Communication Efficient Data Exchange Among Multiple Nodes

Midterm Report
EP 299: Project
for
M.Tech, Communication and Networks

by

Soumya Subhra Banerjee
SR No. 04-02-04-37-42-16-1-14191

Under the guidance of

Himanshu Tyagi,
Assistant Professor

Department of Electrical Communication Engineering
Indian Institute of Science
Bangalore - 560 012

January 2018

Contents

List of Figures

Abstract

Multiple parties observing correlated data seek to exchange their data using minimum amount of communication. In case the underlying joint distribution is known, a rate optimal solution to this problem is offered by Slepian-Wolf compression. In absence of this knowledge, interactive communication is necessary to attain universal optimality. In both the cases, decoding is of exponential complexity. However, Slepian-Wolf compression can be implemented using structured channel codes. In this project, we design rateless Polar codes to enable universal Slepian-Wolf compression and data exchange, when the underlying distribution generating the data is unknown.

A key observation driving our project is the similarity between the interactive Slepian-Wolf compression and the design of an Hybrid ARQ protocol for communication over a network. Unlike H-ARQ, in our setting, the error-detection capability offered by higher layers is not available. Hence, we propose a PHY-layer error-detection scheme based on soft outputs of the Successive Cancellation decoder of Polar codes. We present simulation results that exhibit the performance of the considered methods. Besides offering a solution to our problem, this scheme provides a CRC-free Rateless Polar Code which promises rate gains for short packet length communication and may be of independent interest.[1]

[1] This project is supported by Robert Bosch Center for Cyber Physical Systems

1 Introduction

Random correlated data (X,Y) is distributed between two parties with the first observing X and the second observing Y. The two parties seek to recover each others data. The *Data-Exchange* problem depicted in figure 1, encompasses this scenario. This project seeks to device a practical protocol which achieves this with minimal communication under a setting where the joint distribution of X and Y is unknown.

A working solution to this problem is the r-sync protocol, as described in [1]. The algorithm identifies parts of the source file which are identical to some part of the destination file, and only sends those parts which cannot be matched in this way. Though this protocol is fast and has low complexity, it does not exploit the correlation of the data to the best extent possible. In fact, we can view r-sync as an algorithm which guesses the source and the destination to be same and then communicates to confirm the guess.
More generally, the destination may form a list of guesses and the source may communicate to resolve the guess. This is precisely captured by Slepian-Wolf compression in [2], which offers a rate optimal solution to the problem.

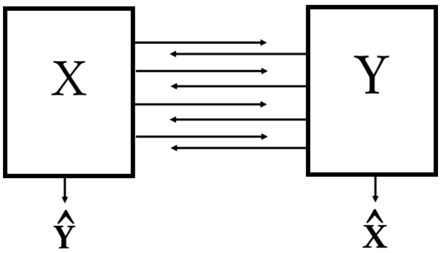


Figure 1: The Data-Exchange problem

Slepian-Wolf compression: As described in [3], consider first the problem where X and Y are correlated discrete-alphabet memoryless sources, we have to compress X losslessly, with Y (side information) being known at the decoder and *not* at the encoder. If Y were known at both ends one can compress X at a theoretical rate of $H(X|Y)$. But if Y were known only at decoder the same can be achieved by just knowing $P_{X|Y}$ at encoder without explicit information of Y, this has been depicted in figure 2. Thus for communicating X to the second party a rate of $H(X|Y)$ is sufficient.

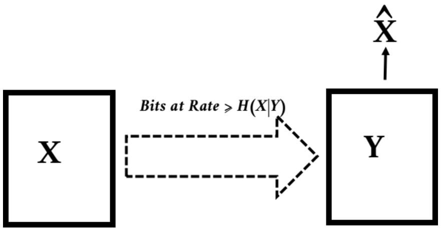


Figure 2: The Slepian-Wolf compression

A practical implementation of Slepian-Wolf compression faces the following difficulties.

- Search is over an exponential list in decoding.
- Knowledge of $P_{X|Y}$ is required.

In this project we circumvent this difficulty by implementing Slepian-Wolf compression using structured channel codes such as Polar Codes as shown in [3],[4]. We use the *recursive data exchange* (RDE) protocol mentioned in [5] for the case the where underlying joint distribution is unknown. Note that in [5] the decoders suggested for the protocol are theoretical constructs and are not practically implementable. We use Rateless Polar Codes for the implementation of RDE.

Suggested Approach for Solving The Data-Exchange Problem

In accordance with the above discussion the suggested approach towards solving the *Data-Exchange* problem may be briefed as follows.

- Implement SW Compression using Polar Codes.
- Achieve universality using *recursive data exchange* protocol (RDE).
- Realize RDE using Rateless Polar Codes with physical layer error detection.

Interestingly, this approach is closely related to the construction of an Hybrid ARQ with error detection in the PHY layer.
In section 2 we provide the background required to build our proposal. Section 3, consolidates and elaborates the proposed scheme. Performance analysis of the scheme is presented in section 4. We conclude with final remark and future work in section 5.

2 Background

In this section we discuss the the artifacts needed for the suggested implementation in brief.

2.1 Recursive Data Exchange Protocol

The data exchange protocol is based on an interactive version of the SW protocol where the length of communication is increased in steps until the second party decodes the data of the first. After each transmission second party sends ACK-NACK feedback signal, the protocol stops when ACK is received or some fixed number of bits have been transmitted [5]. Note, this protocol is universal as it does not rely on knowledge of the joint distribution, instead uses an iterative variable length approach to reach rate optimality universally. The decoders suggested in [5] are theoretical constructs which use type classes to form a list of guesses for data of other parties and thus has exponential complexity.
In [4], SW compression is approached with Polar Codes, in this work we use Rateless Polar Codes to implement RDE with a similar ideology.

2.2 Brief Introduction to Polar Codes

In 2008, E. Arikan in his paper [6] introduced Polar Codes, which provably achieves capacity on symmetric channels. Polar Codes rely on the phenomenon of channel polarization which can be described as follows.

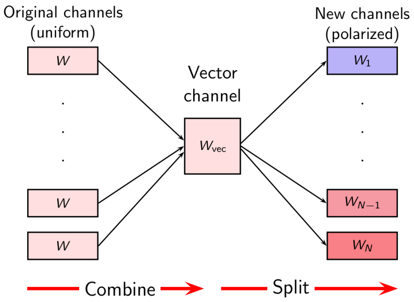


Figure 3: Channel combining and splitting

Channel Polarization: It is an operation by which one manufactures out of N independent copies of a given B-DMC (W), a second set of N channels $\{W_N^{(i)} : 1 \leq i \leq N\}$ that show a polarization effect in the sense that, as N becomes large, the symmetric capacity terms $I(W_N^{(i)})$ tend towards 0 or 1, for all but a vanishing fraction of indices i. Hence, the Bhattacharya parameter $Z(W_N^{(i)})$ tends to 1 or 0 respectively. The channels with $Z(W_N^{(i)}) = 0$ captures the capacity of W_{vec}.

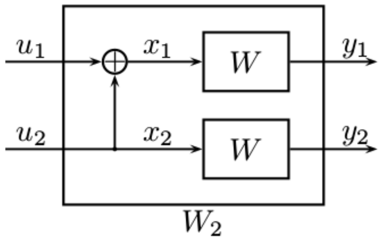


Figure 4: Arikan transformation butterfly

This operation consists of a channel combining and a channel splitting phase, as shown in figure 3. The channel transformation for two independent channels is shown in 4, which is used recursively. The encoding process sends data on transformed channels with $Z(W_N^{(i)}) = 0$ (*good channels*) and treats the channels with $Z(W_N^{(i)}) = 1$ as *frozen*, sending no useful data on them. This scheme of error control coding with Polar Codes is illustrated in figure 5. In figure 5, U_N is a uniform message vector, T is a linear transform equivalent to the butterfly in figure 4 for a block length of N. W_N is analogous to W_{vec}. It is useful to note here that $T = T^{-1}$. In figure 5, U_N is a uniform message vector, T is a linear transform equivalent to the butterfly in figure 4 for a block length of N. W_N is analogous to W_{vec}. It is useful to note here that $T = T^{-1}$. For our purpose, we shall be using Successive Cancellation (SC) decoding. Each independent channel will considered as BSC(p). The Polar Code construction indicated in [7] has been employed for simplicity. A performance analysis of implementation of the scheme in figure 5 is presented in figure 6.

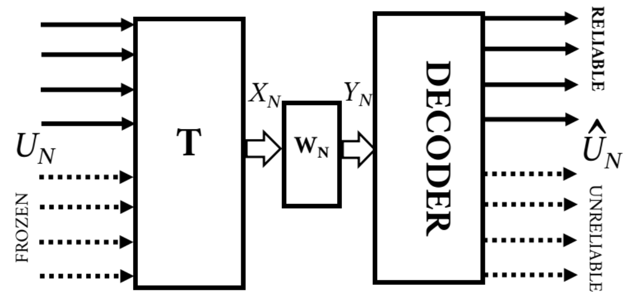


Figure 5: Polar Coding

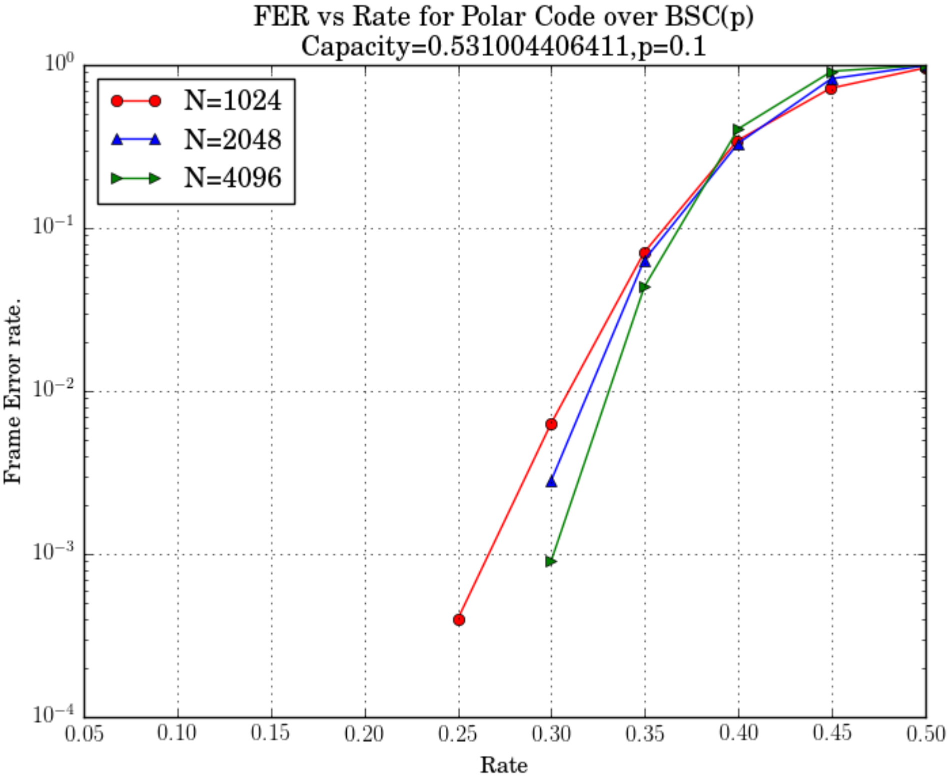


Figure 6: FER vs Rate for Polar Coding with SC-decoding

2.3 Implementation of SW Compression using Polar Codes

In [3], the use of structured codes for SW compression has been discussed. Further, in [4] a specific scheme using Polar Codes has been illustrated, as shown in figure 7.

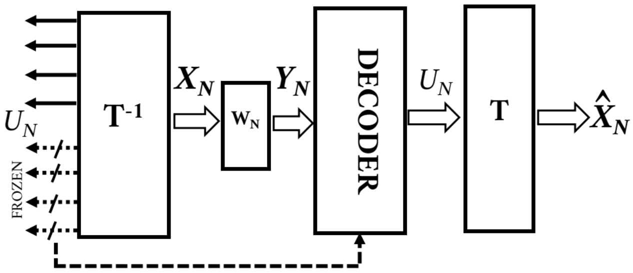


Figure 7: Polar Coding for SW compression

Consider the setting where X_N and Y_N are uniform. Y_N is a corrupted version of X_N by N BSC(p) channels. The bits that are to be sent for estimation of X_N from Y_N are the frozen bits in U_N. In other words applying T^{-1} to X_N and choosing the bits at frozen positions (the syndrome) essentially describes the compression operation. These bits are communicated error free to the SC-decoder. Thus $H(X_N) - I(W_N) = H(X_N/Y_N)$ bits are sent. On applying T to the output of SC-decoder the estimate $\hat{X}_N$ is received. It is to be noted here that in general the frozen positions are set to 0 in channel coding but in SW compression they are non-zero by the nature of the scheme. Performance analysis of this scheme is presented in figure 8.

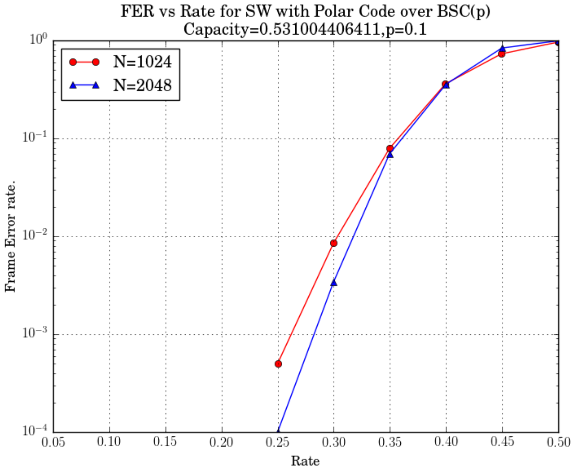


Figure 8: FER vs Rate for SW compression with Polar Code

Intuitively, a method to generate the syndrome incrementally will lead us to an implementation of RDE. Rateless Polar Codes will be instrumental to accomplish this.

2.4 Rateless Polar Codes

Rateless Code: A rateless coding scheme transmits incrementally more and more coded bits over an unknown channel until all the information bits are decoded reliably by the receiver. A fixed rate code is designed for a specific channel. In contrast, a rateless code is designed for a set of channels and judged for its performance for the entire set (the compound channel). In general a rateless code design is based on Hybrid-ARQ (HARQ) techniques and uses code puncturing.
For Polar Codes, puncturing is not straightforward. HARQ schemes and puncturing of Polar Codes has been proposed and compared in [8], [9] and [10].
In [11], the authors have proposed a provably capacity achieving rateless coding scheme based on Polar Codes. This scheme is useful for broad class of channels as long as they are ordered by degradation[2]. The scheme stems from the inherent nesting property and degradedness of Polar Codes.

2.4.1 Degradedness and Nesting Property

Degraded Channels: A symmetric binary input channel W_2 is said to be degraded with respect to a channel W_1 if there exists random variables X, Y, Z such that X–Y–Z forms a Markov chain and $W_1 = P_{Y|X}$, $W_2 = P_{Z|X}$. By data processing inequality it is evident that the capacity of W_2 is lesser than that of W_1. This is denoted by $W_2 \preceq W_1$. For example, $BSC(p_1) \preceq BSC(p_2)$ if $p_1 \geq p_2$.

Nesting Property: Polarization suggests that $W_2 \preceq W_1$ will reflect as lesser number of *good channels* for W_2. As the polarization operation preserves degradedness [12], the good bit indices of W_2 must be a subset of the good bit indices of W_1. This is nesting property.

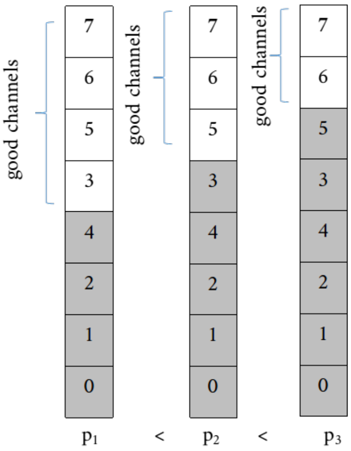


Figure 9: Nesting in BSC(p) channels

[2] using construction methods in [12], [13] this method can be extended to a broader class of *less noisy* ordered channels

This leads to a *reliability ordering* of the bit-channels, such that a more reliable bit-channel is always noiseless if a less reliable bit-channel is noiseless regardless of the underlying channel.

2.4.2 Incremental Freezing

Given the reliability ordering, the rateless scheme in [11] can be described as follows. The initial transmission can be done using a high rate Polar Code with many information bits and less frozen bits. If this transmission cannot be decoded[3] then among the informations bits sent, the ones on comparatively lesser reliable channels are retransmitted. By decoding these bits from future transmissions they effectively become frozen, allowing the rest of the information bits sent on the first transmission to be decoded. As future transmissions successively freeze more and more bits sent in earlier transmission, this scheme can be called *Incremental Freezing* (Inc-Frz).

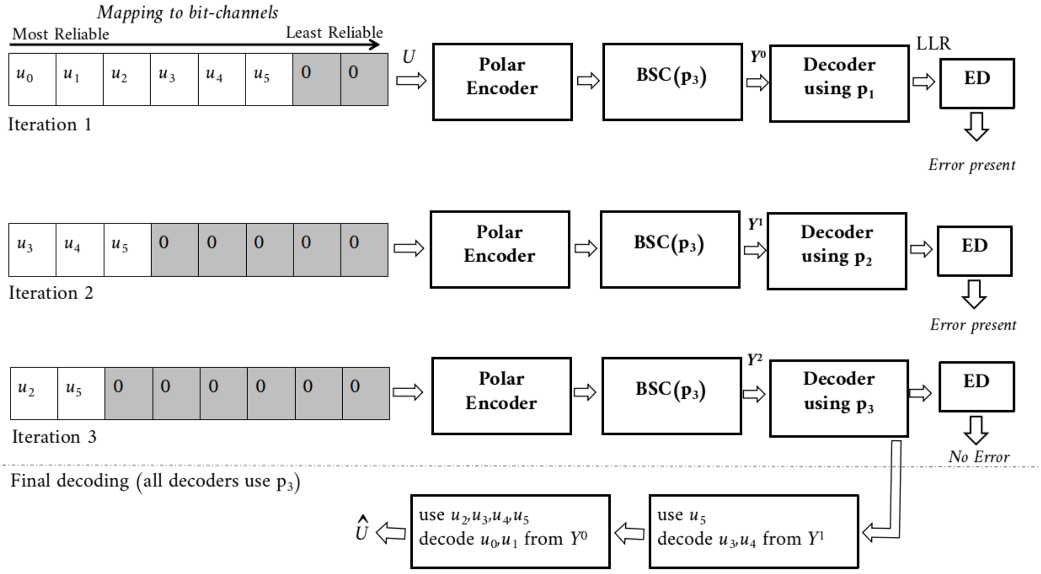


Figure 10: Incremental Freezing for $N = 8$, $K = 6$ and 3 iterations

Figure 10 illustrates the scheme for a set of channels with rates $\{R_1 = 6/8, R_2 = R_1/2 = 3/8, R_3 = R_1/3 = 1/4\}$. Here u_i are the message bits. Note that with the

[3] Decodability is checked by CRC in higher layers.

3^{rd} transmission u_2 to u_5 have been incrementally frozen. This scheme is capacity achieving in the sense that no rate has been wasted, the final rate achieved is,

$$R^* = \frac{6}{8 * 3} = \frac{1}{4} = R_3$$

In case the real channel has capacity $\geq R_3$ the scheme stops at appropriate iteration to achieve that capacity.

Though this scheme is not truly rateless as it can achieve only R, $R/2$, $R/3$,... rather than a set of arbitrary rates, adding new information bits in future transmissions allow us to rectify this. In similar fashion this can be extended to parallel channels.

It is useful to note that a certain number of channels in this scheme is "*always available*" guaranteeing a certain rate in each transmission. An application of this will be discussed in section 5.

In [11], The performance evaluation of the scheme establishes that n iterations of the scheme is almost equivalent in performance to a R/n fixed rate Polar Code.

Adaptation of this scheme for RDE will be indicated in section 3.

The following subsections briefly discuss Rateless Polar Coding from the perspective of HARQ.

2.4.3 HARQ for Polar Codes

Hybrid ARQ: Hybrid automatic repeat request (hybrid ARQ or HARQ) is a combination of high-rate forward error-correcting coding and ARQ error-control. In standard ARQ, redundant bits are added to data to be transmitted using an error-detecting (ED) code such as a cyclic redundancy check (CRC). Receivers detecting a corrupted message will request a new message from the sender.

In Hybrid ARQ, the original data is encoded with a forward error correction (FEC) code, and the parity bits are either immediately sent along with the message (Type-I) or only transmitted upon request when a receiver detects an erroneous message (Type-II). The retransmission vector is also known as the *Redundancy Vector* (RV).

The ED code may be omitted when a code is used that can perform both forward error correction (FEC) in addition to error detection, such as a Reed-Solomon code or Turbo Product Code (TPC) [14].

The FEC code is chosen to correct an expected subset of all errors that may occur, while the ARQ method is used as a fall-back to correct errors that are uncorrectable using only the redundancy sent in the initial transmission.

Construction of HARQ requires a Rate Compatible Code and a choice of RV.

Rate Compatible Codes: Given a fixed number of information bits, consider a family of codes $\{C_1, C_2 ... C_n\}$ with rates $R_1 \geq R_2 \geq R_3 ... \geq R_n$, and block lengths $N_1 \leq N_2 \leq ... \leq N_n$. Then the set is rate compatible if codeword for C_i can be built by removing $N_j - N_i$ bits from codewords of code C_j , $j \geq i$, [13]. Rate Compatible Codes can be constructed by puncturing low rate codes.
Polar Codes for degraded channels are inherently rate compatible due to reliability ordering and nesting.

HARQ Schemes for Polar Codes

In [10], a HARQ scheme for Polar codes based on selective repetition has been studied. The Incremental Freezing scheme described in [11] and [13] can be classified as a HARQ scheme. In [8] another H-ARQ scheme based on Subset Polar Codes is described along with performance comparison with the other schemes. Here these schemes are discussed in brief.

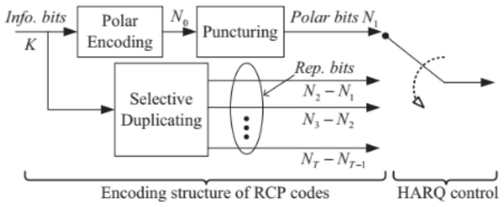


Figure 11: HARQ for Polar Codes with selective repetition

Selective Repetition: As the initial transmission, an information block of K bits is fed into a polar encoder. The output codeword of N_0 bits is punctured into N_1 bits and sent over the channel. If the receiver fails to decode the codeword, an NACK (negative acknowledgement) is sent to the transmitter through the feedback channel. And then, $N_2 - N_1$ of the information bits are retransmitted. This time, the receiver tries to perform decoding with all the N_2 received bits. If the decoding is failed again, another $N_3 - N_2$ bits are transmitted. This process continues until the transmitter receives an ACK (acknowledgement), or a maximum number of transmissions T is achieved. The scheme has been illustrated in figure 11. The retransmitted bits (RV) are chosen one at a time as the most unreliable of the K bits transmitted, reliability is calculated after choosing one bit and the process is iterated.

Incremental Freezing: This scheme is an improvement over the above scheme, it uses the reliability ordering to choose the RV as discussed in 2.4.2.

HARQ for Polar Codes based on Subset Polar Codes: A Subset Polar Code can be created by greedily puncturing a low-rate mother code without re-optimizing the information bits. The scheme uses equivalent Subset Polar Codes as RV. In [8], the author has claimed from simulation results that this scheme performs better among the ones discussed. Nevertheless, we have used Incremental Freezing for its simplicity.

Reliability Based HARQ: The schemes described above use CRC for checking decodability. Reliability based HARQ technique (RBHARQ) [15], eliminates the use of CRC by approximating bit and word error probability from likelihood ratios (LLR). As the magnitude of a log-likelihood value is directly connected to the error probability of the corresponding bit, it can be used to determine which bits most likely caused a word error. The bit error probability for the k^{th} bit can be estimated from LLR ($\tilde{u}_k$) as,

$$P_{b,k} = P(\hat{u_k} \neq u_k) = \frac{1}{1 + e^{|\tilde{u}_k|}} \tag{1}$$

then word error probability becomes,

$$P_w = 1 - e^{log\bar{P}_w} \tag{2}$$

where,

$$log\bar{P}_w = log\prod_{k=1}^{K}(1 - P_{b,k})$$

if the word error probability does not meet the requirements the bits with higher bit error probability may be retransmitted. This retransmission criterion results in increase of throughput, particularly evident in case of short packet lengths.

3 Proposed implementation of Recursive Data Exchange

As indicated in section 1, our approach towards implementation of a solution to the *Data-Exchange* problem using RDE and Polar Codes can be summarised as below,

- Use Rateless Polar Codes with Incremental Freezing to implement RDE.
- Use a PHY layer error detection as retransmission criterion in Incremental Freezing.

The following subsections elaborate the same.

3.1 Adaptation of Rateless Polar Codes for RDE

The RDE, which is essentially iterative SW compression, can be constructed using Rateless Polar Code as illustrated by the example in figure 12. Here, a compound BSC channel $\mathcal{C} = \{p_1 \leq p_2 \leq p_3 \leq p_4\}$ is considered, where p_i is the flipover probability of channel i. It is assumed that the rates supported by the channels are $\{R_1 =$

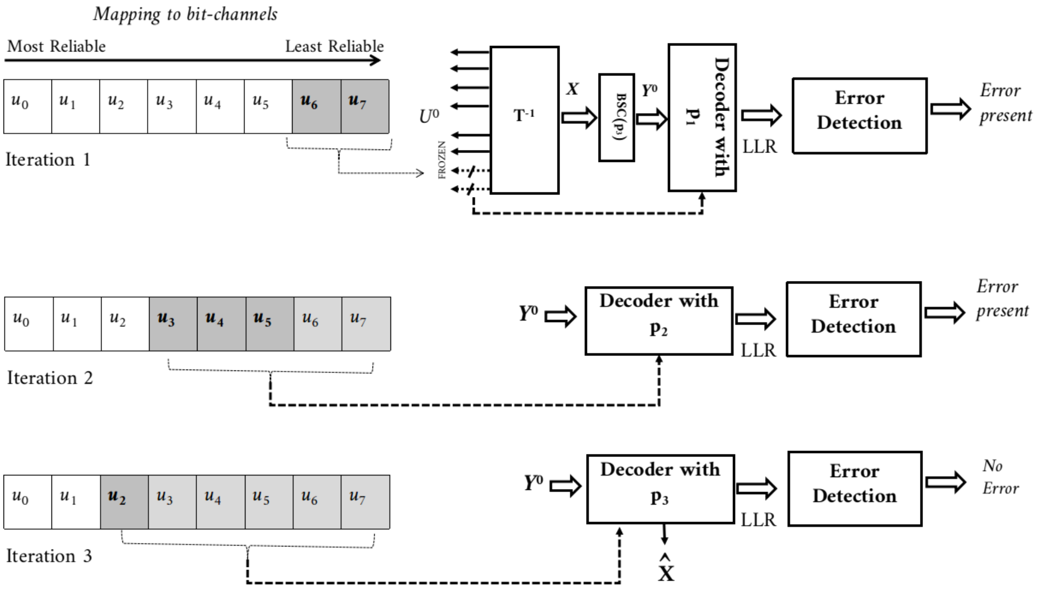


Figure 12: Example of Iterative SW compression with Incremental Freezing.

$R, R_2 = R/2, R_3 = R/3, R_4 = R/4\}$. The actual channel in the example is BSC(p_3), henceforth, we shall denote this as BSC($p_{channel}$). The vector channel is manufactured by polarization of N such BSC($p_{channel}$) channels.

First Iteration: In the first iteration, the SW compression with Polar Coding scheme suggested in figure 7 is followed. Additionally, the soft outputs (LLRs) generated from the decoder block are passed through an error-detection (ED) block. In case error is present, receiver replies with a NACK and further iterations are commenced. The decoder *guesses* the channel to be BSC(p_1) greedily and computes LLR accordingly. Henceforth, we shall denote this as BSC(p_{guess}). Note, in this case the frozen bits are non-zero, and are communicated error-free to receiver, unlike in the channel coding scenario in figure 10.

Subsequent Iterations: In the subsequent iterations, the decoder makes greedy guesses about the channel from a subset of the compound channel which does not include previous guesses. In other words, in second iteration $p_{guess} = p_2$, in third $p_{guess} = p_3$. Iterations continue as long as the ED does not declare error-free transmission, which is communicated as ACK by the receiver.
The error-free transmission in these iterations consist of information bits which are suspected to be sent unreliably in previous iterations under impression that present p_{guess} is correct.

Final Decoding: When the ED declares no error the decoder infers that the $p_{channel} = p_{guess}$, let us denote this as p_{final}. The decoder now decodes the received channel output vector using p_{final} and considering the bits transmitted error-free in all the iterations as frozen. Finally, with Arikan transform X is estimated. Note, X and Y have there usual meanings as in figure 2.

3.2 PHY Layer Error Detection

A difference between use of Polar Codes for error control and that for SW compression is that, in the latter case the U_N is generated from X_N. Hence using a CRC in U_N which can be exploited for error detection and retransmission in Incremental Freezing does not remain feasible.
This necessitates an error detection scheme which is free of CRC and works in PHY layer. At a high level the ED scheme can be viewed as a hypothesis testing problem

for *each iteration* where,

$$\begin{aligned}\mathcal{H}_0 &: \text{The channel is the current guess, i.e., } p_{channel} = p_{guess}\\ \mathcal{H}_1 &: \text{The channel is worse, i.e., } p_{channel} > p_{guess}\end{aligned}$$

It is understood that due to the sequential and greedy nature of Inc-Frz, cases where $p_{channel} < p_{guess}$ need not be considered. Since the magnitude of the LLR dictates the error probability of corresponding bit, a hypothesis test based on $|LLR|$ has been proposed.

3.2.1 The Error Detection Test

Consider the setting as in section 3.1. Let there be K good bit-channels after polarization. Since Inc-Frz guesses the best channel first, we can say that at j^{th} iteration $p_{guess} = p_j$. Let $\Lambda_j^i(k)$ be the magnitude of the LLR of k^{th} bit-channel, $k \in 1, 2, ...N$ at the output of the decoder at the end of j^{th} iteration such that $p_{guess} = p_j$ for $p_{channel} = p_i$. Empirical distribution of $\Lambda_j^i(k)$ based on a simulation has been shown in figure 13.
Thus, the hypotheses may be restated as,

$$\begin{aligned}\mathcal{H}_0 &: j = i\\ \mathcal{H}_1 &: j < i\end{aligned}$$

Note that, for $j < i$, $p_i > p_j$. Hence, the true channel is worse and further iterations are needed. Two tests have been suggested based on statistics which are functions of $\Lambda_j^i(k)$ in the following subsection.

3.2.2 Proposed Tests

TEST 1: All good channels are above a given threshold. Initially, a test was considered where $p_{guess} = p_{channel}$ is declared if the |LLR| of all the good channels clear a given threshold under the current guess. That is, after j^{th} iteration,

$$\begin{aligned}&j = i, \begin{cases}\text{if, } \Lambda_j^i(k) > \lambda, \forall k \in 1, 2, 3...K\\ \text{alternatively, } \min\limits_{k \in [K]} \Lambda_j^i(k) > \lambda\end{cases}\\ &j < i, \text{ o.w.}\end{aligned}$$

Here λ is chosen to reflect the required reliability. As it can be seen in figure 13, the support of the distributions of $\Lambda_{j=i}^i(k)$ and $\Lambda_{j\neq i}^i(k)$ overlap considerably and $\Lambda_{j=i}^i(k)$

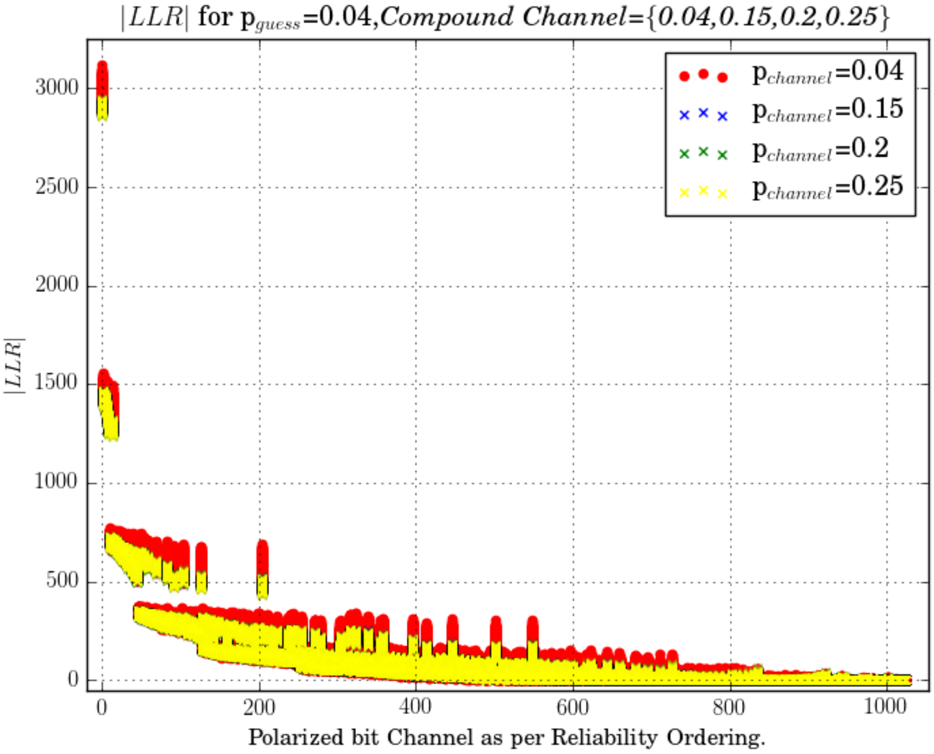


Figure 13: $|LLR|$ for $p_{guess} = 0.04$ and $p_{channel} \in \mathcal{C}$.

has a higher variance. This gave rise to high missed detection probability P_M, which resulted in premature termination of the Inc-Frz scheme and high probability of error. Here, $P_M = P_1(p_i > p_j)$ indicates the probability that the test declares a better channel as the true channel.

TEST 2: A given fraction of good channels are above a threshold. In this test $p_{guess} = p_{channel}$ is declared if the |LLR| of a certain fraction of the good channels clear a threshold. The fraction is dependent on the iteration number. After j^{th} iteration,

$$\begin{aligned} & j = i, \text{ if, } \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}_{\{\Lambda_j^i(k) > \lambda\}} > \Theta_j \\ & j < i, \text{ o.w.} \end{aligned}$$

The results of Monte-Carlo simulation for estimating P_M and P_F for $\mathcal{C} = \{p_1 = 0.04, p_2 = 0.15, p_3 = 0.2, p_4 = 0.25\}$ is shown in figure 14,15a, 15b. Note, in the figures Θ is considered to be percentage.

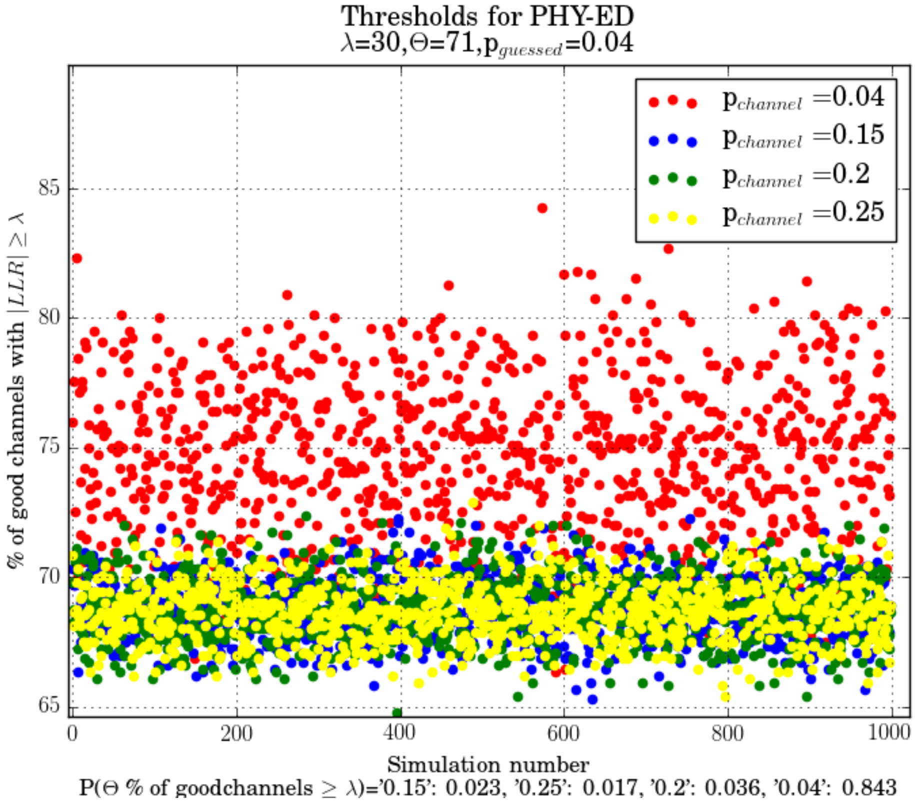
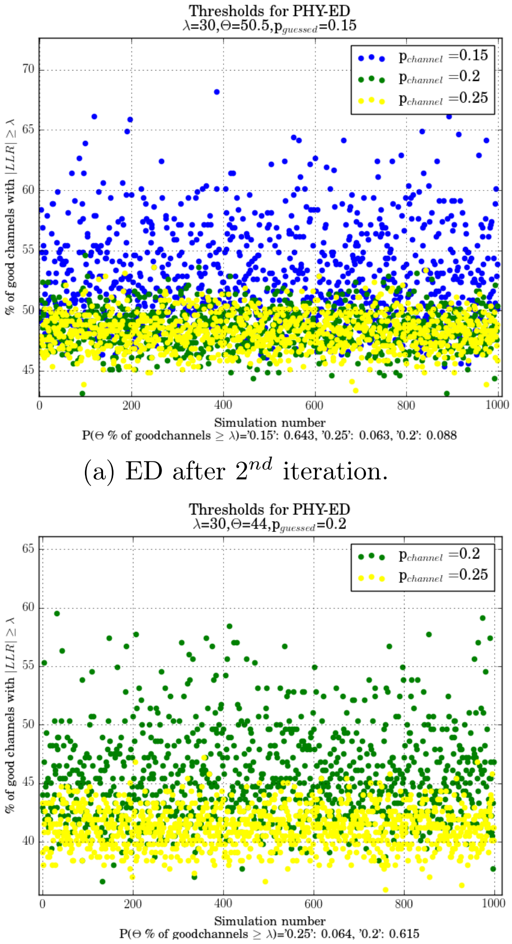


Figure 14: P_M and P_F estimation for ED after 1^{st} iteration, $p_{guess} = 0.04$.

From figure 14, the detection error probabilities for the first iteration can be calculated as $P_F = 0.16$ and $P_M = 0.06$. For the values of Θ used, $P_F \approx 0.4$ for second and third iteration with $P_M \approx 0.13$. A higher value of P_F does not affect the frame error probabilities but causes rate loss. Though higher value of P_M affects frame error probabilities adversely, it should be noted that the detection error probabilities in later iterations are weighted by that of the previous ones and hence this effect diminishes with iteration.



(a) ED after 2^{nd} iteration.

(b) ED after 3^{rd} iteration.

Figure 15: P_M and P_F estimation for ED after 2^{nd} and 3^{rd} iteration.

4 Performance Evaluation

Simulation results for the scheme in figure 12 used for iterative SW compression with Inc-Frz and PHY-ED are presented in figure 16, 17, 18 and 19. Each plot corresponds to a fixed channel with $p_{channel} \in \mathcal{C}$. The performance of Polar Codes designed for the specific $p_{channel}$ has been given for comparison. Figure 21 shows similar results for Rateless Polar Coding with Inc-Frz and PHY-ED.

Let K be the number of bit-channels assumed to be good at the first iteration. For simulation, K is varied from 0 to K^* such that K^*/N is the capacity of the best channel.

In case of Rateless Polar Coding, the scheme uses iterations to communicate these K bits, thus achieving some rate and corresponding frame error rate (FER). In case of SW compression the number of bits that remain unfrozen after the final iteration divided by N is viewed as the achieved rate. The FER has been plotted against the achieved rates for performance evaluation. The plots capture the error detection and retransmission action which results in change in effective block length of the code. Figure 20 captures the rate loss seen for varying $p_{channel}$.

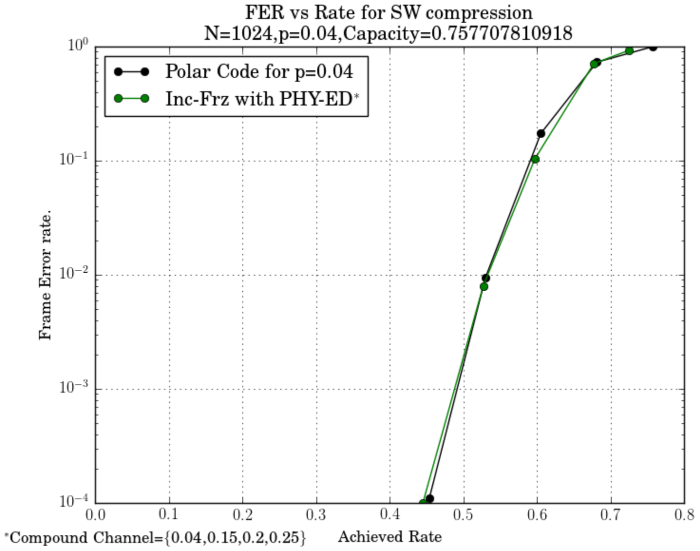


Figure 16: FER vs Rate for SW compression with $p_{channel} = 0.04$

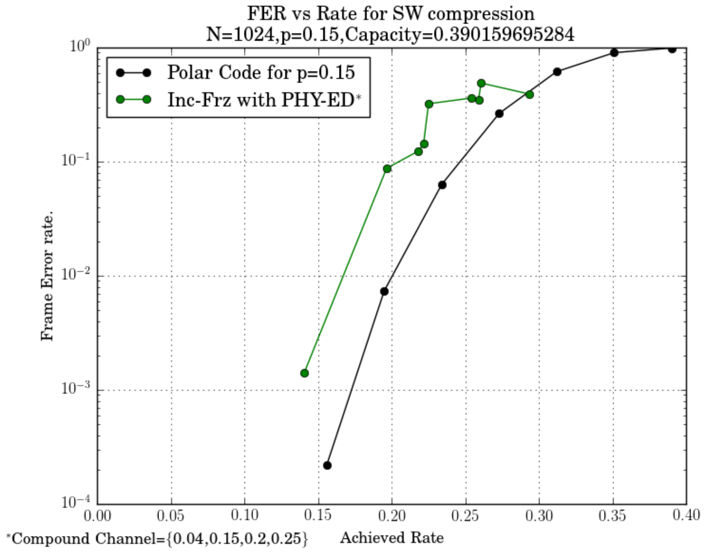


Figure 17: FER vs Rate for SW compression with $p_{channel} = 0.15$

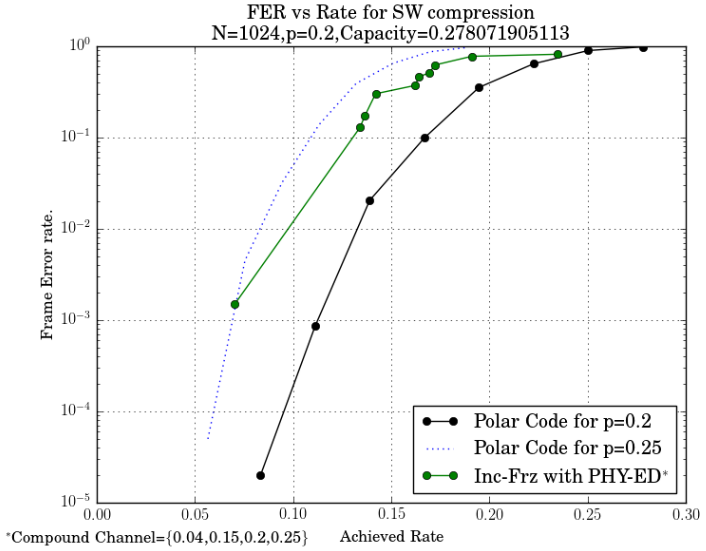


Figure 18: FER vs Rate for SW compression with $p_{channel} = 0.2$

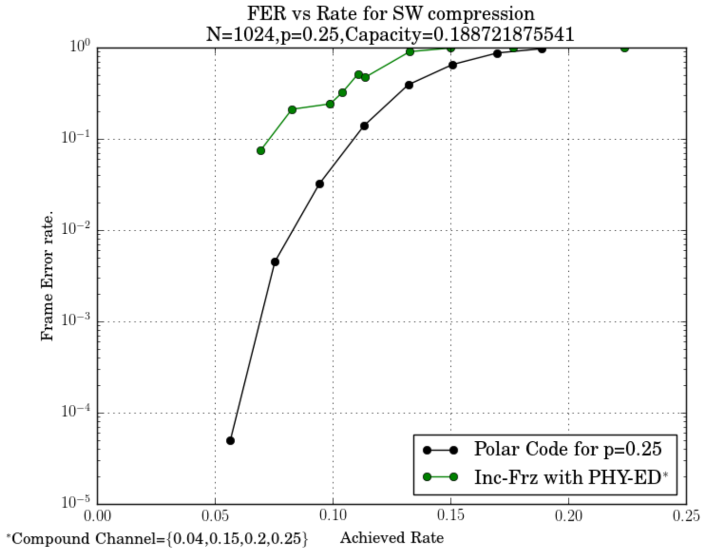


Figure 19: FER vs Rate for SW compression with $p_{channel} = 0.25$

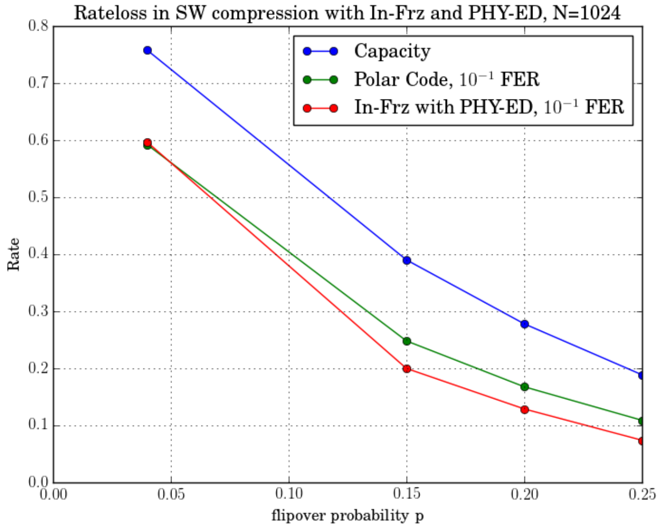
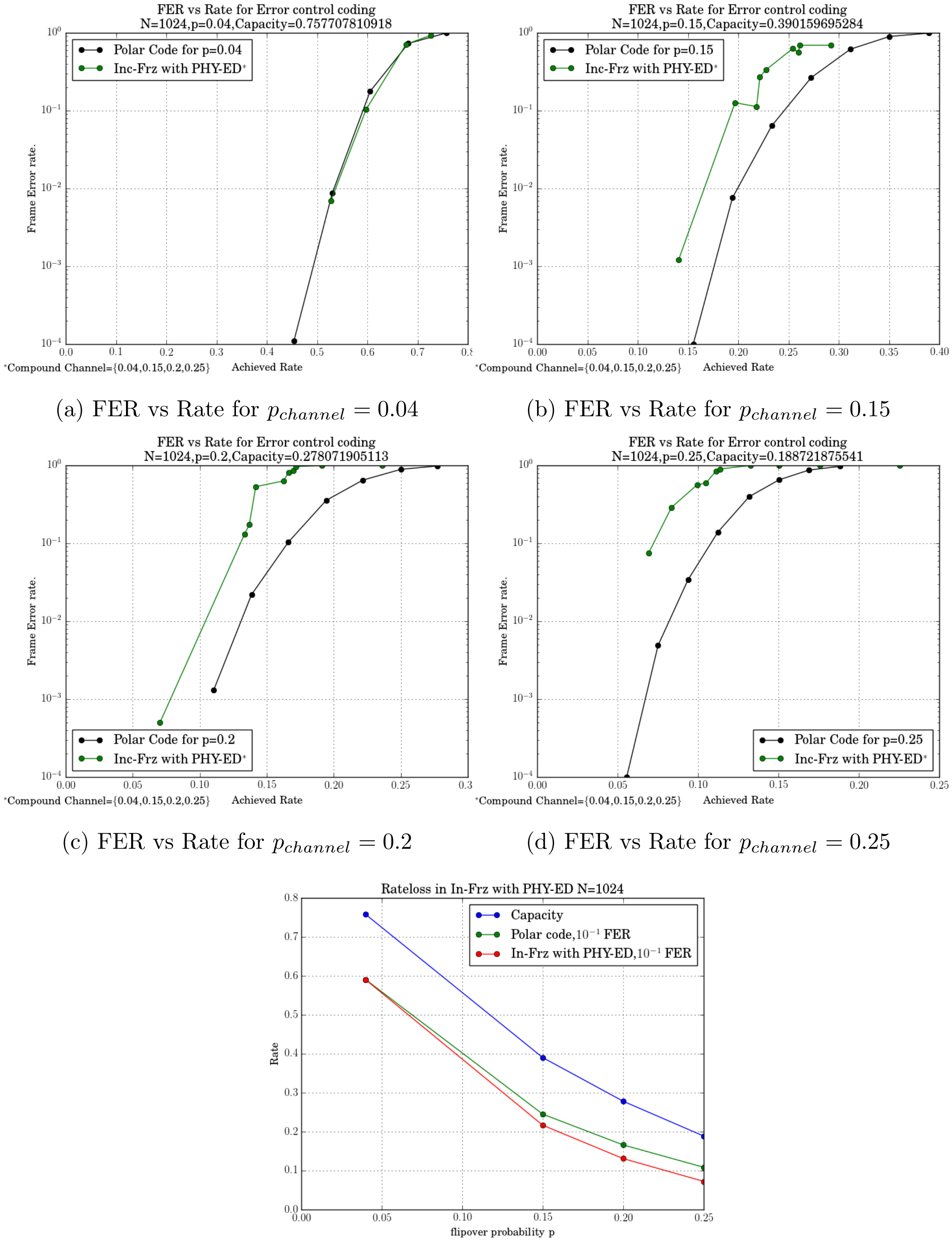


Figure 20: Rate loss for SW compression with Inc-Frz.



(a) FER vs Rate for $p_{channel} = 0.04$

(b) FER vs Rate for $p_{channel} = 0.15$

(c) FER vs Rate for $p_{channel} = 0.2$

(d) FER vs Rate for $p_{channel} = 0.25$

(e) Rate loss in Inc-Frz with PHY-ED

Figure 21: Performance Evaluation of Rateless Polar Coding with Inc-Frz and PHY-ED.

5 Conclusion and Future work

The proposed scheme is an implementable solution to the *Data-Exchange* problem. It reduces the communication among nodes. The CRC-free universal polar code promises considerable rate gain for communication using short packet lengths. There are few channels which are good for use during the entire transmission. Communicating critical data over these channel ensure high reliability and availability .

- Future work.
    - Extensive performance analysis and theoretical analysis of proposed error detection scheme as a RB-HARQ for Polar Codes.
    - Implementation of the scheme for multiparty data exchange.

References

[1] A. Tridgell and P. Mackerras. The r-sync algorithm. *Joint Computer Science and Technical Report Series*, TR-CS-96-05, 1996.

[2] D. Slepian and J. K. Wolf. Noiseless Coding of Correlated Information Sources. *IEEE Transactions on Information Theory*, 1973.

[3] S. Sandeep Pradhan and K. Ramachanderan. Distributed Source Coding Using Syndromes (DISCUS):Design and Construction. *IEEE Transactions on Information Theory*, 49,No. 3, 2003.

[4] S. Onay. Polar Codes for Nonassymetric Slepian-Wolf Coding. *arXiv.1208.3056v1[cs.IT]*, August 2012.

[5] H. Tyagi and S. Watanabe. Universal Multiparty Data Exchange and Secret Key Arrangement. *ISIT*, 2016.

[6] E. Arıkan. Channel polarization: A method for constructing capacity- achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55, no. 7:3051–3073, July 2009.

[7] Y. Zhang et al. A Practical Construction Method for Polar Codes. *IEEE Communication letters*, 18, no. 11, Nov. 2014.

[8] S. Tavildar. A H-ARQ Scheme for Polar codes. *arXiv.1606.08545v1[cs.IT]*, Jun. 2016.

[9] J. Cheng. Coding Performance of Hybrid ARQ schemes. *IEEE Transaction on Communications*, VOL. 54, NO. 6, Jun. 2006.

[10] J. Lin K. Chen, K. Niu. A Hybrid ARQ Scheme based on Polar codes. *IEEE Transaction on Communications*, VOL. 17, NO. 10, Oct. 2013.

[11] K. Chen B. Li, D. Tse and H. Shen. Capacity Achieving Rateless Polar Codes. *arXiv.1508.0311v1[cs.IT]*, Aug. 2015.

[12] E. Sasoglu and L. Wang. Universal Polarization. *arXiv:1307.7495v2[cs.IT]*, Dec. 2013.

[13] M. Mondelli et al. Capacity achieving rate compatible polar codes for general channels. *arXiv.1611.01199v2[cs.IT]*, Jan. 2017.

[14] M. Al-Mualla H. Mukhtar, A. Al-Dweik. CRC-Free Hybrid ARQ using Turbo Product codes. *IEEE Transaction on Communications*, VOL. 62, NO. 12, Dec. 2014.

[15] H. Schoeneich J. Fricke and P. A. Hoeher. Reliability based retransmission criterion for HARQ. *IEEE Transaction on Communications*, VOL. 57, NO. 8, Aug. 2009.