\}$, we set $k_i = \lceil c_i \log_2 \log_2(N/P) \rceil$, where c_1 and c_2 are constants to be determined later. Further assume that, for $i \in \{1, 2\}$,

$$c_i \gamma_i h_2^{(-1)}\left(\frac{\gamma_i(\mu+1)-1}{\gamma_i\mu}\right) > 1, \tag{16}$$

where the constants γ_1 and γ_2 will be also determined later. If (16) holds, then, as $P \leq N^{0.99}$, for sufficiently large values of N,

$$2^{-2^{k_i\gamma_i h_2^{(-1)}\left(\frac{\gamma_i(\mu+1)-1}{\gamma_i\mu}\right)}} \leq \frac{1}{N^3}, \tag{17}$$

for $i \in \{1, 2\}$. We choose $c_1 = 2 + \epsilon$ for a positive ϵ and we note that $\gamma_1 h_2^{(-1)}\left(\frac{\gamma_1(\mu+1)-1}{\gamma_1\mu}\right) \to \frac{1}{2}$ as $\gamma_1 \to 1$. Thus, there exists $\delta > 0$ such that (16) is satisfied for $i = 1$ by taking $\gamma_1 = 1 - \delta$. Furthermore, we pick $\gamma_2 = 0.9$ and $c_2 = 100$. Selecting $\mu \geq 2$ ensures that (16) holds for $i = 2$.

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 layers after pruning the tree at layer 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

$$\sum_{i=1}^{k_1} 2^i \left\lceil \frac{N}{2^i P} \right\rceil \leq 2^{k_1+1} + k_1 \frac{N}{P}. \tag{18}$$

At layer k_1, there are a total of 2^{k_1} nodes prior to the pruning. By using Lemma 1 and (17), there are at most $a_1 \triangleq c2^{k_1(1-\frac{1-\gamma_1}{\mu})}$ 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

$$a_1 \sum_{i=1}^{k_2} 2^i \left\lceil \frac{N}{2^{i+k_1} P} \right\rceil \leq a_1\, 2^{k_2+1} + a_1\, k_2 \frac{N}{P 2^{k_1}}. \tag{19}$$

At layer 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 (17), 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 layers in this subtree, there are a total of 2^{k_1} nodes in it prior to pruning. However, by Lemma 1 and (17), 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 $a_2 \triangleq c^2 2^{k_1(1-\frac{1-\gamma_1}{\mu})} 2^{k_2(1-\frac{1-\gamma_2}{\mu})}$. Consequently, the total decoding weight for part *(iii)* can be upper bounded by

$$a_2 \sum_{i=1}^{\lceil \log_2(N/P) \rceil - k_1 - k_2} 2^i \left\lceil \frac{N}{2^{i+k_1+k_2} P} \right\rceil \leq a_2\, 2^{\lceil \log_2(N/P) \rceil - k_1 - k_2 + 1} + a_2 \left(\left\lceil \log_2\left(\frac{N}{P}\right) \right\rceil - k_1 - k_2 \right) \frac{N}{P 2^{k_1+k_2}}. \tag{20}$$

As a result, the latency associated to $\mathcal{F}_1$ is upper bounded by the sum of the terms in (18), (19), and (20). By using the definitions of k_1, k_2, a_1 and a_2, after some algebraic manipulations (see the extended version [22] for the details), we conclude that, for sufficiently large N, this latency is upper bounded by

$$(2+\epsilon)\frac{N}{P}\log_2\log_2\frac{N}{P}, \tag{21}$$

for any $\epsilon > 0$.

Let us now look at $\mathcal{F}_2$, where pruning starts at layer $k_3 = \lceil \log_2 \frac{N}{P} \rceil$. By applying Lemma 1 of [18] at level k_3, we deduce that for any $\nu > 1$,

$$\mathbb{P}(Z_{k_3} \in [2^{-\nu k_3}, 1 - 2^{-\nu k_3}]) \leq c2^{-k_3/\mu}, \tag{22}$$

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

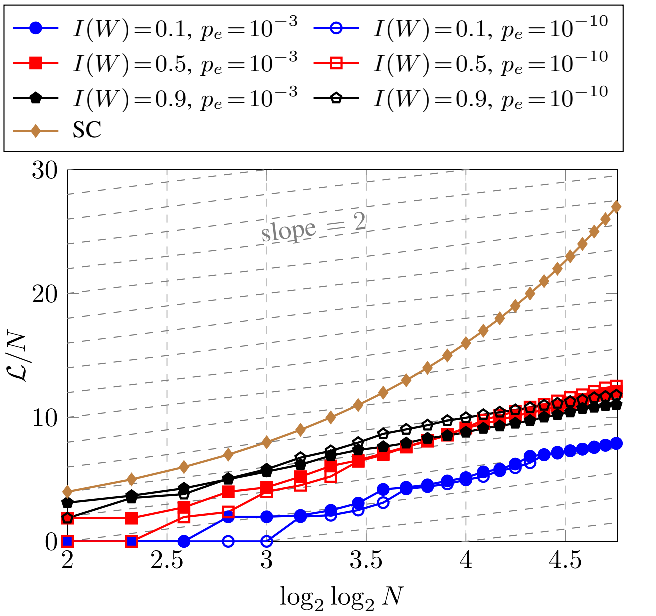


Fig. 3: Normalized latency of SC and SSC decoding of polar codes in a fully-serial implementation ($P = 1$). As the code length N increases, the slope of the curves for SSC decoding tends to 2, confirming that the latency of the simplified decoder scales as $(2 + o(1))N \log_2 \log_2 N$.

$\nu = 300$ in (22), 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 $a_3 \triangleq c_3\, 2^{k_3(1-\frac{1}{\mu})}$, 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 layer 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 [18] can be applied. Recall that $P \geq N^{0.01}$. Thus, by following the same strategy as in the proof of Theorem 1 in [18] and by boosting the constants ν by a factor of 100, after pruning, each such sub-tree has a decoding weight of at most $c_4 P^{1-\frac{1}{\mu}}$, for some constant c_4. Therefore, the decoding latency over $\mathcal{F}_2$ can be upper bounded by $a_3\, c_4\, P^{1-\frac{1}{\mu}} = c_5 N^{1-\frac{1}{\mu}}$, for some constant c_5. Combining this upper bound with the one in (21) concludes the proof. □

IV. Numerical Results

This section numerically evaluates SSC-decoding latency for polar codes, constructed based on Definition 1 with $4 \leq \log_2 N \leq 27$, when a limited number of PEs are available. To illustrate the SSC-decoding latency in a fully-serial implementation ($P = 1$), Fig. 3 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. 3 shows that the SSC decoder's normalized decoding latency grows linearly with $\log_2 \log_2 N$, confirming Theorem 1's upper bound (see (13)). 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 $P = 1$.

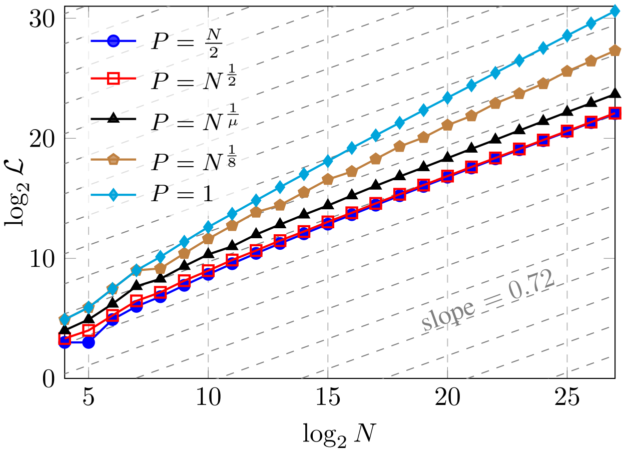


Fig. 4: Latency of SSC decoding of a polar code constructed for a BEC with $I(W) = 0.5$ and $p_e = 10^{-3}$ considering different values of P. The slope of the curve when $P = N^{\frac{1}{\mu}}$ is $1 - \frac{1}{\mu} = 0.72$ and is similar to the case where $P = \frac{N}{2}$.

Fig. 4 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 $p_e = 10^{-3}$. 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 this case. This scaling is the same as the lowest achievable latency when $P = \frac{N}{2}$.

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 [18]. 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. Future work includes the analysis of SSC decoding for large kernels.

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, Jul. 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, Jul. 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*, pp. 1–1, 2020.

[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] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] S. A. Hashemi, M. Mondelli, A. Fazeli, A. Vardy, J. Cioffi, and A. Goldsmith, “Parallelism versus latency in simplified successive-cancellation decoding of polar codes,” Dec. 2020. [Online]. Available: https://arxiv.org/abs/2012.13378