Combining Belief Propagation and Successive Cancellation List Decoding of Polar Codes on a GPU Platform

S. Cammerer, B. Leible, M. Stahl, J. Hoydis, S. ten Brink

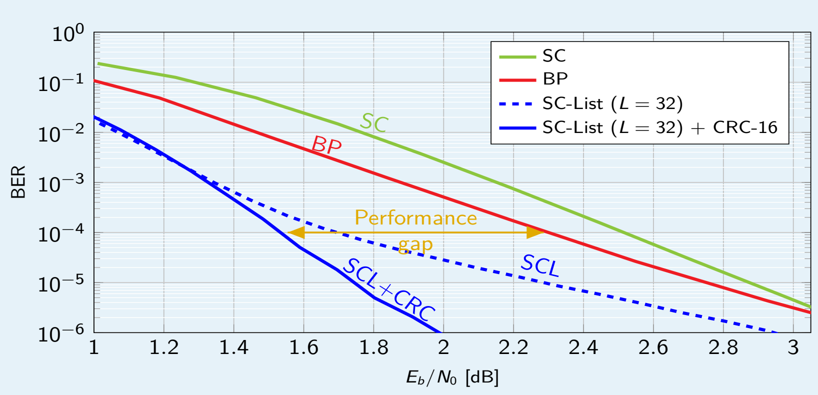
Abstract

The decoding performance of polar codes strongly depends on the decoding algorithm used, while also the decoder throughput and its latency mainly depend on the decoding algorithm. In this work, we implement the powerful successive cancellation list (SCL) decoder on a GPU and identify the bottlenecks of this algorithm with respect to parallel computing and its difficulties. The inherent serial decoding property of the SCL algorithm naturally limits the achievable speed-up gains on GPUs when compared to CPU implementations. In order to increase the decoding throughput, we use a hybrid decoding scheme based on the belief propagation (BP) decoder, which can be intra- and inter-frame parallelized. The proposed scheme combines excellent decoding performance and high throughput within the signal-to-noise ratio (SNR) region of interest.

Motivation

Polar codes are proven to be capacity achieving under successive cancellation (SC) decoding [1] for infinite block lengths. However, for short lengths:

- Error correction capability of polar codes depend on decoding algorithm
- SCL+CRC shows best correction performance
- But: SCL decoding algorithm exhibits low throughput



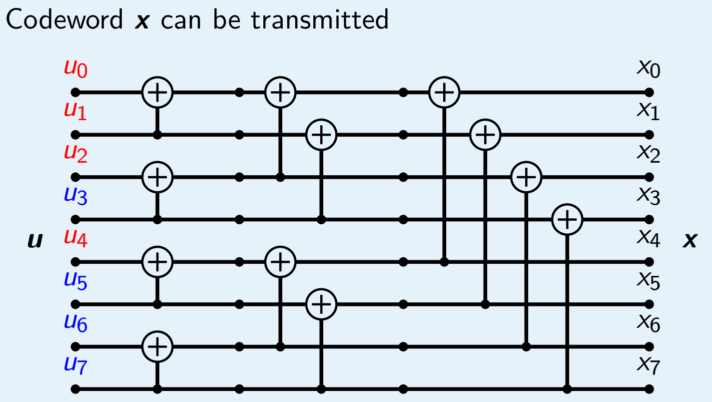
One of the biggest current trends in the telecommunications industry is virtualization with the goal of replacing specialized hardware by software running on commodity servers [2].

- Are polar codes practical for such a scenario?
  - Throughput?
  - Latency?
- Not many GPU implementations of the polar decoder can be found in literature
  - BP decoder can be efficiently implemented [3]
  - Very recently: *fast simplified successive cancellation* GPU implementation [4]
- We focus on GPU implementations using NVIDIA CUDA
- All simulations are performed on an Intel i7-4790K CPU @ 4.00GHz and a NVIDIA GTX 980 Ti.

Polar Codes

A Polar encoder maps the k information bits onto the k most reliable bit positions of the vector $\mathbf{u}$ while the remaining $N-k$ positions are treated as frozen positions. The input block $\mathbf{u}$ of length N is encoded according to $\mathbf{x} = \mathbf{u} \cdot \mathbf{G}_N$, where $\mathbf{G} = \mathbf{F}^{\otimes n}$ is the generator matrix and $\mathbf{F}^{\otimes n}$ denotes the n^{th} Kronecker power of the kernel $\mathbf{F} = \left[\begin{smallmatrix}1 & 0\\ 1 & 1\end{smallmatrix}\right]$.

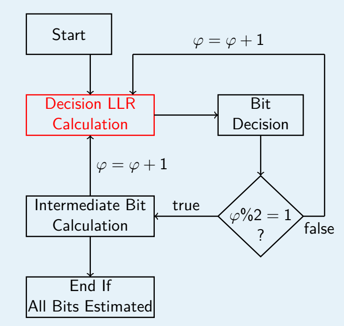
- Frozen bits u_f are set to an arbitrary value
- Encoder graph can be used to encode information bits $\boldsymbol{u}$
- Codeword $\boldsymbol{x}$ can be transmitted



Successive Cancellation Decoding

The successive cancellation decoding algorithm [1] describes an inherently serial algorithm:

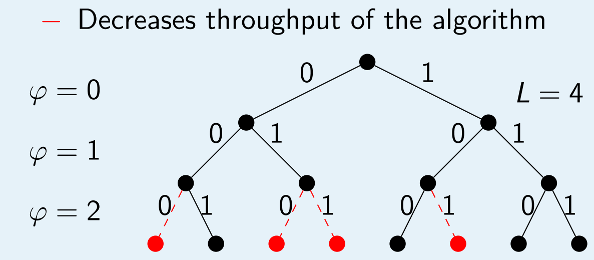
- Compute probability that current bit was 0 or 1
- Decide bit after bit
- Take all previously decided bits into consideration
- Complexity $\mathcal{O}(N \log N)$



Successive Cancellation List Decoding and CRC aided decoding

The SCL decoder utilizes the bitwise-serial decoding algorithm of the SC decoder [1] and adds a list, holding up to L of the most probable paths for the estimated codeword $\hat{\boldsymbol{x}}$ of length $N = 2^n$, resulting in a overall decoding complexity of $\mathcal{O}(L \cdot N \cdot \log(N))$ [5].

- Consider up to L of the most reliable codewords
- Instead of maximum likelihood bit decision
  - Branch every active path for $\hat{u}_\varphi = 0$ and $\hat{u}_\varphi = 1$
  - Calculate up to $2 \cdot L$ path metrics $\tilde{M}_{l,\varphi,u}$ with $u \in \{0, 1\}$
  - Keep a maximum of L paths in the list
- \+ Increases error correction capabilities of the decoder
- − Decreases throughput of the algorithm



The decoding performance can be enhanced by an additional CRC check [5]:

- For SCL: Final estimate is chosen by reliability metric M_l
- For SCL-CRC:
  - CRC is added to the sent information bits (c added CRC bits)
  - CRC is used to verify the estimates of the SCL decoder
- \+ Increases error correction capabilities of the decoder
- − Coderate is slightly reduced from $R_{\mathrm{SCL}} = \frac{k}{N}$ to $R_{\mathrm{SCL-CRC}} = \frac{k-c}{N}$
- − Additional CRC (slightly) reduces throughput

Iterative Decoding of Polar Codes

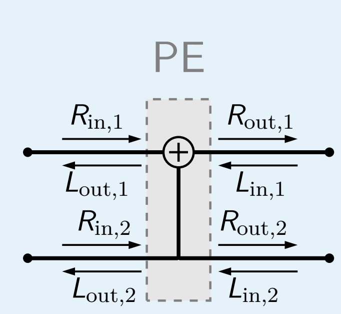
BP decoding of polar codes is a message passing algorithm based on the encoding scheme with decoding complexity $\mathcal{O}(N \cdot \log N)$. The transmitted codeword $\hat{\boldsymbol{x}}$ and the message $\hat{\boldsymbol{u}}$ can be both estimated simultaneously.

- Iterative decoder, update stage-per-stage
- $\frac{N}{2}$ processing elements (PE) per stage, n stages
- Log-likelihood ratios calculated:

$$R_{\mathrm{out},1} = g(R_{\mathrm{in},1}, L_{\mathrm{in},2} + R_{\mathrm{in},2})$$
$$R_{\mathrm{out},2} = g(R_{\mathrm{in},1}, L_{\mathrm{in},1}) + R_{\mathrm{in},2}$$

$$L_{\mathrm{out},1} = g(L_{\mathrm{in},1}, L_{\mathrm{in},2} + R_{\mathrm{in},2})$$
$$L_{\mathrm{out},2} = g(R_{\mathrm{in},1}, L_{\mathrm{in},1}) + L_{\mathrm{in},2}$$

- Initialize
  - $R_{i,1} = \begin{cases} LLR_{max}, & i \in u_f \\ 0, & \text{else} \end{cases}$
  - $L_{i,n+1} = LLR(y_i)$ LLR channel output
- Hard-decision to assign channel output
  - $LLR(\hat{u}_i) = L_{i,1} + R_{i,1}$
  - $LLR(\hat{x}_i) = L_{i,n+1} + R_{i,n+1}$

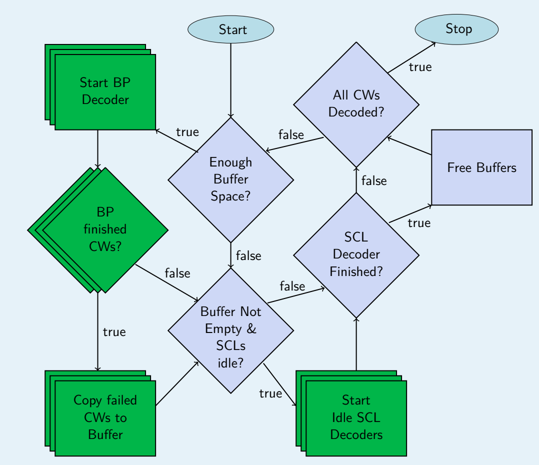


Combining BP and SCL Decoding

The bit-error-rate (BER) performance of polar codes under SCL decoding is better than that under BP decoding [6]. However, in terms of suitability of parallelization, the BP decoder shows a higher potential because all bits can be calculated in parallel while the latency can be decreased.

- BP algorithm: High throughput
- SCL-CRC algorithm: Good error correction capability
- Idea: if BP-decoder fails SCL-CRC decoder is started
  - Throughput of hybrid decoder depends on SNR
  - BER performance equals SCL performance

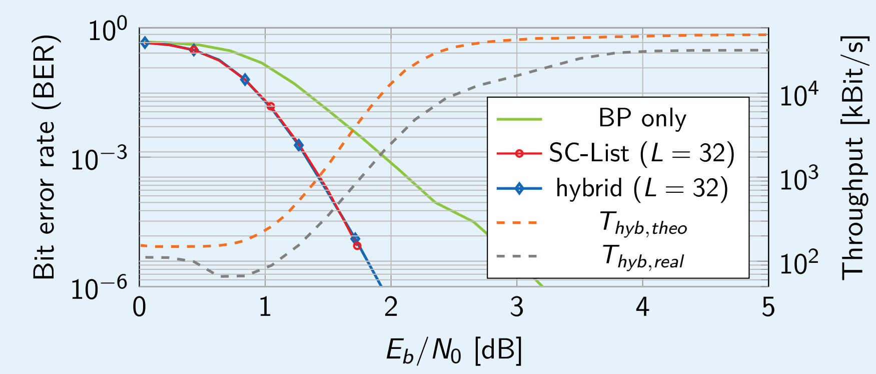
In order to avoid data transfer overhead, the whole setup is implemented on the GPU platform.



Decoding Performance

A maximum decoding throughput of 34 Mbit/s can be achieved for N=4096, L=32 and R=0.5. Additionally, it can be seen that the BER does not differ from the SCL curve.

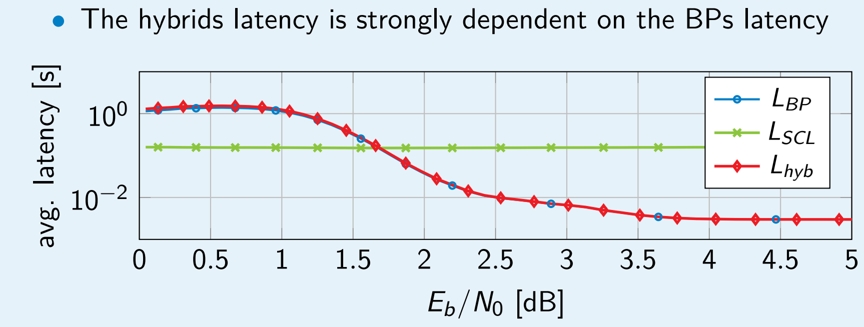
- Hybrid decoder exhibits same error correction capabilities as SCL-CRC
- Throughput the hybrid decoder depending on channel SNR



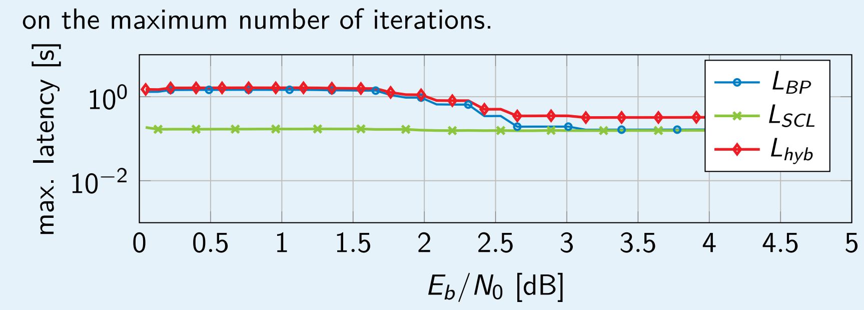
Latency

Whenever BP decoding fails, SCL decoding must be performed; then, the total latency L_{hyb} increases.

- Average latency of SCL-CRC is independent of the SNR
- The hybrids latency is strongly dependent on the BPs latency



The maximum latency of the hybrid is lower bounded by to the SCL-CRCs latency. For BP decoding the maximum latency depends on the maximum number of iterations.



When compared to other adaptive decoding concepts such as e.g., an adaptive list size [7], our approach shows better latency performance in the target SNR region.

References

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[2] P. Rost, I. Berberana, A. Maeder, H. Paul, V. Suryaprakash, M. Valenti, D. Wübben, A. Dekorsy, and G. Fettweis, "Benefits and challenges of virtualization in 5G radio access networks," vol. 53, no. 12, pp. 75–82, 2015.

[3] B. K. Reddy L. and N. Chandrachoodan, "A GPU implementation of belief propagation decoder for polar codes," in *Proc. Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2012, pp. 1272–1276.

[4] P. Giard, G. Sarkis, C. Leroux, C. Thibeault, and W. J. Gross, "Low-latency software polar decoders," *CoRR*, vol. abs/1504.00353, 2015.

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

[6] K. Niu, K. Chen, J. Lin, and Q. T. Zhang, "Polar codes: Primary concepts and practical decoding algorithms," vol. 52, no. 7, pp. 192–203, Jul. 2014.

[7] B. Li, H. Shen, and D. Tse, "An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check," *IEEE Comm. Letters*, Dec. 2012.

Summary and Outlook

- SCL-CRC decoding algorithm:
  - Not optimal for a parallel implementation
  - Small speedup observable for optimized parallel version
  - Advantage: For parallel simulations no data transfer between GPU and CPU necessary
- Hybrid decoder algorithm:
  - Achievable throughput: Up to $30\frac{\text{MBit}}{\text{s}}$
  - No degradation of error correction behaviour of SCL-CRC decoder
  - Decrease in average latency

NOKIA Bell Labs

ICASSP 2017
March 5-9, 2017
New Orleans, USA

University of Stuttgart
Institute of Telecommunications
Prof. Dr. Ing. Stephan ten Brink