COMMUNICATION EFFICIENT DATA EXCHANGE AMONG MULTIPLE NODES

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

MID-TERM PROJECT REPORT

A report by

SOUMYA SUBHRA BANERJEE
SR No. 04-02-04-37-42-16-1-14191
DEPARTMENT OF ECE,
INDIAN INSTITUTE OF SCIENCE.

Under guidance of

HIMANSHU TYAGI
ASSISSTANT PROFFESSOR
DEPARTMENT OF ECE,
INDIAN INSTITUTE OF SCIENCE.

January 15, 2018

CONTENTS

LIST OF FIGURES

[1]

[1] This Project was supported by Robert Bosch Center for Cyber Physical Systems

ABSTRACT

1 INTRODUCTION

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

A working solution for this problem is r-sync protocol, as described in [2]. 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 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 uses only one guess, and thus ends up using more communication.

In [3], David Slepian and Jack Wolf had shown that the optimal solution to this problem is Slepian-Wolf compression.

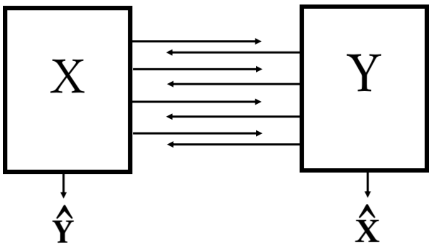

Figure 1: The Data-Exchange Problem

SLEPIAN-WOLF CODING THEOREM: states under joint decoding of X and Y a total rate H(X,Y) is sufficient.

As described in [4], 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 theoritical 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.

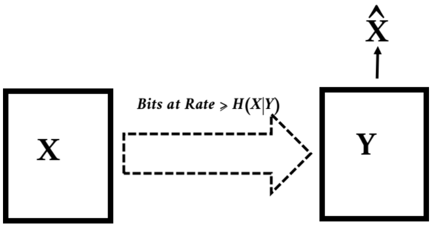

Figure 2: The Slepian-Wolf Compression

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.

Using structured channel codes as indicated in [4], particularly Polar Codes as shown in [5], alongwith *recursive data exchange protocol* mentioned in [6] for Slepian-Wolf compression eases the aforementioned implementation.

1.1 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 Slepian-Wolf Compression using Polar Codes.
- Achieve universality using *Recursive Data Exchange* protocol (RDE).
- Realise RDE using Rateless Polar Codes with Physical layer Error Detection.

The following sub-sections discuss the artefacts needed for this implementation in brief. Section 2, consolidates and elaborates the proposed scheme.

1.2 Interactive Communication for Data Exchange

The data exchange protocol is based on an interactive version of the Slepian-Wolf 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 recieved or some pre-decided number of bits (l_{max}) have been transmitted [6]. 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 [6] are theoritical constructs which use type classes to form a list of guesses for data of other parties and thus has exponential complexity.
In [5], Slepian Wolf compression is approached with structered (Polar) codes, in this work we use Rateless Polar Codes to implement RDE with a similar ideology.

1.3 Brief Introduction to Polar Codes

In 2008, E. Arikan in his paper [1] 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.

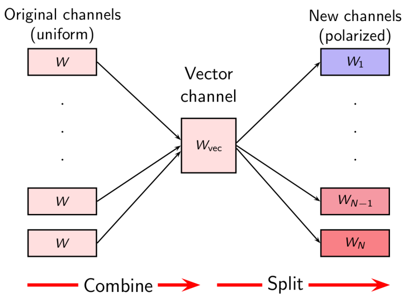

Figure 3: Channel combining and splitting

CHANNEL POLARIZATION , is an operation by which one manufactures out of N independent copies of a given B-DMC W, a second set of N channels $\{W_N^{(i)} : 1 \leqslant i \leqslant N\}$ 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, Bhattacharya parameter $Z(W_N^{(i)})$ tend to 1 or 0 respectively. The channels with $Z(W_N^{(i)}) = 0$ captures the capacity of W_{vec}.

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, this 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 *bad or frozen*, sending no useful data on them. This scheme of error control coding with polar codes is illustrated in figure 5.

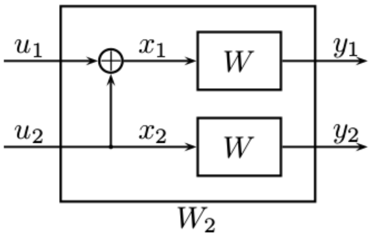

Figure 4: Arikan transformation butterfly

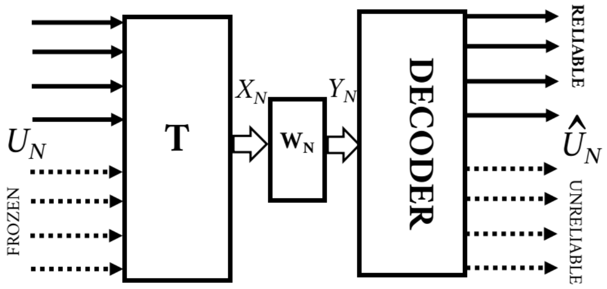

Figure 5: Polar Coding

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 $T = T^{-1}$.

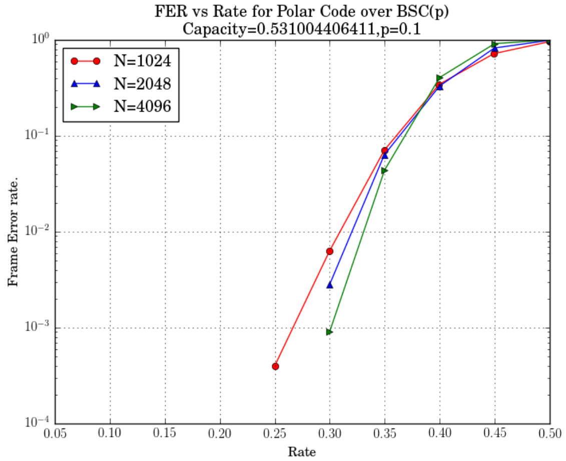

Figure 6: FER vs Rate for Polar Coding with SC

For our purpose, we shall be using Succesive 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

1.4 Implementation of SW Compression using Polar Codes

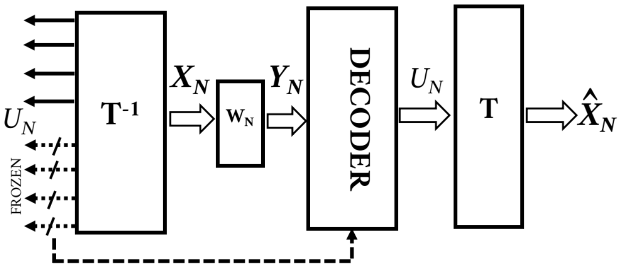

Figure 7: Polar Coding for SW Compression

In [4], the use of structured codes for Slepian-Wolf compression has been discussed.Further, in [5] a specific scheme using Polar Codes has been illustrated, as shown in figure 7. Consider the setting where X_N and Y_N are uniform. Y_N is a corrupted version of X_N by N BSC(p) channel uses. 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 Slepian-Wolf compression they are non-zero by the nature of the scheme. Performance analysis of this scheme is presented in figure 8.

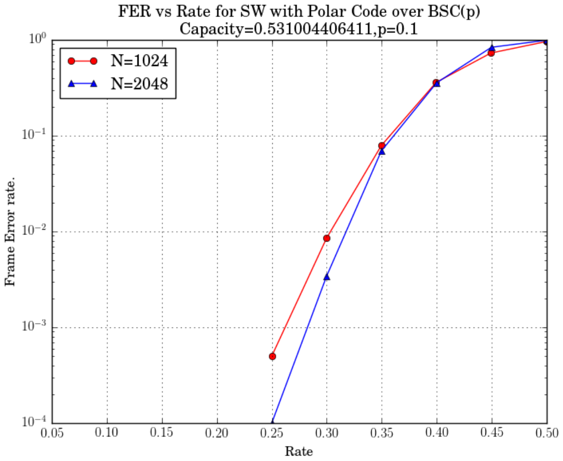

Figure 8: FER vs Rate for SW with Polar Code

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.

1.5 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 techniques and uses code puncturing.
For Polar Codes, puncturing is not straightforward. Hybrid-ARQ 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, this makes puncturing a relatively simple affair.

1.5.1 *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, $\mathrm{BSC}(p_1) \preceq \mathrm{BSC}(p_2)$ if $p_1 \geqslant p_2$.

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

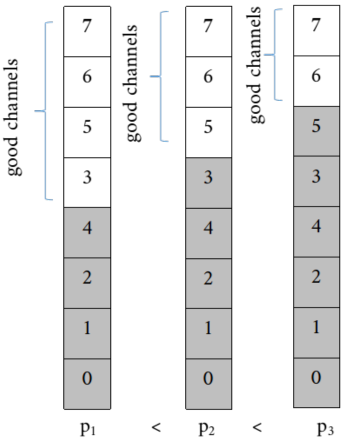

Figure 9: Nesting in BSC(p) channels

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

1.5.2 *Incremental freezing*

Given the reliability ordering, a 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*.

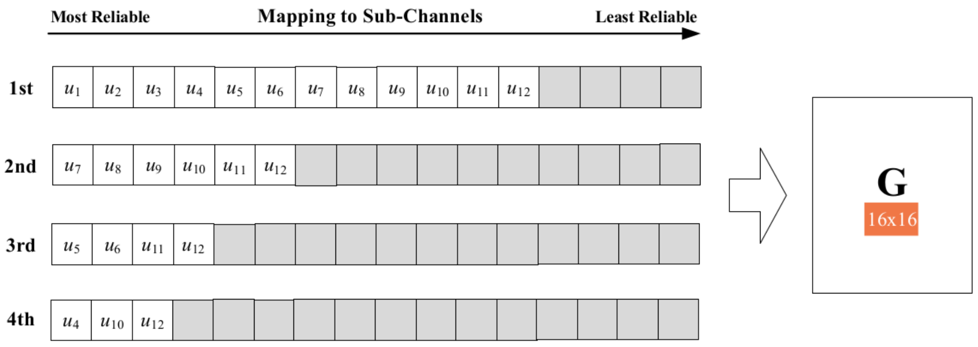
Most Reliable
Mapping to Sub-Channels
Least Reliable
1st
u_1 u_2 u_3 u_4 u_5 u_6 u_7 u_8 u_9 u_{10} u_{11} u_{12}
2nd
u_7 u_8 u_9 u_{10} u_{11} u_{12}
3rd
u_5 u_6 u_{11} u_{12}
4th
u_4 u_{10} u_{12}
G
16x16

Figure 10: Incremental freezing for N=16, K=12 and 4 iterations

Figure 10 illustrates the scheme for a set of channels with rates $\{R = 12/16, R_1 = R/2, R_2 = R/3, R_3 = R/4\}$. Here u_i are the message bits. Note that with the 4^th transmission u_4 to u_{12} has been incrementally frozen. This scheme is capacity achieving in the sense that no rate has been wasted, the final rate achieved is,

$$R^* = \frac{12}{16x4} = \frac{3}{16} = R_3$$

. In case the real channel has capacity $\geqslant R_3$ 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. A specific application of this will be discussed in section 3.

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 2.

The following subsections briefly discuss Rateless Polar Coding from the perspective of Hybrid-ARQ.

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

1.5.3 *H-ARQ 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 called 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 H-ARQ requires a Rate Compatible Code and a choice of RV (retransmission scheme).

RATE COMPATIBLE CODES: Given a fixed number of information bits, consider a family of codes $\{C_1, C_2 ... C_n\}$ with rates $R_1 \geqslant R_2 \geqslant R_3 ... \geqslant R_n$, and block lengths $N_1 \leqslant N_2 \leqslant ... \leqslant N_n$. 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 ,$j \geqslant i$, [13]. Rate Compatible Codes can be constructed by puncturing low rate codes.
Polar codes for degraded channels is inherently rate compatible due to *reliability ordering* and *nesting*.

H-ARQ schemes for Polar Codes

In [10], a H-ARQ 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 H-ARQ 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 briefly discussed.

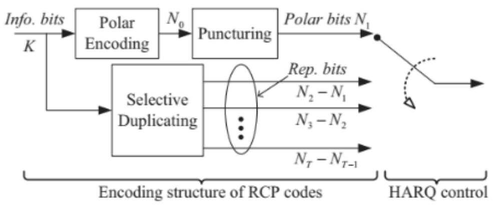

Figure 11: H-ARQ for Polar Codes with selective repetition

H-ARQ FOR POLAR CODES WITH 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 re-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 1.5.2.

H-ARQ 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 H-ARQ: It is to be noted that 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 form 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.

2 PROPOSED IMPLEMENTATION OF RECURSIVE DATA EXCHANGE

As indicated in section 1 and 1.1, our approach towards implementation of a solution to *The Data Exchange* using RDE and Polar codes can be summarised as below,

- Use Rateless Polar Codes with Incremental Freezing to implement iterative Slepian-Wolf Compression.
- Use a PHY layer error detection as retransmission criterion in Incremantal freezing.

The following subsections elaborate on the same.

2.1 Adaptation of Rateless Polar Codes for RDE

The RDE, which is essentially an iterative Slepian-Wolf compression, can be constructed using Rateless Polar Code as shown in the example of figure 12. Here, a compund BSC channel $\mathcal{C} = \{p_1 \leqslant p_2 \leqslant p_3 \leqslant p_4\}$, is considered, where p_i is the flipover probability of channel i. It is assumed that the rates supported by the channels are $\{R_1 = R, R_2 = R/2, R_3 = R/3, R_4 = R/4\}$. The actual channel in the example is BSC(p_3), henceforth we shall denote this as BSC($p_{channel}$).

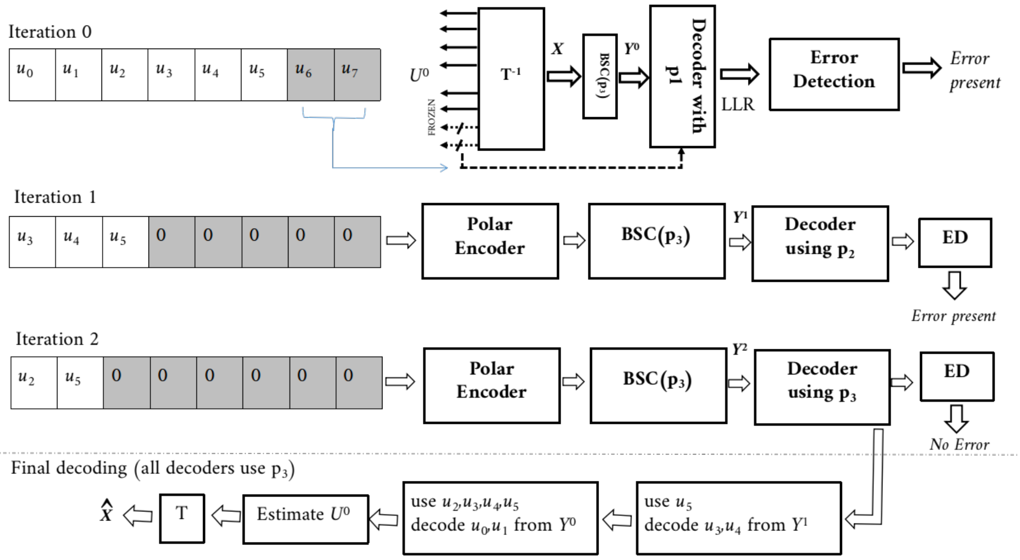

Figure 12: Example of Iterative Slepian-Wolf Compression with Incremental Freezing.

FIRST ITERATION: In the first iteration, the Slepian-Wolf 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. It is to be noted that 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 iteration the frozen bits are non-zero, and are communicated error-free to receiver.

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 message sent 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, as discussed in section 1.5.2 . The frozen bits are set to zero and need not sent to receiver. Transmission is as illustrated in figure 5.

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 bits of penultimate iteration from the channel output of the penultimate iteration using p_{final} and considering the information bits decoded in last iteration as frozen. This continues till all bits are decoded successfully. Finally, with arikan transform X is estimated.
Note, X and Y have there usual meanings as in figure 2.

2.2 PHY-Layer error Detection

A distinct difference between Polar codes for error control and Slepian-Wolf compression using polar codes is that, in the latter 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

2.2.1 *The Error Detection Test*

2.3 Proposed Tests

2.4 Performance Evaluation

3 CONCLUSION AND FUTURE WORK

proposed scheme shortpacket implementation

Definition 1. We can redefine Ω_i as a set of indicator vectors of A.Let,

$$[\phi_i(A)]_l = \begin{cases} \mathbf{1}_A(l) \text{ if } l < i \\ \mathbf{1}_A(l+1) \text{ if } l \geqslant i. \end{cases}$$

$$\begin{aligned} \Omega_i' &\triangleq \{\phi_i(A) \in \{0,1\}^{N-1} : A \in \Omega_i\} \\ \partial_j \Omega_i' &\triangleq \{\phi_i(A) \in \{0,1\}^{N-1} : A \in \partial_j \Omega_i\}. \\ &= \{\underline{x} \in \{0,1\}^{N-1} | \mathbb{1}_{\Omega_i}(\underline{x}) \neq \mathbb{1}_{\Omega_i}(\underline{x}^{(j)})\} \end{aligned}$$

Here the last equality follows from definition of $\partial_j \Omega_i$.
Consider the space $\{0,1\}^M$,we can redefine measure μ_p such that

$$\mu_p(\Omega) = \sum_{\underline{x} \in \Omega} p^{|\underline{x}|}(1-p)^{M-|\underline{x}|}, \text{ for } \Omega \subseteq \{0,1\}^M,$$

where the weight $|\underline{x}| = x_1 + x_2 + \ldots + x_M$ is the number of 1's in $\underline{x}$.

Definition 2. For a monotone set Ω.The influence of bit $j \in [N]$,is defined by,

$$I_j^{(p)}(\Omega) \triangleq \mu_p(\partial_j \Omega)$$

The total influence in defined by,

$$I^{(p)} \triangleq \sum_{j=1}^{N} I_j^{(p)}.$$

Using proposition **??**(a) and proposition 7, we have,

$$h(p) = h_i(p) = \mu_p(\Omega_i')$$

Further,from proposition **??**(b), we get,

$$I_j^p(\Omega_i') = \mu_p(\partial_j \Omega_i') = \left.\frac{\partial h_i(\underline{p})}{\partial p_j'}\right|_{\underline{p}=(p,p,\ldots,p)}$$

where j' is given by

$$j' = \begin{cases} j \text{ if } j < i \\ j+1 \text{ if } j \geqslant i. \end{cases}$$

Since $\mathcal{G}$ is doubly transitive, from propostion **??**,

$$I_j^p(\Omega_i') = I_k^p(\Omega_i') \text{ for all } j,k \in [N-1].$$

Hence,Ω_i is a *symmetric monotone set.*

The following theorem could be seen as a consequence of the result by Talagrand [**?**].

Theorem 1. *Let Ω be a monotone set and suppose that, for all $0 \leqslant p \leqslant 1$, the influences of all bits are equal $I_1^{(p)}(\Omega) = \ldots = I_M^{(p)}(\Omega)$. Then, for any $0 < \epsilon \leqslant 1/2$,*

$$p_{1-\epsilon} - p_\epsilon \leqslant \frac{2 \log \frac{1-\epsilon}{\epsilon}}{C \log(N-1)},$$

where $p_t = \inf\{p \in [0,1] : \mu_p(\Omega) \geqslant t\}$ is well defined because $\mu_p(\Omega)$ is strictly increasing in p with $\mu_0(\Omega) = 0$ and $\mu_1(\Omega) = 1$.

Proof. Using Russo's lemma [**?**]. □

We see that Ω_i satisfies the conditions of Theorem 1. Hence,

$$\lim_{n \to \infty} (p_{1-\epsilon} - p_\epsilon) = 0.$$

Further , using proposition **??**, we state,
$\{\mathcal{C}_n\}$ *is capacity achieving on the BEC under bit-MAP decoding.*

REFERENCES

[1] 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.

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

[3] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 1973.

[4] S. Sandeep Pradhan and K. Ramachanderan. Distributed source coding using syndromes (discus):design and construction. *IEEE Transactions on Information Theory*, 49,No. 3, 2003.

[5] S. Onay. Polar codes for nonassymetric slepian-wolf coding. *arXiv.1208.3056v1[cs.IT]*, August 2012.

[6] H. Tyagi and S. Watanabe. Universal multiparty data exchange and secret key arrangement. *ISIT*, 2016.

[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 TRANSACTIONS ON COMMUNICATIONS*, VOL. 54, NO. 6, Jun. 2006.

[10] J. Lin K. Chen, K. Niu. A hybrid arq scheme based on polar codes. *IEEE TRANSACTIONS 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 TRANSACTIONS ON COMMUNICATIONS*, VOL. 62, NO. 12, Dec. 2014.

[15] H. Schoeneich J. Fricke and P. A. Hoeher. Reliability based retransmission criterion for harq. *IEEE TRANSACTIONS ON COMMUNICATIONS*, VOL. 57, NO. 8, Aug. 2009.