Practical Universal Data Exchange using Polar Codes

Soumya Subhra Banerjee* Himanshu Tyagi*

Abstract—**In the multiparty data exchange problem, parties with correlated data seek to recover each-other's data. We study practical, universal schemes for this problem that accomplish data exchange using optimal rate communication for any distribution of observations. Our focus in this work is on binary symmetric distributions where each user observes bit sequences with uniform marginals. We consider binary symmetric Markov trees as a natural multiparty extension of the binary symmetric source and seek universally rate optimal algorithms for this family. Our main theoretical result is a completeness theorem which shows that any universal Slepian-Wolf scheme can be converted efficiently to a universal data exchange scheme for a subfamily of binary symmetric Markov trees. We instantiate this result using Polar codes. In particular, we provide a universal Slepian-Wolf code using Polar codes and use our reduction algorithm to convert it to a multiparty data exchange protocol. The resulting scheme provides the first practical construction of codes for universal data exchange, which we evaluate numerically.**

I. Introduction

Multiparty data exchange is a multiterminal source coding problem where parties observing correlated data seek to recover each-other's data. A solution for this problem will have many applications, including extending two-party functionalities such as `rsync` to multiple parties. A Shannon theoretic variant of this problem was introduced in [5], where it was termed the "omniscience" problem and the minimum rate of communication for omniscience was characterized. It was also shown in [5] that a data exchange protocol can be used to generate a multiparty secret key of maximum possible rate, which constitutes another application of such a primitive. Motivated by these applications, we seek practical algorithms for multiparty data exchange.

An important step towards this goal was taken in [11] (see, also, [10]) where a *universal data exchange* (UDE) protocol called Recursive Data Exchange (RDE) was given, namely an interactive protocol that achieves data exchange using minimum rate of communication without knowing the underlying generating distribution. While theoretically pleasing, decoding in RDE involves search over a list of exponential size (in file length). Therefore, the solution provided in [11] falls short of practical feasibility.

In this work, we take the first step towards a practical data exchange protocol. We restrict to the class of binary sources where marginals of each sequence is uniform Bernoulli, and consider a Markov chain over tree model as its multiparty extension. An important structural property of RDE is that it recursively applies a two-party primitive to obtain a multiparty solution. Such results where a subclass of problems capture the most general version of the problem are called *completeness results*. However, even this two-party primitive used in RDE is not practically feasible. Our proposed solution has a similar structural form. We show that a two-party, universal Slepian-Wolf (USW) code can be extended to a UDE protocol for an interesting subclass of Markov chain over trees – we call this subclass *progressively degraded Markov trees* (DMT). This *completeness result* is the main theoretical contribution of our work.

As a consequence of this result, we design a UDE protocol for DMTs using Polar codes. We first present a new USW code using Polar codes and use the general reduction mentioned above to convert it to a UDE. The proposed USW is based on our recent Polar code based scheme for implementing universal HARQs [3]. We have implemented our scheme in Python and have numerically validated the practical feasibility of our scheme.

We remark at the outset that our completeness result is grounded in theory, and allows us to limit the design problem to that of USW design. However, the Polar code construction for USW we present is based on heuristics and is only validated through numerical experiments. An approximate error analysis under a decoupling assumption for polarized channels can be provided along the lines of [3], but the exact analysis evades us. Finally, even the theoretical benchmark used to compare our numerical performance is not satisfactory, since it ignores the finite blocklength effects and the role of limited rounds of interaction.

We set the stage in the next section with preliminaries. Section III contains our general procedure for extending a USW to a UDE for DMTs, along with the proof of completeness theorem. Our scheme based on Polar codes is described in Section IV, along with numerical evaluations in the final subsection.

II. Preliminaries

In this section, we review briefly some basic notions that will be used throughout this paper.

A. Interactive protocols and oracle access

We present only a "semi-formal" definition of multiparty *interactive communication protocol* in the interest of space. (A

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

formal definition can be given using tree-protocols; see [7].) A multiparty interactive communication protocol proceeds in rounds, with one party communicating in each round. The communication sent in each round depends on the previously received communication and the local observations of the party. We allow randomized protocols where the input of the parties includes their local input and a random variable U shared with all the parties. The communication sent in each round is either a binary string of arbitrary length, namely an element in $\{0,1\}^*$, or a special symbol indicating termination, upon receiving which the protocol stops. Also, communication in each round determines which party will communicate next. We use the notation π for a protocol and Π for its terminated *transcript*, namely all the bits sent in the protocol.

An *output of a protocol* is a random variable that can be computed by parties using its transcript Π and their respective observations. The *length of the protocol* is the (random) length $|\Pi|$ of its transcript Π, namely the number of bits in Π (excluding the termination symbol). Finally, the *rate of a protocol* working with inputs of length n bits (for each party) is given by $|\Pi|/n$.

We conclude this part with a brief description of *oracle access to a protocol*. In computational complexity theory, oracle access is a standard notion where an algorithm A has oracle access to an algorithm B if A in its execution is allowed to send inputs to B and use the resulting outputs. In this work, we allow a slight variant of this basic oracle access, which we call *oracle access with pause*, where A in its oracle access to B is allowed to "pause" the execution of B midway and "resume" later as needed.

B. The SW problem and multiparty data exchange

The two-party data exchange problem is closely related to the classic Slepian-Wolf (SW) problem [9]. In this problem, party 1 observes X^n and party 2 observes Y^n, where $(X_t, Y_t)_{t=1}^n$ is an iid sequence generated from P_{XY}. Party 1 uses a mapping $e : \mathcal{X}^n \to \{0,1\}^\ell$ to compute the message $c = e(X^n)$, which it sends to party 2. Party 2 uses a mapping $d : \{0,1\}^\ell \times \mathcal{Y}^n \to \mathcal{X}^n$ forms an estimate $\hat{X} = d(c, Y^n)$ of X^n. The rate of the SW code (e,d) is $R = \ell/n$ and its probability of error is given by $\mathbb{P}\left(\hat{X} \neq X^n\right)$. The main result of [9] is that the minimum rate of a SW code for achieving a vanishing probability of error is $R^* = H(X|Y)$.

The two-party data exchange problem extends the SW problem and requires both parties to recover each-other's observations. Further, instead of the one-way communication protocol above, we allow interactive communication protocols. The multiparty data exchange problem has m parties observing n-length iid sequences, with party i observing $X(i) = X_{i1}, ..., X_{in}$ and $(X_{1t}, ..., X_{mt})_{t=1}^n$ generated iid from $\mathrm{P}_{X_1 \ldots X_m}$. Parties execute an interactive communication protocol and use its transcript Π and their local observations $X(i)$ to form estimates of observations of all the other parties observations. Such a protocol constitutes an ε-DE protocol if the estimates of all the parties are correct with probability of error less ε. This problem was formulated first in [5] where the minimum asymptotic rate of communication required for enabling ε_n-DE with asymptotically vanishing ε_n was characterized. This optimal rate, termed the *minimum communication of omniscience* in [5] is given by[1]

$$R^*(\mathrm{P}_{X_1 \ldots X_m}) = \max_{\sigma_1, \ldots, \sigma_k} \frac{1}{k-1} \sum_{i=1}^{k} H(X_{[m]}|X_{\sigma_i}),$$

where the maximum is over all partitions $\sigma_1, ..., \sigma_k$ of $[m]$ and all k. An important consequence of this form, that we exploit in our analysis, is that if a protocol completes data exchange using communication $(1/(k-1)) \sum_{i=1}^k H(X_{[m]}|X_{\sigma_i})$ for any partition $\sigma_1, ..., \sigma_k$, then it must be of optimal rate.

For convenience, we distinguish the communication protocol, which we denote by the encoder mapping e, from the function used to compute the output, which we denote by decoder mapping d. Furthermore, we include the termination decision as a part of d. For brevity, we will abbreviate a data exchange protocol π as (e,d).

C. Universal data exchange

Given a family of distributions $\mathcal{P}$ for iid data of the parties, a universal data exchange (UDE) for $\mathcal{P}$ protocol is an interactive protocol that accomplishes data exchange for any source distribution $\mathrm{P}_{X_1 \ldots X_m} \in \mathcal{P}$ using communication of rate $R^*(P_{X_1 \ldots X_m})$ (or arbitrarily close to it asymptotically). Such a protocol was given in [11] for the family of all distributions. In fact, the protocol proposed in [11] satisfied universality in a stronger, individual sequence sense. Formally, we work with the following definition of UDE protocols.

Definition 1. For $\varepsilon \in (0,1)$ and $\eta > 0$, an interactive protocol $\pi = (e,d)$ is (ε,η)-UDE for n-length $\mathcal{P}$ if when the observations $(X_{1t}, ..., X_{mt})_{t=1}^n$ are generated iid from $P \in \mathcal{P}$, the estimates $\hat{X}(j,i)$ of $X(i)$ formed by party i, $j \neq i$, and the (random) transcript Π of the protocol satisfy for all $j \in [m] \setminus \{i\}, i \in [m]$ that $\mathbb{P}\left(|\Pi| > n(R^*(P) + \eta)\right) \leq \varepsilon$ and $\mathbb{P}\left(X(j) \neq \hat{X}(j,i)\right) \leq \varepsilon$, where $X(j) = (X_{j1}, ..., X_{jn})$ is the observation of party j.

A related notion which we will need is that of a universal SW (USW) protocol. We enforce a little more structure on this notion to enable our completeness theorem. We assume the following: (i) The protocol proceeds interactively in rounds and in each round $n\eta$ bits are sent; (ii) when the protocol terminates, the decoder mapping d is applied to Y^n and transcript Π to obtain the estimate $\hat{X} = d(\Pi, Y^n)$.

Definition 2. For $\varepsilon \in (0,1)$ and $\eta > 0$, a two-party interactive protocol $\pi = (e,d)$ is (ε,η)-USW for n-length $\mathcal{P}$ if when the observations of parties (X^n, Y^n) are generated iid from $P \in \mathcal{P}$, the estimates $\hat{X}$ of X^n formed by party 2, and the (random) transcript Π of the protocol satisfy $\mathbb{P}\left(|\Pi| > n(H_P(X|Y) + \eta)\right) \leq \varepsilon$ and $\mathbb{P}\left(X^n \neq \hat{X}\right) \leq \varepsilon$.

[1] The expression derived in [5] was different and the tightness of alternative expression we present was shown in [4].

D. Progressively degraded Markov tree model

Finally, we lay down the family $\mathcal{P}$ of sources we will be handling in this work. We consider binary sources of length n, namely sources where the observations of each party takes values in $\{0,1\}^n$. Furthermore, we only consider symmetric models where marginal distribution of each party's observation is $\texttt{Ber}(1/2)$.

For two-parties, such a family of distribution corresponds to *binary symmetric sources* (BSSs) $\mathrm{P}_{X_1X_2}$ where X_1 is a random bit, U is $\texttt{Ber}(\delta)$, and $X_2 = X_1 \oplus U$, for some flipover probability $\delta \in [0,1]$. We denote the channel with X_1 as input and X_2 as output by $\texttt{BSC}(\delta)$.

A natural multiparty extension of this family is the family of *binary symmetric Markov tree* (BMT) sources where binary random variables $X_1, ..., X_m$ form a Markov tree with the joint distribution of any parent-child pair given by a BSS. Note that such a distribution can be represented by a weighted tree with edge weights $p_i \in [0,1]$, $1 \leq i \leq m-1$, representing the flipover probability for the corresponding parent-child edge. We will work with this representation throughout.

The source family we shall work with in this paper is a special case of BMTs – we term it a *progressively degraded BMT* (DMT). This comprises BMT distributions such that for ordered edge weights such that $h(p_1) \leq h(p_2) \leq \cdots \leq h(p_{m-1})$, the restriction of the tree to edges corresponding to weights $p_1, ..., p_k$ is also a tree, for every $2 \leq k \leq m-1$. Alternatively, we can view the weighted tree being generated by adding one edge at a time in such a way that every new edge has weight more than all the previously existing edges.

III. Main Result: A Completeness Theorem

We now present our main result which states that a universal SW protocol for BSSs can be used to construct a universal DE protocol for DMTs. Namely, the universal SW problem for BSSs is complete for universal DE for DMTs.

Formally, assume the existence of an (ε, η)-USW protocol $\pi = (e, d)$. Our UDE protocol, described below in Alg. 1, uses π in an oracle access with pause. In particular, it initiates m^2 copies of π and pauses and resumes a subset of them at various points. All running copies are in sync and proceed at one round at a time. Recall that by definition π sends $n\eta$ bits per round. When we pause a copy, it halts at its current round, and when it is resumed, it proceeds by incrementing rounds. Therefore, the number of bits sent by each copy is simply determined by the time (or rounds) for which it is executed. When Alg. 1 terminates party i, $i \in [m]$, has estimates $\hat{X}(j,i) \in \{0,1\}^n$, $j \in [m] \setminus \{i\}$, for observation $x(j) \in \{0,1\}^n$ of party j.

The result below characterizes the performance of UDE proposed in Alg. 1.

Theorem 1 (Completeness of USW for UDE)**.** *For $\varepsilon \in (0,1)$ and $\eta > 0$, let π be an (ε,η)-USW for n-length BSS. Then, Alg. 1 constitutes an $(m^2\varepsilon, m^2\eta)$-UDE for n-length DMTs.*

Algorithm 1 UDE from USW

```
Initial step. All pairs of parties $i, j \in [m]$ execute copies of the USW protocol $\pi$ with party $i$ communicating[2] by running the encoder $e$ with its data $x(i)$ and party $j$ executing decoder $d$ with its data $x(j)$ and communication sent by $i$. We denote the copy with $i$ as transmitter and $j$ as receiver by $\pi_{ij}$.
Root-pair discovery. Let $(i_0, j_0)$ denote the pair for which $\pi_{i_0 j_0}$ terminate first; if there are multiple, choose one arbitrarily. Pause all the other copies of $\pi$ except $\pi_{j_0 i_0}$, which is continued till termination.
    ▷ Parties $i_0$ and $j_0$ have exchanged $x(i_0)$ and $x(j_0)$.
$\mathcal{A} \leftarrow \{i_0, j_0\}$.
    ▷ $\mathcal{A}$ is the set of parties for which DE is complete.
Branch-growth iteration. while $\mathcal{A} \neq [m]$ do
    New branch discovery. For each $j \in \mathcal{A}$ and $i \in \mathcal{A}^c$, the pair $(i,j)$ resumes $\pi_{ij}$ until the protocol for any one pair, say $(i_1, j_1)$, terminates. Pause all the other copies of $\pi$.
        ▷ At this point, party $j_1 \in \mathcal{A}$ has an estimate of $x(i_1)$.
    Assimilation. First $\pi_{j_1 i_1}$ resumes till termination to make $x(j_1)$ available to party $i_1$. Next, party $i_1$ uses its estimate of $x(j_1)$ and transcripts of $\pi_{ij_1}$, $i \in \mathcal{A}$, to obtain estimates of $x(i)$ for every $i \in \mathcal{A}$. Finally, each party in $\mathcal{A}$ uses the transcript $\pi_{i_1 j_1}$ and its respective estimate of $x(j_1)$ to construct an estimate of $x(i_1)$.
        ▷ Now parties in $\mathcal{A} \cup \{i_1\}$ have exchanged data.
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{i_1\}$.
```

Proof. Consider a DMT representation by a tree $\mathcal{T}$ with edge weights $p_1, .., p_{m-1}$. We assume that $h(p_1) \leq ... \leq h(p_{m-1})$. We present the proof for the simple (infeasible) case when π is an ideal USW, *i.e.*, $\varepsilon = \eta = 0$. In this case, each copy of π terminates precisely when it reaches the required rate for the SW problem. Therefore, the root-pair discovery step concludes with data exchange for the pair with the least weight, namely that corresponding to the edge e_1 with weight p_1. At this point, the communication rate of each node is $h(p_1)$. Next, in the new branch discovery step, the node in the edge e_2 with weight p_2 that is not included in e_1 concludes first. At this point, the overall communication rate of nodes in $\mathcal{A}$ is $h(p_1)$ and those of outside $\mathcal{A}$ is $h(p_2)$. Then, a node in $\mathcal{A}$ (that on which e_2 is incident) continues to rate $h(p_2)$ to complete data exchange for the edge e_2 in the assimilation step. This process continues iteratively. When the ith edge has completed data exchange, the overall rate of communication for all the nodes is $\sum_{j=1}^{i-1} h(p_j) + (m-i+1)h(p_i)$, where the sum corresponds to the rate of nodes in $\mathcal{A}$ except the newly assimilated pair. Therefore, when the protocol terminates, the overall rate of communication is $\sum_{j=1}^{m-2} h(p_j) + 2h(p_{m-1})$. This rate equals the minimum communication needed for omniscience. Indeed, denoting by $X_1, ..., X_m$ random variables such that (X_i, X_{i+1}) correspond to the edge e_i with weight p_i, it can

[2] All bits are broadcasted to all the parties, with nodes communicating at equal rate.

be seen that $H(X_1,...,X_m|X_m)+H(X_m|X_1,...,X_{m-1}) = \sum_{j=1}^{m-2} h(p_j)+2h(p_{m-1}) \leq R^*(P_{X_1\dots X_m})$, where the final bound uses the expression for minimum communication for omniscience. But since this rate can be achieved by our protocol, it must equal $R^*(P_{X_1\dots X_m})$, establishing universal optimality of our protocol in the ideal case.

Moving now to the proof in the real world, the main challenge is handling the slack of η in rate that can change the order in which the pairs complete data exchange. In fact, even the pairs that do not have an edge between them in $\mathcal{T}$ can complete data exchange before a pair with edge. Nonetheless, we show that this change in order will not change the overall rate by much (depending on η). Denote by $1 < i_1,...,i_k$ the increasing sequence of indices in $[m-1]$ such that $h(p_{i_\ell})+\eta < h(p_{i_\ell+1})$, for $1 \leq \ell \leq k$. We show that with high probability our protocol must "sync" with the ideal protocol when edges e_{i_ℓ}s complete data exchange. Specifically, let $\mathcal{A}_1,...,\mathcal{A}_k,\mathcal{A}_{k+1}$ be the partition of nodes $[m]$ such that $\mathcal{A}_\ell$ comprises the nodes incident on edges e_j for $i_{\ell-1} < j < i_\ell$, the right-node of $e_{i_{\ell-1}}$ and the left-node of e_{i_ℓ} (where the direction is left to right from e_1 to e_{m-1}). Also, the rate of communication required for decoding a node in $A_{\ell+1}$ is no more than $h(p_{i_\ell})+(|\mathcal{A}_{\ell+1}|+1)\eta$. Due to lack of space, we delegate the proof details to a longer version, to be uploaded on arXiv.

□

IV. A construction using Polar codes

In this section, we present a construction of USW using Polar codes which can be extended to a UDE using our general scheme in Alg. 1. We refer the reader to [1], [2], [6] for background on Polar codes and SW compression using Polar codes.

A. USW using Polar Codes

A Polar encoder applies a linear transform T to an n-bit input vector and converts n independent copies of a given binary input channel W to a set of q "good" binary input channels on which the message $u \in \{0,1\}^q$ is transmitted almost error free, and $n-q$ "bad" channels on which the $(n-q)$-length sequence of *frozen bits* is transmitted. 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. At finite blocklength, the polarized channels can be arranged in increasing order of the Bhattacharyya parameter $Z\big(W_n^{(i)}\big)$, the "good channels" may be chosen by putting a threshold on the same. We adhere to this ordering.

For SW compression of a BSS (X_1,X_2) with flipover probability p, we may implement the asymptotically optimal encoder described earlier by extracting the frozen bits from $T^{-1}(X_1)$, whereas a simple SC decoder which has access to X_2 and these frozen bits can obtain an estimate $\hat{X}_1$ of X_1 [6]. Recall, that the number of good channels tends to the symmetric capacity $I(W_n)$ of vector channel W_n [1]. Thus, the asymptotic rate of communication needed for this SW scheme is $H(X_1)-I(W_n) = H(X_1|X_2) = h(p)$. This construction, however, relies on an exact value of p.

A similar problem of universality in channel coding regime can be solved using a HARQ code. We base our construction of USW on such a HARQ code, called the RT-Polar HARQ [3], and name it the RT-Polar Data Exchange code. Our construction tries to get the closest estimate of p from a set $\mathcal{G} = \{p_1,...,p_r\}$, $p_i < p_{i+1}$, of possible values of 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 $R_i \leq C(p_i)$ holds for every $1 \leq i \leq r$. Initially, the encoder assumes that the BSS can be described by a $\texttt{BSC}(p_1)$, thus, sends $n-nR_1$ frozen bits to the decoder over an error free channel. Under the same assumption, the decoder tries to estimate $\hat{X}_1$. If the receiver detects a decoding error, 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 value of p, so does the receiver. Accordingly, number of frozen bits are successively increased to $n-nR_i$, $R_i = R_1/i$, $i = 2,3,...,r$, in the ith iteration, where r denotes the maximum number of iterations. Note, the choice of $R_i = R_1/i$ is to facilitate ease of implementation, $R_i = R_1-(i-1)\Delta$ as discussed in [8] is another valid choice. Though, p can be anything in the interval $[p_i,p_{i+1}]$, the restrictions on p_i helps to limit the number of rounds allowed. After the i-th iteration the total number of bits transmitted by the source to the destination is $n-nR_i$. If $p = p_i$, for $n \to \infty$ the allowable rate R_i approaches $1-h(p_i)$. Hence, with ideal decoding error detection, the number of bits required at destination to recover X_1 tends to $nh(p_i)$ asymptotically, with no extra bits being sent.

Including a similar communication for USW of sending X_2 to X_1, the total rate is $2nh(p_i)$ asymptotically. We use an error detection mechanism similar to that of RT-Polar HARQ scheme of [3]. Specifically, the bottom t bits of the vector $T^{-1}(X_1)$ are sent on the error free channel along with the frozen bits, as a reference for error detection. The decoder compares these bits with their decoded value to detect an error. We present a detailed description of the destination side of the scheme, namely the RT-Polar DE scheme, in Alg. 2. Note that unlike its channel coding counterpart, for source coding recursive decoding of previous transmission at each step is not necessary.

The subroutine polar(x) performs the Polar encoding transform, append(a,b) appends the vector b to the vector a, and extract$(a,l,\texttt{POS})$ extracts l bits from specified $\texttt{POS}$ of vector a. The subroutine decode(y,f,nR,p) is a 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. The error detection takes place in step 10, where if the transmitted and decoded values of the bottom vector match, we accept the estimate of x and conclude the algorithm.

B. Extending to UDE and assimilation step improvements

We extend USW to a UDE using Alg. 1 where each copy of USW is executed until the decoder returns an ACK. While

Algorithm 2 RT-Polar DE decoder at the ith iteration.

```
Input: the destination vector $x_2$, error-free transmission in $i$th iteration $w_i$.
Output: ACK$_i$/NACK$_i$ and the decoded source vector $\hat{x}_1$
$f_i \leftarrow \emptyset$
for $j = 1, 2, \ldots, i$ do
    $f_i \leftarrow$ append$(w_j, f_i)$
$v_i \leftarrow$ decode$(x_2, f_i, nR_i, p_i)$
$\hat{u} \leftarrow$ append$(v_i, f_i)$
$H_i \leftarrow$ extract$(w_i, t, \texttt{TOP})$
$\hat{H}_i \leftarrow$ extract$(v_i, t, \texttt{BOTTOM})$
if $H_i = \hat{H}_i$ then
    $\hat{x} \leftarrow$ polar$(\hat{u})$
    return ACK, $\hat{x}$
else
    if $i < r$ then
        return NACK,
    else
        $\hat{x} \leftarrow$ polar$(\hat{u})$
        return $\hat{x}$
```

the USW estimates the relation between any two nodes, the UDE uses the order of discoveries made by it's constituent USWs to figure out the possible construction of the DMT. In our implementation we relegate the assimilation stages of all branch-growth iteration steps to a final decoding step which involves no communication. Here every node uses the previously received bits and their decoded estimates to decode the vectors they are missing when the algorithm concludes. At this step, it is critical to carefully follow the history of the communication and complete final decoding in the order of discovery of nodes, by backtracking the received communication.

A shortcoming of our theoretical construct in Alg. 1 is the assimilation step. Here, errors add-up significantly when we use an estimate of a vector to decode another vector, causing performance degradation with increase in number of nodes. In practice, we circumvent this difficulty by communicating an additional CRC hash for each vector.

C. Numerical evaluation of performance

We present the performance of RTPolar DE for $r = 3$, $p_1 = 0.03$, $p_2 = 0.11$, $p_3 = 0.17$. These probabilities correspond to $\texttt{BSC}(p)$ with capacities roughly $4/5$, $1/2$, and $1/3$. In our Polar Code construction, the set of good bit channels $\mathcal{I}_{n,p}$ is selected to ensure that $\max\{Z(W_n^{(j)}(p)) : j \in \mathcal{I}_{n,p}\} \le \delta$, where $\delta = 0.05$ corresponds to $R_1 = 1/2$ approximately. Our choice of parameters corresponds to the analysis in [3]. Figure 1 illustrates the performance of RT-Polar DE scheme as a USW and compares it to a genie-aided variant which detects decoding error without an error. Here, $l(\mathcal{P})$ denotes the communication rate considering correct exchanges only, where $\mathcal{P}$ is the set of flipover probabilities associated with the edges of the DMT. In figure 2, we present the performance of extension of the scheme to three parties with genie-aided as well as CRC-aided final error detection. Figure 3 illustrates the performance of the algorithm for a 4 party path with $p'' \le p', p'' \le p'''$, which constitutes a DMT. We have considered, $p_1 = 0.03$, $p_2 = 0.15$, $p_3 = 0.3$. Our simulations suggest that the RT-Polar DE scheme and its extensions successfully capture the trend of optimal communication required to perform a universal data exchange. We expect better performance of RT-Polar DE if more efficient coding schemes are used.

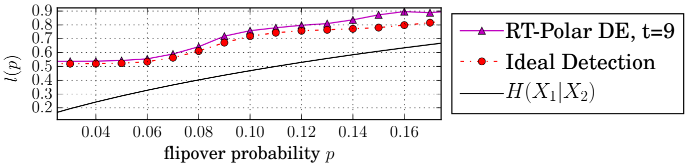

Fig. 1. Performance comparison of RT-Polar DE for 2 parties, with $t = 9$, $\delta = 0.05$, $n = 512$.

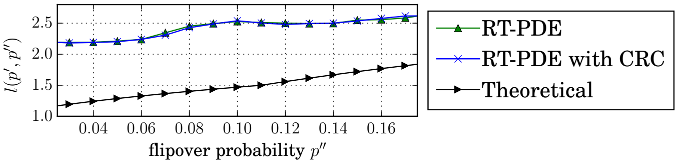

Fig. 2. Performance comparison of RT-Polar DE for 3 party DMT, with $t = 9$, $\delta = 0.05$, $n = 512$, $p' = 0.11$.

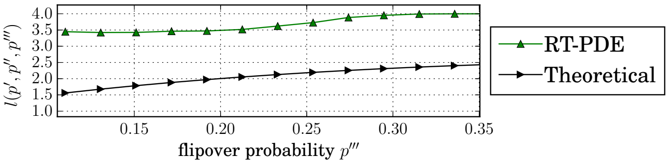

Fig. 3. Performance comparison of RT-Polar DE for 4 party, with $t = 50$, $\delta = 0.05$, $n = 512$, $p' = 0.07$, $p'' = 0.03$.

References

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

[2] ——, "Source polarization," in *Inform. Theory Proceedings (ISIT), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 899–903.

[3] S. S. Banerjee and H. Tyagi, "RT-Polar: An harq scheme with universally competitive rates," in *IEEE Inform. Theory Workshop (ITW)*, 2018.

[4] C. Chan and L. Zheng, "Mutual dependence for secret key agreement," *Proc. Annual Conference on Inform. Sciences and Systems (CISS)*, 2010.

[5] I. Csiszár and P. Narayan, "Secrecy capacities for multiple terminals," *IEEE Trans. Inf. Theory*, vol. 50, no. 12.

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

[7] E. Kushilevitz and N. Nisan, *Communication Complexity*. New York, NY, USA: Cambridge University Press, 1997.

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

[9] D. Slepian and J. Wolf, "Noiseless coding of correlated inform. source," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[10] H. Tyagi and S. Watanabe, "Optimality of the recursive data exchange protocol," *Proc. IEEE Int. Symp. Inf. Theory*, 2017.

[11] ——, "Universal multiparty data exchange and secret key agreement," *IEEE Trans. on Inform. Theory*, vol. 63, no. 7, pp. 4057–4074, 2017.