RT-Polar: An HARQ Scheme with Universally Competitive Rates

Soumya Subhra Banerjee* Himanshu Tyagi*

***Abstract*—We present a construction for a universal channel code with feedback using Polar Codes. Our construction includes an error detection mechanism that is used to compute the ACK/NACK feedback directly from the received vector, without a higher layer CRC. Our scheme, termed the Repeat-Top Polar Code (RT-Polar), builds on a rate-compatible Polar Code and retransmits the t message bits sent over the most reliable polarized good channels over the least reliable good channels. At the decoder, these two t-bit strings are decoded and compared to detect an error. Through simulations, we illustrate the universal performance of our scheme for a binary symmetric channel with an unknown flipover probability. Our scheme performs comparably with a genie-aided scheme, where the detection mechanism is assumed to be error-free, for practically relevant message lengths of roughly 512 bits; this is the first instance of such a universal performance reported in literature. The proposed scheme is suitable for use as a HARQ in low-latency communication where including a higher-layer CRC will induce computational delays.**

I. Introduction

In a typical, cross-layer implementation of a Hybrid Automatic Repeat reQuest (HARQ), a set of codes designed for different noise parameters are used for forward error correction at the PHY layer. At each iteration, the decoder uses a CRC included in the higher layer header to check if the message has been correctly decoded; if not, the transmitter sends additional redundancy bits. This allows us to attain, in effect, a reliable transmission rate that is commensurate with underlying noise level. In this paper, our goal is to offer a HARQ scheme that is implementable without any separate CRC bits from a higher layer and yet offer rate performance comparable with the CRC aided scheme. In particular, we focus on the *universal* performance of the scheme for the class of binary symmetric channels $\texttt{BSC}(p)$ with unknown channel parameter p.

The theoretical problem underlying our setup is that of universal channel codes with feedback. Specifically, we seek to design channel codes with feedback without actually knowing the underlying channel. While channel codes with feedback have been analyzed thoroughly in the information theory literature, their universal counterpart that is relevant for our HARQ application has not been considered. In particular, we don't have a handle on the additional number of channel uses required, owing to universality, to send a fixed number of bits. But we do not consider this interesting theoretical problem in this work. Rather we propose a scheme which offers this universal behavior and evaluate its performance numerically.

Specifically, we present a Polar Code based construction of universal channel codes with feedback. Our scheme builds on a *rate-compatible* Polar Code and modifies it as follows: The sender retransmits the bits sent over the t least reliable polarized good channels over the top t reliable polarized good channels; as a mnemonic for this structure, we term our scheme *Repeat-Top Polar Code* (RT-Polar). At the receiver, upon decoding, these two bit-strings are compared and error is detected when they mismatch. If there is a mismatch, a NACK is sent and the rate-compatible Polar Code switches to a lower rate. We illustrate numerically that our proposed scheme achieves rates comparable with genie-aided ideal variants that assume perfect error detection or perfect channel knowledge.

HARQ design is a classic topic which has seen renewed interest in recent years motivated by ultra reliable and low latency communication requirements in the upcoming 5G standard ($cf.$ [2]). Motivated by these applications, rate-compatible Polar Codes for HARQ have been suggested; see [3], [5], [7], [8], [12]. However, we seek modifications of these schemes where no external CRC is assumed and the rate loss due to universality is accounted for in performance evaluations. We note that a CRC-free HARQ construction based on Turbo Codes has been proposed in [9]. Our proposed scheme relies on a *universal* order of Polarization; such a Polarization was introduced for general channels recently in [4], [11].

The next section contains preliminaries required to describe our scheme. Our scheme is given in Section III, and numerical results for its performance evaluation are given in Section IV. We conclude with a section on heuristic approximations for error analysis of our scheme.

II. Preliminaries

We set the stage with a brisk review of the preliminaries.

A. *Efficiency of a feedback code*

We consider channel codes with feedback where after each round of communication over the channel, the decoder sends an ACK/NACK feedback to the encoder. The encoder sends the next transmission over the channel on receiving a NACK and terminates once it receives an ACK. Denote by $\ell^*(m,\varepsilon)$ the least ℓ such that we can find a feedback code to send m message bits using no more than ℓ channel uses on average and with average (block) probability of error at most ε. A result of [10] yields $\ell^*(m,\varepsilon) = m(1-\varepsilon)/C(W) + O(\log m)$, where

This work is supported by Robert Bosch Centre for Cyber-Physical Systems, Indian Institute of Science, Bangalore.

*Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India. Email: {bsoumya, htyagi}@iisc.ac.in

$C(W)$ denotes the capacity of channel W. Motivated by this result, a benchmark of performance for a feedback code with random length N is its throughput efficiency η defined by

$$\eta = \frac{m(1-\varepsilon)}{\mathbb{E}[N]}.$$

In view of the aforementioned result, the maximum (asymptotic) efficiency $\eta^*(W)$ for any feedback code equals $C(W)$.

The information theoretic formulation above is related innately to HARQs, but does not capture the universality required of a HARQ. For such a universal setting, the benchmark of $C(W)$ for efficiency is too optimistic, and an information theoretic characterization of the optimal universally achievable efficiency is not available. In absence of such theoretical benchmark, we will compare the performance of our proposed universal scheme with its natural genie-aided variants.

B. *Polar Codes and their rate-compatible versions*

We base our construction on Arikan's Polar Codes, the first provably capacity achieving practical codes [1]. We refer the reader to [1], [6] for a description of Polar Code and review here only the components we need. We restrict our attention to the channel $\texttt{BSC}(p)$. For this channel, a Polar encoder sends the message $u \in \{0,1\}^m$ by appending $n-m$ randomly generated[1] bits, termed the *frozen bits*, to it and passing it through a linear *Polar transform*. We apply the *successive cancellation* (SC) decoder that decodes the message bit u_i by using as observation the previously decoded message bits u^{i-1}, the frozen bits, and the received bits y^n.

For use in HARQs designed for channels $\texttt{BSC}(p)$ with different values of p, we seek a rate-compatible family of Polar Codes comprising codes that operate for different values of p. Each of these codes will be active in different iterations of the decoding process, and we can switch between the codes by transmitting the redundancy bits incrementally.

Note that the Polar transform, in effect, converts n independent uses of the channel W to channels from message bits and frozen bits. At finite blocklength, the set $\mathcal{I}(p)$ of "good" channel inputs is chosen by using a threshold for the Bhattacharyya parameter $Z\big(W_n^{(i)}\big)$ of the polarized channels[2]. For the family $\{\texttt{BSC}(p), p \in (p_{\min}, p_{\max})\}$, for many popular selection methods for good channels, we have $\mathcal{I}(p) \subset \mathcal{I}(p')$ for $p > p'$. This monotonicity property is often exploited when designing rate-compatible Polar Codes and is instrumental in enabling our RT-Polar scheme.

III. RT-Polar: A universal feedback scheme

In recent years, several rate-compatible Polar Code constructions have been proposed ($cf.$ [3], [5], [7], [8], [12]).We base our construction on the scheme proposed in [7], which is provably capacity-achieving for a degraded family of channels. Consequently, our proposed scheme, too, is universally capacity-achieving for degraded channels. However, our scheme can work with the rate-compatible Polar codes for general channels proposed in [8] as well; the details are left for future work.

We review the construction from [7] first. In absence of the knowledge about the true channel statistics, the encoder initiates optimistically by sending a message at a higher rate, attained by freezing a small number of bits. If the receiver detects decoding failure, it sends a NACK to the encoder; else it sends an ACK and the transmission is complete. On receiving a NACK in feedback, the sender now moves to a less optimistic rate by successively freezing more bits and retransmitting. Starting with an initial rate R_1, the rates are successively decreased to $R_i = R_1/i$, $i = 2, 3, ..., r$, in the ith iteration, where r denotes the maximum number of iterations.

In our treatment, we restrict to BSCs with flipover probability p and apply our scheme to the set $\mathcal{G} = \{p_1, ..., p_r\}$, $p_i < p_{i+1}$, of possible values for p. For each channel $\texttt{BSC}(p_i)$ with capacity $C(p_i) = 1 - h(p_i)$, we associate a rate $R_i < C(p_i)$. In the scheme above, we choose p_is so that $R_i = R_1/i \le C(p_i)$ holds for every $1 \le i \le r$. Note that while our scheme moves from a code for $\texttt{BSC}(p_i)$ to $\texttt{BSC}(p_{i+1})$, the actual flipover probability p can be anything in the interval $[p_i, p_{i+1}]$. Thus, when evaluating the scheme, we must consider the entire interval $[p_i, p_{i+1}]$.

For this setting, the rate-compatible scheme described earlier initiates with a Polar Code of blocklength n for transmitting $k = nR_1$ information bits; in accordance with the foregoing discussion, we set $R_i = k/(n \cdot i)$. We choose the set $\mathcal{I}_i$ of good channels used in the ith iteration to satisfy $|\mathcal{I}_i| = n.R_i = m/i$. Since our underlying set of channels forms a degraded family, we can find $\mathcal{I}_i$s such that $\mathcal{I}_{i+1} \subset \mathcal{I}_i$ for $1 \le i \le r$ (see [6]). Heuristically, the scheme above proceeds by making an optimistic guess $\texttt{BSC}(p_i)$ for the channel in iteration i. If there is a decoding error, our guess is deemed incorrect, and we update it to $\texttt{BSC}(p_{i+1})$. The procedure stops when we have either decoded correctly or have used r iterations.

We build-on the scheme above by introducing an error detection mechanism in the code. Recall that in a Polar Code, the information bits are sent over the good channels, namely those polarized channels $W_n^{(i)}$ for which $Z(W_n^{(i)})$ is below a threshold δ. In fact, these good channels can be ordered in the increasing order of $Z(W_n^{(i)})$, with the lower $Z(W_n^{(i)})$ corresponding to more reliable channels [7]. In our RT-Polar error detection mechanism, we take the t-bit block of our k-bit message that is to be transmitted over the good channels with t smallest values of Z and resend it over the good channels with t largest values of Z.

At the receiver, upon decoding the message, the top t bits (most reliable) are treated as the "hash" of the bottom t (least reliable) message bits. Specifically, denoting by H the decoded value of the top bits and by $\hat{H}$ the decoded value of the bottom bits, we accept our decoding only if $H = \hat{H}$. On the one hand, with significant probability, the top bits are received without an error for all flipover probabilities p in the range considered. On the other hand, it is unlikely that the bottom

[1]In implementation, we have set the frozen bits to 0.

[2]For ease of implementation, we use the more explicit construction PCC-0 from [13, pg. 4] to select the set of good channels.

bits are decoded correctly before we have guessed the correct p. Based on this heuristic, in each iteration of our scheme, we obtain H_i and $\hat{H}_i$ from the decoded codeword. In view of the foregoing discussion, we do not expect H_i and $\hat{H}_i$ to coincide before the iteration i corresponding to the true p.

The complete description[3] is given in Algorithm 1 and Algorithm 2; the variables and subroutines involved are explained in the subsequent text.

Algorithm 1 RT-Polar encoder at the ith iteration.

1: **Input:** the message vector u and ACK$_{i-1}$/NACK$_{i-1}$ from previous iteration
2: **Output:** the retransmission information vector v_i and the encoded retransmission vector c_i
3: $v_i \leftarrow \emptyset$
4: **if** $i = 1$ **then**
5: $\quad H \leftarrow \text{extract}(u, t, \texttt{TOP})$
6: $\quad v_i \leftarrow \text{append}(u, H)$
7: **else if** ACK$_{i-1}$ **or** $i = r + 1$ **then** terminate
8: **else if** NACK$_{i-1}$ **then**
9: $\quad$ **for** j=1, 2, …, i-1 **do**
10: $\quad\quad \alpha_j \leftarrow \text{extract}(v_j, nR_{i-1}, \texttt{TOP})$
11: $\quad\quad \beta_j \leftarrow \text{extract}(\alpha_j, n(R_{i-1} - R_i), \texttt{BOTTOM})$
12: $\quad\quad v_i \leftarrow \text{append}(v_i, \beta_j)$
13: $\quad c_i \leftarrow \text{encode}(v_i, nR_i)$
14: **return** (v_i, c_i) $\qquad \triangleright$ c_i is sent over the channel

Algorithm 2 RT-Polar decoder at the ith iteration.

1: **Input:** the received vector y_i
2: **Output:** ACK$_i$/NACK$_i$ and the decoded vector $\hat{u}$
3: $f_i \leftarrow \text{zeros}(n - nR_i)$
4: $\hat{v}_i \leftarrow \text{decode}(y_i, f_i, nR_i, p_i)$
5: **for** j = $i-1$, $i-2, \ldots$,1 **do**
6: $\quad f_j \leftarrow \text{zeros}(n - nR_j)$
7: $\quad \rho_j \leftarrow \emptyset$
8: $\quad$ **for** l=i, $i-1$, …, $j+1$ **do**
9: $\quad\quad \rho_j \leftarrow \text{append}(\rho_j, \text{Beta}(j, \hat{v}_l))$
10: $\quad f_j \leftarrow \text{append}(\rho_j, f_j)$
11: $\quad \hat{v}_j \leftarrow \text{decode}(y_j, f_j, nR_j, p_i)$
12: $\hat{u} \leftarrow \text{extract}(\hat{v}_1, nR_1, \texttt{TOP})$
13: $H_i \leftarrow \text{extract}(\hat{v}_1, t, \texttt{TOP})$
14: $\hat{H}_i \leftarrow \text{extract}(\hat{v}_1, t, \texttt{BOTTOM})$
15: **if** $H_i = \hat{H}_i$ **then**
16: $\quad$ **return** ACK, $\hat{u}$
17: **else**
18: $\quad$ **if** $i < r$ **then**
19: $\quad\quad$ **return** NACK,
20: $\quad$ **else return** $\hat{u}$

The subroutine append(a, b) used in Algorithm 1 appends the vector b to the vector a, and the extract$(a, l, \texttt{POS})$ with $\texttt{POS}$ equal to $\texttt{TOP}$ or $\texttt{BOTTOM}$, respectively, extracts the top or bottom l bits from the vector a. Here we have followed the convention that the bits are placed top-to-bottom in the increasing order of the Bhattacharyya parameters of the corresponding channels they will be sent over. The vector β_j in line 11 of Algorithm 1 denotes the set of bits that were transmitted in jth iteration mistakenly due to optimistically guessing p and need retransmission in the ith iteration.The subroutine encode(u, nR) denotes the Polar encoder of rate R and length n, decode(y, f, nR, p) the SC decoder for rate R and $\texttt{BSC}(p)$ applied to the received vector y with frozen bits f, and zeros(ℓ) the length ℓ zero vector.

After decoding the vector received at the ith iteration, all the vectors received in previous iterations are decoded recursively considering the bits transmitted in subsequent iterations as frozen. The lines 5 to 11 of Algorithm 2 denote the same, wherein the subroutine Beta$(j, \hat{v}_l)$ in line 9 returns the bits that correspond to β_j in line 11 of Algorithm 1 (chosen at the lth iteration during encoding), decoded under the impression that the channel is $\texttt{BSC}(p_i)$.

IV. Numerical evaluations

When applying this scheme, we need to select the *check-length* t, the blocklength n, and the initial rate R_1. We choose these parameters to "optimize" the throughput $\eta(p)$, which for our scheme is given by

$$\eta(p) = \frac{m(1 - \varepsilon_p)}{n\mathbb{E}_p\{I\}},$$

where p denotes the flipover probability to be used in probability calculations, $m = nR_1 - t$ is the number of message bits sent, I is the random number of iterations used, and ε_p is the average probability of error under $\texttt{BSC}(p)$.

For most of this section, we restrict to the simple case of $r = 3$. Specifically, we consider the previous scheme with $p_1 = 0.03$, $p_2 = 0.11$, $p_3 = 0.17$. These probabilities correspond to $\texttt{BSC}(p)$ with capacities roughly $4/5$, $1/2$, and $1/3$. These values have been chosen to allow us to accommodate reasonable rates $R_1, R_2 = R_1/2, R_3 = R_1/3$. In our Polar Code construction, the set of good bit channels $\mathcal{I}_{n,p}$ is selected to ensure that $\max\{Z(W_n^{(j)}(p)) : j \in \mathcal{I}_{n,p}\} \leq \delta$, where $\delta = 0.05$ corresponds to $R_1 = 1/2$ approximately.

We begin with simulations to illustrate the trade-off of throughput with check-length t for $n = 1024$, which can be used to select a reasonable t. In doing so, we focus only on the throughput $\eta(p)$ achieved by our scheme for $p = p_i$, $i = 1, 2, 3$ since we switch between Polar Codes designed only for these values of p. Specifically, in Figure 1, we have evaluated the throughput $\eta(p_i)$, $i = 1, 2, 3$, for values of t in the interval $[1, nR_3]$. Note that along with the simulated behavior, the figure also includes an analytic approximation which follows closely the trend of the simulated curve and will be discussed in Section V. We denote the (numerically) optimum check-length t for p_i as t_i^*. As can be seen from Figure 1, $t_1^* = 1$, $t_2^* = 9$, $t_3^* = 17$. Choosing $t = t_1^*$ will result in the best η for $\texttt{BSC}(p_1)$, but η for $\texttt{BSC}(p_2)$ and $\texttt{BSC}(p_3)$ will

[3]The lines 9-13 in Algorithm 1 and 4-8 in Algorithm 2 are the same as the scheme of [7].

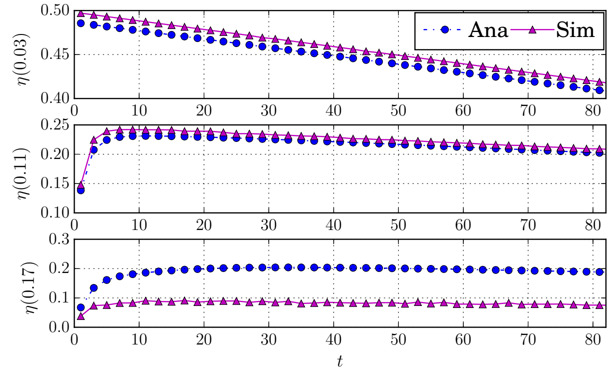

Fig. 1. Throughput $\eta(p)$ for RT-Polar scheme for $\texttt{BSC}(p)$, $p \in \{0.03.0.11, 0.17\}$, $n = 1024$, $\delta = 0.05$, $t \in [1, nR_3]$.

suffer considerably. However, by choosing $t = t_2^*$ we achieve the best η for $\texttt{BSC}(p_2)$ with negligible loss for $\texttt{BSC}(p_1)$. This choice exploits the behavior of $\eta(p)$ as a function of t, for a fixed p, where it rises rapidly with t to it's maximum but decays slowly thereafter. Furthermore, there is no advantage in including the $\eta(p_3)$ trend in deciding the best t since it does not vary much with increasing t.

Next, we illustrate through simulations the trade-off of η with the parameter δ. A smaller δ allows a smaller R_1 which leads to a conservative design that enhances the performance for worse channels but penalizing the better channels. On the other hand, we observe that choosing δ to be large affects the worse channels adversely. In Figure 2, we compare the performance of our scheme for $\delta = 0.5, 0.05, 0.005$, where each value of δ is used with the corresponding optimal value of t identified in the manner described above. We note from Figure 2 that $\delta = 0.05$ constitutes the most reasonable choice. This choice is seen to be reasonable for other blocklengths as well, although we omit these simulations due to lack of space.

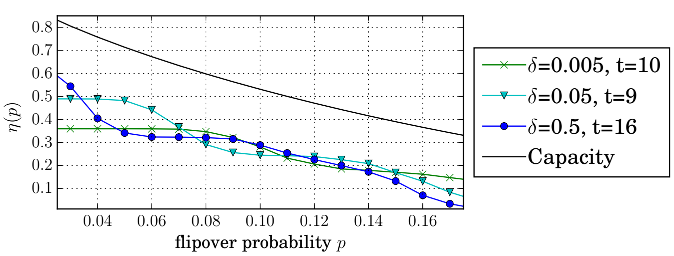

Fig. 2. Effect of varying δ on the throughput $\eta(p)$ for RT-Polar for $n = 1024$ and value of t optimized for each δ.

Finally, we illustrate the trade-off of throughput with code-length n and then choose optimal n. In Figure 3, we have shown $\eta(p)$ for $n \in \{64, 128, 256, 512, 1024, 2048\}$ with $\delta = 0.05$ and t optimized using the procedure described earlier. From our simulations, it is evident that given a δ, higher blocklengths achieve better throughput for better channels, but for worse channels the performance of smaller blocklengths takes over.

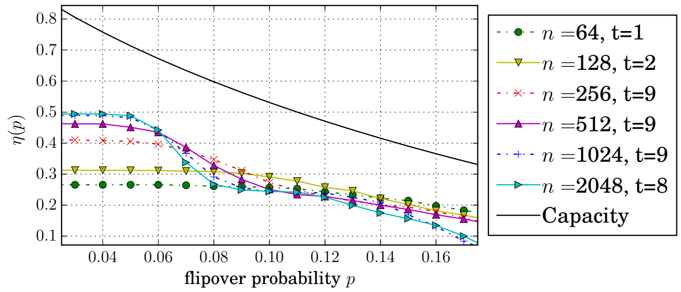

Fig. 3. Effect of n on $\eta(p)$ for RT-Polar scheme with t optimized for $\texttt{BSC}(p)$, $p \in \{0.03.0.11, 0.17\}$, $\delta = 0.05$

Based on these simulations, we have selected $\delta = 0.05$, $n = 512$, and $t = 9$ as the best choice of parameters to obtain a fair throughput for the entire range $[p_1, p_3]$. Figure 4 illustrates the performance of RT-Polar scheme with this choice of parameters. It provides a comparison of the RT-Polar scheme with the ideal case where the receiver has an oracle to detect decoding failures without an error (this is essentially the assumption in prior work). Our scheme compares closely with this ideal case with an additional throughput-loss of roughly t/n. In addition, we compare our scheme with the throughput of fixed rate Polar Codes[4] designed with the knowledge of $p \in [p_1, p_3]$, and SC decoding.

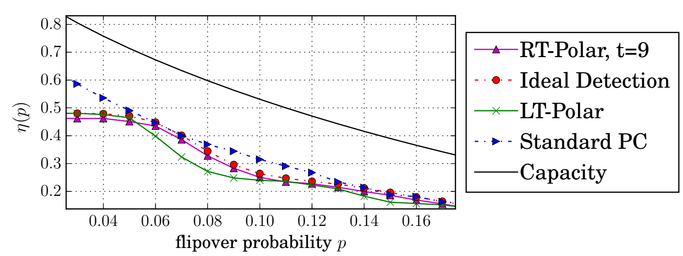

Fig. 4. Performance comparison of RT-Polar with $t = 9$, $\delta = 0.05$, $n = 512$.

Remarkably, the performance of RT-Polar code is comparable with these ideal cases where either perfect error detection or perfect channel knowledge is assumed, especially near the values $p = p_i$ for which we have optimized the scheme.

Additionally, in Figure 5 we illustrate that our RT-Polar scheme fares comparable with an extension of Polar Codes that uses a CRC of length t computed from the entire message.

Both CRC-based implementation and our RT-Polar code entail a rateloss of t-bits due to inclusion of check-bits for error detection. Alternatively, one can design an error detection mechanism based on the log-likelihoods computed at the Polar decoder. We close this section with description of one such scheme that we have devised. For brevity, we only give an informal description of this scheme, termed the *loglikelihood-threshold Polar* (LT-Polar) scheme. The LT-Polar scheme,

[4]For fixed rate Polar Code of rate R, we use $\eta = R(1 - \varepsilon)$.

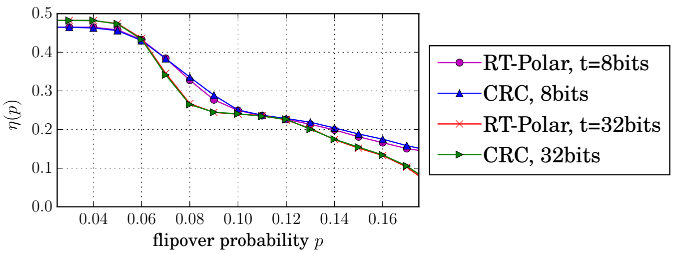

Fig. 5. Comparison of RT-Polar for $\delta = 0.05$, $n = 512$ with a CRC based scheme.

too, uses a rate-compatible Polar Code except that the error detection mechanism is now changed. Specifically, we accept the current guess $p = p_i$ if and only if the fraction of good channels with absolute log-likelihood ratios above a threshold λ is more than θ. Figure 4 above compares the performance of this LT-Polar scheme with the RT-Polar scheme by choosing λ and θ appropriately from simulations. It can be seen that RT-Polar scheme outperforms the LT-Polar scheme.

V. Approximate throughput analysis optimization

In this final section, we present an approximate analysis of the throughput that can be used to select the check-length t. A mathematically rigorous analysis of the gap-to-capacity at a finite blocklength that takes into account the loss due to universality is unavailable. Nevertheless, we present a simple handle over the performance using heuristic simplifications which yield a choice of optimal t that matches the choice based on simulations in the previous section.

Recall that the throughput of our scheme when the underlying channel is $\texttt{BSC}(p)$ is given by $\eta(p) = m(1-\varepsilon)/(n\mathbb{E}_p\{I\})$. For simplicity we only focus on the case $p \approx p_i$. To obtain tractable bounds, we assume first that for $p = p_i$, the probabilities $\mathrm{P}_p(I < i-1)$ and $\mathrm{P}_p(I > i+1)$ are negligible. Under this assumption, the throughput depends only on the performance of the error detection mechanism for $p = p_i$. Also, we approximate further the error detection mechanism by a simple binary hypothesis testing problem[5] with null hypothesis denoting the case $p = p_i$ and the alternative $p = p_{i+1}$. Denote by $P_{F,i}$ and $P_{M,i}$, respectively, the probabilities of false alarm $\mathrm{P}_{p_i}\left(H_i \neq \hat{H}_i\right)$ and missed detection $\mathrm{P}_{p_{i+1}}\left(H_i = \hat{H}_i\right)$. Using these notations and assumptions, we can approximate the expected number of iterations $\mathbb{E}_{p_i}\{I\}$ as

$$(i-1)P_{M,i-1} + (1 - P_{M,i-1})(i(1 - P_{F,i}) + (i+1)P_{F,i}),$$

and the average probability of error ε as

$$P_{M,i-1}\varepsilon_{i,i-1} + (1 - P_{M,i-1})((1 - P_{F,i})\varepsilon_{i,i} + P_{F,i}\varepsilon_{i,i+1}),$$

where $\varepsilon_{i,\ell}$ denotes the probability of error under $\texttt{BSC}(p_i)$ when we stop at iteration ℓ. It only remains to estimate $P_{F,i}$, $P_{M,i}$, and $\varepsilon_{i,\ell}$ for $\ell = i-1, i, i+1$. To that end, we make another assumption that the top bits are sent error-free for our entire range of p. Indeed, if this is not the case, reliable transmission will not be possible at all in our range. Then, the probability $P_{F,i}$ approximately equals $\mathrm{P}_{p_i}\left(\text{one of } \hat{H}_i \text{ bits is flipped}\right)$ which is bounded above by $\sum_{j=nR_i-t}^{nR_i} Z_{i,j}$, where $Z_{i,j}$ denotes $Z\big(W_n^{(j)}\big)$ for $W = \texttt{BSC}(p_i)$. Moving next to $P_{M,i}$, note that this error happens roughly when all the bottom t bits get correctly decoded, an event very unlikely under p_{i+1}. Since the probability of each check bit being erroneously decoded here is close to 1, union bound will not be useful here. Instead, we proceed with our final approximation: We assume the errors for the bits of $\hat{H}_i$ are independent of the other bits and independent of each other. That is, $P_{M,i} \approx \prod_{j=nR_i-t}^{nR_i} \mathrm{P}_{p_{i+1}}(\text{bit } j \text{ is sent correctly})$ where the terms in the product are no more than $0.5\big(1 + \sqrt{1 - Z_{i+1,j}^2}\big)$ each. Furthermore, using the assumed independence of the error in bits $\hat{H}_i$ and the remaining message bits, we approximate $\varepsilon_{i,i-1} \lesssim 1$ and $\varepsilon_{i,\ell} \lesssim \ell \sum_{j=1}^{nR_\ell} Z_{i,j}$ when $\ell = i, i+1$. Using these approximations, we can evaluate the efficiency $\eta(p_i)$ using only the parameters $Z_{i,j}$. In Figure 1, we have compared this heuristic analytical approximation for $\eta(p_i)$, $i = 1, 2, 3$, with simulated values. The two set of curves match roughly in their prescription for optimal choice of t.

[5]This, too, is a simplistic approximation; the actual problem is composite.

Acknowledgment

We thank Krishna Narayanan for the suggestion to use a CRC for obtaining a universal channel code with feedback and other illuminating discussions related to this work.

References

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

[2] G. Berardinelli *et al.*, "Enabling early HARQ feedback in 5G networks," in *IEEE 83rd Vehicular Technology Conference*, 2016.

[3] K. Chen, K. Niu, and J. Lin, "A hybrid ARQ scheme based on Polar Codes," *IEEE Commun. Letters*, vol. 17, no. 10, pp. 1996–1999, 2013.

[4] S. H. Hassani and R. Urbanke, "Universal polar codes," in *IEEE Int. Symposium on Inform. Theory, 2014*. IEEE, 2014, pp. 1451–1455.

[5] S.-N. Hong, D. Hui, and I. Marić, "Capacity-achieving rate-compatible polar codes," *IEEE Trans. Inform. Theory*, vol. 63, no. 12, pp. 7620–7632, 2017.

[6] S. B. Korada, "Polar codes for channel and source coding," *Ph.D. thesis, Swiss Federal Inst. Technology (EPFL)*, 2009.

[7] B. Li, D. Tse, K. Chen, and H. Shen, "Capacity-achieving rateless polar codes," in *IEEE Int. Symposium on Inform. Theory*, 2016, pp. 46–50.

[8] M. Mondelli, S. H. Hassani, I. Maric, D. Hui, and S.-N. Hong, "Capacity-achieving rate-compatible Polar Codes for general channels," in *Wireless Communications and Networking Conference Workshops, 2017*. IEEE, 2017, pp. 1–6.

[9] H. Mukhtar, A. Al-Dweik, and M. Al-Mualla, "CRC-free hybrid ARQ system using Turbo Product Codes," *IEEE Trans. Commun.*, vol. 62, no. 12, pp. 4220–4229, 2014.

[10] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Variable-length coding with feedback in the non-asymptotic regime," in *IEEE Int. Symposium on Inform. Theory*, 2010, pp. 231–235.

[11] E. Şaşoğlu and L. Wang, "Universal polarization," *IEEE Trans. Inform. Theory*, vol. 62, no. 6, pp. 2937–2946, 2016.

[12] S. R. Tavildar, "A H-ARQ scheme for Polar Codes," *arXiv:1606.08545*, 2016.

[13] H. Vangala, E. Viterbo, and Y. Hong, "A comparative study of Polar Code constructions for the AWGN channel," *arXiv:1501.02473*, 2015.