A Lower Bound on the Essential Interactive Capacity of Binary Memoryless Symmetric Channels

Assaf Ben-Yishai, Young-Han Kim, Or Ordentlich and Ofer Shayevitz

***Abstract*—The essential interactive capacity of a discrete memoryless channel is defined in this paper as the maximal rate at which the transcript of any interactive protocol can be reliably simulated over the channel, using a deterministic coding scheme. In contrast to other interactive capacity definitions in the literature, this definition makes no assumptions on the order of speakers (which can be adaptive) and does not allow any use of private / public randomness; hence, the essential interactive capacity is a function of the channel model only. It is shown that the essential interactive capacity of any binary memoryless symmetric (BMS) channel is at least 0.0302 its Shannon capacity. To that end, we present a simple coding scheme, based on extended-Hamming codes combined with error detection, that achieves the lower bound in the special case of the binary symmetric channel (BSC). We then adapt the scheme to the entire family of BMS channels, and show that it achieves the same lower bound using extremes of the Bhattacharyya parameter.**

I. Introduction

In the classical Shannon one-way communication problem, a transmitter (Alice) wishes to send a message reliably to a receiver (Bob) over a memoryless noisy channel. She does so by mapping her message into a sequence of channel inputs (codeword) in a predetermined way, which is corrupted by the channel and then observed by Bob, who tries to recover the original message. The *Shannon capacity* of the channel, which is the maximal number of message bits per channel use that Alice can convey to Bob with vanishingly low error probability, quantifies the most efficient way to do so. In the two-way channel setup [1], both parties draw independent messages and wish to exchange them over a two-input two-output memoryless noisy channel, and the Shannon capacity (region) is defined similarly. Unlike the one-way case, both parties can now employ adaptive coding by incorporating their respective observations of the past channel outputs into their transmission processes. However, just as in the one-way setup, the messages they wish to exchange are determined before communication begins. In other words, if Alice and Bob had been connected by a noiseless bit pipe, they could have simply sent their messages without any regard to the message of their counterpart.

In a different two-way communication setup, generally referred to as *interactive communication*, the latter assumption is no longer held true. In this interactive communication setup, Alice and Bob do not necessarily wish to disclose all their local information. What they want to tell each other depends, just like in human conversation, on what the other would tell them. A simple instructive example (taken from [2]) is the following. Suppose that Alice and Bob play chess remotely, by announcing their moves over a communication channel (using,

O. Ordentlich and A. Ben-Yishai are with the School of Computer Science and Engineering, Hebrew University of Jerusalem, Israel. O. Shayevitz is with the Department of EE–Systems, Tel Aviv University, Tel Aviv, Israel. Y.-H. Kim is with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093 USA. Most of this work has been performed while A. Ben-Yishai was with the Department of EE–Systems, Tel Aviv University. Emails: {assafbster@gmail.com, yhk@ucsd.edu, or.ordentlich@mail.huji.ac.il, ofersha@eng.tau.ac.il} The work of A. Ben-Yishai was supported by an ISF grant no. 1367/14. The work of O. Orderntlich was supported by an ISF grant no. 1791/17. The work of O. Shayevitz was supported by an ISF grant no. 1495/18 and an ERC grant no. 639573. This paper was presented in part at ISIT 2019.

say, 12 bits per move, which is clearly sufficient). If the moves are conveyed without error, then both parties can keep track of the state of the board, and the game can proceed to its termination. The sequence of moves occurring over the course of this noiseless game is called a *transcript*, and it is dictated by the *protocol* of the game, which constitutes Alice and Bob's respective strategies determining their moves at any given state of the board.

Now, assume that Alice and Bob play the game over a noisy two-way channel, yet wish to simulate the transcript as if no noises were present. In other words, they would like to communicate back and forth in a way that ensures, once communication is over, that the transcript of the noiseless game can be reproduced by to both parties with a small error probability. They would also like to achieve this goal as efficiently as possible, i.e., with the least number of channel uses. One direct way to achieve this is by having both parties describe their entire protocol to their counterpart, i.e., each and every move they might take given each and every possible state of the board. This reduces the interactive problem to a non-interactive one, with the protocol becoming a pair of messages to be exchanged. However, this solution is grossly inefficient; the parties now know much more than they really need in order to simply reconstruct the transcript. At the other extreme, Alice and Bob may choose to describe the transcript itself by encoding each move separately on the fly, using a short error correcting code. Unfortunately, this code must have some fixed error probability and hence an undetected error is bound to occur at some unknown point, causing the states of the board held by the two parties to diverge, and rendering the remainder of the game useless. It is important to note that if Alice and Bob had wanted to play sufficiently many games in parallel, then they could have used a long error-correcting code to simultaneously protect the set of all moves taken at each time point, which in principle would have let them operate at the one-way Shannon capacity (which is the best possible). The crux of the matter therefore lies in the fact that the interactive problem is *one-shot*, namely, only a single instance of the game is being played.

In light of the above, it is perhaps surprising that it is nevertheless possible to simulate any one-shot interactive protocol using a number of channel uses that is proportional to the length of the transcript. In other words, a positive rate of simulation is achievable whenever the Shannon capacity is nonzero. This fact was initially proved by Schulman [3], who was also the first to introduce the notion of interactive communication over noisy channels. However, this rate of reliable simulation has never been quantified; it is only known to be some nonzero fraction of the Shannon capacity. Moreover, several subtly different notions of achievability exist in the literature, depending in particular on various assumptions on the structure of the protocol and on the randomness resources (see Section IV). In order to circumvent these issues, we define a stringent notion of achievability that depends only on the channel; in particular, our definition does not make any assumptions on the simulated protocol, and does not allow the use of public or private randomness. We show that the maximal achievable rate under this definition, which we call the *essential interactive capacity*, is at least a 0.0302 fraction of the Shannon capacity for the entire family of binary memoryless symmetric (BMS) channels, which includes in particular the binary symmetric channel (BSC).

The rest of the paper is organized as follows. In Section II we present the problem formulation and a high level description of the techniques. In Section III we present the main contribution. In Section IV we put our work in context of existing results in the literature. We provide some necessary preliminaries in Section V, and then state the main results in Section VI. The coding scheme used in the proof for the binary symmetric channel (BSC) is presented and analyzed in Sections VII and VIII respectively, and then generalized to binary memoryless symmetric (BMS) channels in Section IX. Finally, in Section X, we explain how the randomized coding scheme can be modified to be fully deterministic.

II. Problem Formulation

A. Interactive Communication and the Essential Interactive Capacity

A *length-n interactive protocol* is the triplet $\boldsymbol{\pi} \triangleq (\boldsymbol{\phi}^{\text{Alice}}, \boldsymbol{\phi}^{\text{Bob}}, \boldsymbol{\psi})$, where

$$\begin{aligned}\boldsymbol{\phi}^{\text{Alice}} &\triangleq \left\{\phi_i^{\text{Alice}} : \{0,1\}^{i-1} \mapsto \{0,1\}\right\}_{i=1}^{n} \\ \boldsymbol{\phi}^{\text{Bob}} &\triangleq \left\{\phi_i^{\text{Bob}} : \{0,1\}^{i-1} \mapsto \{0,1\}\right\}_{i=1}^{n} \\ \boldsymbol{\psi} &\triangleq \left\{\psi_i : \{0,1\}^{i-1} \mapsto \{\text{Alice}, \text{Bob}\}\right\}_{i=1}^{n}.\end{aligned}$$

The functions $\boldsymbol{\phi}^{\text{Alice}}$ are known only to Alice, and the functions $\boldsymbol{\phi}^{\text{Bob}}$ are known only to Bob. The *speaker order functions* $\boldsymbol{\psi}$ are known to both parties. The *transcript* $\boldsymbol{\tau}$ associated with the *input* protocol $\boldsymbol{\pi}$ is sequentially generated by Alice and Bob as follows

$$\tau_i = \begin{cases} \phi_i^{\text{Alice}}(\boldsymbol{\tau}^{i-1}) & \sigma_i = \text{Alice} \\ \phi_i^{\text{Bob}}(\boldsymbol{\tau}^{i-1}) & \sigma_i = \text{Bob} \end{cases} \tag{1}$$

where σ_i is the identity of the speaker at time i, which is given by:

$$\sigma_i = \psi_i(\boldsymbol{\tau}^{i-1}). \tag{2}$$

In the *interactive simulation problem* Alice and Bob would like to *simulate* the transcript $\boldsymbol{\tau}$, by communicating back and forth over a noisy memoryless channel $P_{Y|X}$. Specifically, we restrict our discussion to channels with a binary input alphabet $\mathcal{X} = \{0,1\}$, and a general (possibly continuous) output alphabet $\mathcal{Y}$. Note that while the order of speakers in the input protocol itself might be determined on the fly (by the sequence of functions $\boldsymbol{\psi}$), we restrict the simulating protocol to use a predetermined order of speakers, due to the fact that our physical channel model does not allow simultaneous transmissions (this point is elaborated in Section IV).

To achieve their goal, Alice and Bob employ a length-N coding scheme Σ that uses the channel N times. The coding scheme consists of a disjoint partition $\tilde{A} \subseteq \{1, ..., N\}$, $\tilde{B} = \{1, ..., N\} \backslash \tilde{A}$, where $\tilde{A}$ (resp. $\tilde{B}$) is the set of time indices where Alice (resp. Bob) speaks. This disjoint partition can be a function of $\boldsymbol{\psi}$, but not of $\boldsymbol{\phi}^{\text{Alice}}, \boldsymbol{\phi}^{\text{Bob}}$. At time $j \in \tilde{A}$ (resp. $j \in \tilde{B}$), Alice (resp. Bob) sends some *deterministic* function X_j of $(\boldsymbol{\phi}^{\text{Alice}}, \boldsymbol{\psi})$ (resp. $(\boldsymbol{\phi}^{\text{Bob}}, \boldsymbol{\psi})$), and of everything she has received so far from her counterpart. The transmitted X_j is observed by Bob (resp. Alice) through the channel $P_{Y|X}$, whose output is denoted by Y_j. Note that we assume that $Y_j - X_j - (X^{j-1}, Y^{j-1})$ forms a Markov chain. The rate of the scheme is $R = \frac{n}{N}$ bits per channel use. When communication terminates, Alice and Bob produce their *simulations* of the transcript τ, denoted by $\hat{\boldsymbol{\tau}}_A(\Sigma, \boldsymbol{\phi}^{\text{Alice}}, \boldsymbol{\psi}) \in \{0,1\}^n$ and $\hat{\boldsymbol{\tau}}_B(\Sigma, \boldsymbol{\phi}^{\text{Bob}}, \boldsymbol{\psi}) \in \{0,1\}^n$ respectively. The error probability attained by the coding scheme is the probability that either of these simulations is incorrect, i.e.,

$$\begin{aligned}&P_e(\Sigma, \boldsymbol{\pi}) \triangleq \\ &\Pr\left(\hat{\boldsymbol{\tau}}_A(\Sigma, \boldsymbol{\phi}^{\text{Alice}}, \boldsymbol{\psi}) \neq \boldsymbol{\tau} \ \vee \ \hat{\boldsymbol{\tau}}_B(\Sigma, \boldsymbol{\phi}^{\text{Bob}}, \boldsymbol{\psi}) \neq \boldsymbol{\tau}\right).\end{aligned}$$

A rate R is called *achievable* if there exists a sequence Σ_n of length-N_n coding schemes that operate on length-n input protocols $\boldsymbol{\pi}$, where $\frac{n}{N_n} \geq R$, and attain a vanishing worst-case error probability, i.e.,

$$\lim_{n \to \infty} \max_{\text{protocols } \boldsymbol{\pi} \text{ of length } n} P_e(\Sigma_n, \boldsymbol{\pi}) = 0.$$

Accordingly, we define the *essential interactive capacity* $C_{\text{I}}(P_{Y|X})$ as the supremum of all achievable rates for the channel $P_{Y|X}$. This definition is more conservative than all other interactive capacity definitions appearing in the literature, as further discussed in Section IV. In particular, note that our capacity definition makes worst case assumptions on the input protocol, and is hence a function of the channel model only. We also note in passing that our assumptions on channel access model are conservative and not worst case, as we permit any predetermined scheduling of speakers (more on that below). This approach makes sense practically, since there seems to be no fundamental reason to limit Alice and Bob in terms of which coding scheme they can use. Moreover, taking a worst case approach in terms of channel access can lead to trivialities, since there exist pessimistic access schedules (e.g., allocating only a single channel use for Alice) that would render the capacity zero.

For simplicity of exposition, we restrict our dis-

cussion from this point on to binary-input channels. Since at least n bits need to be exchanged in order to reliably simulate a general length-n input protocol, the essential interactive capacity for such channels must satisfy $C_{\mathrm{I}}(P_{Y|X}) \leq 1$. In the special case of a noiseless channel, i.e., where the output deterministically reveals the input bit, and assuming that the order of speakers is predetermined (namely $\boldsymbol{\psi}$ contains only constant functions), this upper bound can be trivially achieved; Alice and Bob can simply evaluate and send τ_i sequentially according to (1) and (2). Note however, that if the order of speakers is general, then this is not a valid solution, since we required the order of speakers in the coding scheme to be fixed in advance. Nevertheless, any general length-n input protocol can be sequentially simulated using the channel $2n$ times with alternating order of speakers, where each party sends a dummy bit whenever it is not their time to speak. Conversely, a factor two blow-up in the input protocol length in order to account for a non pre-determined order of speakers is also necessary. To see this, consider an example of an input protocol where Alice's first bit determines the identity of the speaker for the rest of time; in order to simulate this protocol using a predetermined order of speakers, it is easy to see that at least $n-1$ channel uses must be allocated to each party in advance. We conclude that under our restrictive capacity definition, the essential interactive capacity of a noiseless (binary-input) channel is exactly $\frac{1}{2}$. It is instructive to note that for a noiseless channel, one could have permitted the order of speakers to be determined on-the-fly, avoiding the need to pre-allocate the channel and eliminating the factor $1/2$ penalty. However, the truly noiseless channel is a singular case, since for any arbitrarily small channel error probability, using an adaptive order of speakers would yield channel access collisions, which are not supported in our channel model. We further elaborated on this point in Section IV.

When the channel is noisy, a tighter trivial upper bound holds:

$$C_{\mathrm{I}}(P_{Y|X}) \leq \frac{1}{2} C_{\mathrm{Sh}}(P_{Y|X}), \tag{3}$$

where $C_{\mathrm{Sh}}(P_{Y|X})$ is the Shannon capacity of the channel. To see this, consider the same example given above, and note that each party must have sufficient time to reliably send $n-1$ bits over the noisy channel. Hence, the problem reduces to a pair of one-way communication problems, in which the Shannon capacity is the fundamental limit. We remark that it is reasonable to expect the bound (3) to be loose, since general input protocols cannot be trivially reduced to one-way communication as the parties cannot generate their part of the transcript without any interaction. However, the tightness of the bound remains a wide open question. We note in passing that if we had considered simulating only protocols with a predetermined order of speakers, the corresponding upper bound would have been $C_{\mathrm{I}}(P_{Y|X}) \leq C_{\mathrm{Sh}}(P_{Y|X})$.

Remark 1. *[The notion of determinism in interactive coding schemes] Let us briefly discuss the difference between deterministic and randomized coding schemes for interactive communication. A deterministic coding scheme is one where the transmission functions used by Alice and Bob to generate their next channel inputs are fixed and given in advance; in other words, the channel inputs generated by both parties are solely determined by the input protocol and the channel outputs. A randomized coding scheme, on the other hand, is allowed to use random bits from an exogenous source; Namely, Alice / Bob pick a random function to apply to their data (which includes all their past observations) each time, and this function can be different even if the data it is applied to is the same.*

We note that in principle, when working over stochastic memoryless channels, any randomized scheme can be converted into a deterministic one by extracting the needed random bits from the noisy channel outputs (e.g., using [4], [5]). However, this procedure incurs a loss in rate due to the overhead of randomness extraction and possibly communication of randomness. While semantically, such a scheme might appear to be randomized, we note that it is in fact deterministic, since all the transmission functions used by the parties (including the ones used for randomness extraction) are fixed in advance. In a related context, see for example [6], where the authors construct an optimal

randomized finite-state machine to estimate the bias of a coin, and then derandomize it by extracting the necessary random bits from the observations themselves, with a modest penalty in performance. We further observe that one could potentially define a more stringent notion of deterministic coding schemes, where the parties' inputs are not allowed to depend on the random channel outputs. However, while this definition would disallow any randomness extraction, it would also remove the interactive component from the problem.

B. Channel Models

The first noisy channel model we consider is the memoryless binary symmetric channel with crossover probability $0 \leq \varepsilon \leq \frac{1}{2}$, BSC($\varepsilon$). The input to output relation of the BSC(ε) is given by

$$Y = X \oplus Z$$

where $X, Y, Z \in \mathbb{F}_2$, $\oplus$ denotes addition over $\mathbb{F}_2$. Z is statistically independent of X with $\Pr(Z = 1) = \varepsilon$. We denote its Shannon capacity by

$$C_{\mathrm{Sh}}(\varepsilon) \triangleq 1 - h(\varepsilon),$$

where $h(\varepsilon) \triangleq -\varepsilon \log \varepsilon - (1-\varepsilon)\log(1-\varepsilon)$ is the binary entropy function, and $\log(x) \triangleq \log_2(x)$. We also use $C_{\mathrm{I}}(\varepsilon)$ to denote the essential interactive capacity of the BSC(ε).

A richer channel model which is commonly used in the coding literature is the binary memoryless symmetric (BMS) channel [7]–[11]. While several equivalent definitions exist, the following definition of a BMS channel as a collection of BSC with various crossover probabilities [12], is most convenient for the derivations in this paper:

Definition 1. *[BMS channels] A memoryless channel with binary input X output Y and a conditional distributions $P_{Y|X}$ is called* binary memoryless symmetric *channel (BMS($P_{Y|X}$)) if there exists a sufficient statistic of Y for X: $g(Y) = (X \oplus Z_T, T)$, where (T, Z_T) are statistically independent of X, Z_T is a binary random variable with $\Pr(Z_T = 1|T = t) = t$, and $0 \leq T \leq \frac{1}{2}$ with probability one.*

Consequently, the Shannon capacity of BMS($P_{Y|X}$) channel is

$$C_{\mathrm{Sh}}(P_{Y|X}) = 1 - \mathbb{E}h(T).$$

The simplest example for a BMS channel is the BSC(ε) for which $T = \varepsilon$ with probability one. The binary erasure channel with erasure probability ϵ, BEC(ϵ), can be cast as a BMS channel taking $T = \frac{1}{2}$ with probability ϵ and $T = 0$ with probability $1 - \epsilon$. It is in place to note, however, that in an actual BEC, a Bernoulli($1/2$) bit is not produced when $T = 1/2$ (this subtle point is further discussed in Subsection X-C). The binary additive white Gaussian noise (BiAWGN) channel, $Y = X + Z$ where $X \in \{-1, +1\}$ and $Z \sim \mathcal{N}(0, \sigma^2)$ is statistically independent of X, is also a BMS where T is a continuous random variable on $[0, 1/2]$ (see [9, Chapter 4]).

III. Main Contribution

In this paper, we derive a lower bound on the essential interactive capacity of any BMS channel, which depends only on its Shannon capacity. In particular, we show that the essential interactive capacity is always at least 0.0302 of the Shannon capacity, uniformly for all BMS channels. Indeed, since $C_{\mathrm{I}}(P_{Y|X}) \leq \frac{1}{2} C_{\mathrm{Sh}}(P_{Y|X})$ always holds (and is tight for noiseless BMS), then using the Shannon capacity as a yardstick is intuitively appealing, and our lower bound can be interpreted as saying that the "cost of interactiveness" is not too large.

Theorem 1. *For any BMS($P_{Y|X}$) channel with positive Shannon capacity $C_{\mathrm{Sh}}(P_{Y|X})$ and essential interactive capacity $C_{\mathrm{I}}(P_{Y|X})$*

$$\frac{C_{\mathrm{I}}(P_{Y|X})}{C_{\mathrm{Sh}}(P_{Y|X})} \geq 0.0302.$$

Note that Theorem 1 also applies to the special case of the BSC with any crossover probability. In fact, we first prove Theorem 1 for the BSC case, and them extend the result to general BMS channels.

The first step in the proof is standardly symmetrizing the order of speakers in the input protocol by possibly adding dummy transmissions, such that

Alice speaks at odd times, and Bob speaks at even times, namely resulting in a modified protocol where $\psi = \{\text{Alice, Bob, Alice, Bob, } \dots\}$. In the sequel, we refer to this order of speakers as *bit-vs.-bit*. This reduces the rate by a factor of two at most. We then use a *rewind-if-error* scheme in the spirit of [3], [13], designed for simulating the transcript of protocols with bit-vs.-bit order of speakers. As explained in the chess game example, the transcript bits of an interactive protocol should in general be decoded instantaneously, which implies that error correction codes (that typically use long blocks) cannot be straightforwardly used. Instead, rewind-if-error schemes are based on *uncoded transmission*, followed by error detection and retransmission. Namely, the transcript is simulated in blocks, as if no errors are present. Then, an error detection phase takes place, initiating the retransmission of the block whenever errors are detected. The scheme presented in Sections VII and VIII of this paper is based on *layered* error detection and retransmission. The rate of the proposed scheme is shown to be mostly effected by the efficiency of the error detection in the first layer. Thus we use extended-Hamming codes for error detection at that layer only, and a standard randomized error detection [14] at higher layers.

Our scheme is premised on the assumption that the channel is unlikely to introduce any errors within a single block. If this is not the case, we first standardly apply repetition coding in order to reduce the error rate to the desired level; crucially, we show that a sufficient number of repetitions in the BMS case is inversely proportional to the Shannon capacity of the channel. Accounting for this repetition overhead, calculating the rate of the rewind-if-error coding scheme, and judicially tuning its parameters, we show that this scheme yields the lower bound of Theorem 1.

Finally, while the scheme delineated above is randomized, we show that it can be converted to a fully deterministic scheme with an asymptotically vanishing rate loss, which makes our bound applicable in the essential interactive capacity setting. To that end, using a careful concentration analysis appearing in Appendix B, we first show that the number of random bits required by our scheme is only $o(n)$. Then, we harvest these bits from the channel via standard techniques, using only $o(n)$ channel uses. This process, which makes our scheme completely deterministic, clearly has a negligible effect on its overall rate.

IV. Connections to the Existing Work

In this section, we put our definition of essential interactive capacity in context of the existing literature. While the classical Shannon capacity of a one-way channel has single agreed-upon definition that depends on the channel model $P_{Y|X}$ only, the same is not true in the interactive setting, where various distinct notions of capacity exist, drastically depending on different possible assumptions. Let us review these assumptions, and point out that our definition is always on the more restrictive side.

- *Order of speakers*. One can assume that the input protocol, π, has either a predetermined order of speakers, or a general (adaptive) one. For a predetermined order, one can further assume that it has some fixed period (e.g., bit-vs.-bit). Our capacity definition does not restrict the order of speakers, hence our lower bound applies in any such setting (and for example, would increase by a factor of two if the order of speakers is bit-v.s-bit).
- *Randomness resources*. In their coding scheme (simulating protocol), Alice and Bob can be allowed to use some exogenous source of (public or private) randomness, in which case the scheme is called *randomized*, or are not allowed to use any exogenous randomness, in which case the scheme is called *deterministic*. We emphasize that in the deterministic case, the channel inputs are uniquely determined by the input protocol and the noisy channel output sequences. Our capacity definition makes the more stringent assumption of allowing only deterministic schemes, hence our lower bound applies to all cases.
- *Rate definition*: The coding scheme can be either fixed-length or variable-length. In the fixed-length case, the number of allocated channel uses is determined in advance, and the

rate is simply the ratio between the protocol length and the number of channel uses. In the variable-length case, the length of the protocol or the number of channel uses is allowed to be random, and the rate is then the ratio between the expected protocol length and the expected number of channel uses (though worst case length analysis also appears in the literature, for example [15], [16]). Our capacity definition adopts the more stringent fixed-length setting, hence our lower bound applies to all cases.

- *Physical channel model*: There are two distinct assumptions that can be made on the underlying structure of the channel. In one setting [3], [13], [17], Alice and Bob are not allowed (at the physical level) to simultaneously access the channel; they must decide in advance who uses the channel at each time point. In another (richer) setting [18], [19], Alice and Bob communicate over a general two-way channel [1], which means that they both input a symbol to the channel at any given time. In the interactive communication literature, a certain two-way channel has received attention. In this model, Alice and Bob each have *three* input symbols $\{0, 1, \mathsf{silence}\}$, and binary output symbols. A party that is not silent receives a zero. If one party is silent and the other is not, the silent one sees the input of its counterpart via a BSC. If both are silent, they observe uniform independent noise[1]. Our capacity definition adopts the more basic setting where no simultaneous channel access is allowed; since any two-way channel can be used this way, our lower bounds essentially applies to all cases.
- *Input protocol*: In the interactive communication literature, it is commonly assumed that the redundancy of the coding scheme is measured with respect to the communication complexity of a function, and the interactive capacity corresponds to the worst case blow-up over all functions (as further explained below). Alternatively, as suggested in this paper, one can measure the redundancy of the coding scheme with respect to *any protocol* (unrelated to any optimal function computation problem), and then the (essential) interactive capacity corresponds to the worst case blow-up over all protocols. Since our capacity definition normalizes by the length of the input protocol, it is at least in principle stricter than the one using communication complexity, and hence our lower bounds apply in both cases.

[1]This has in fact been considered in the adversarial setting, where in the case that both parties are silent, it was assumed that they observe undetermined symbols. What we described above is arguably the most natural way to adapt this adversarial assumption to the probabilistic setting.

Let us now review the main relevant literature. The interactive communication problem introduced by Schulman [3], [17] is motivated by Yao's communication complexity scenario [20]. In that latter scenario, the input of a function f is distributed between Alice and Bob, who wish to compute f with negligible error by exchanging (noiseless) bits using some interactive protocol. The length of the shortest protocol achieving this is called the *communication complexity* of f, and denoted by $CC(f)$. In Schulman's (random) interactive communication setup, Alice and Bob must achieve their goal by communicating through a pair of independent noisy channels, where the physical model does not allow simultaneous transmissions. For that setup, Schulman showed that one can attain this goal with negligible error, using only a constant blow-up in the length of the communication.

In [13], Kol and Raz considered the interactive communication problem, with no simultaneous transmissions, over a BSC(ε). They denoted the minimal *expected* length of a coding scheme computing f with a negligible error probability, by $CC_\varepsilon(f)$. They then defined the corresponding interactive capacity as:

$$C_{\mathrm{I}}^{\mathsf{KR}}(\varepsilon) \triangleq \lim_{n\to\infty} \min_{f:CC(f)=n} \frac{n}{CC_\varepsilon(f)}. \qquad (4)$$

with the additional assumption that the order of speakers in the input protocol is predetermined. They proved that

$$C_{\mathrm{I}}^{\mathsf{KR}}(\varepsilon) \leq 1 - \Omega\left(\sqrt{h(\varepsilon)}\right). \qquad (5)$$

in the limit of $\varepsilon \to 0$. They further proved that a rate

of $1-O(\sqrt{h(\varepsilon)})$ is achievable under an additional assumption that the order of speakers in the input protocol is has a small period. The assumption on the order of speakers is crucial. Indeed, consider again the example where the function f is either Alice's input or Bob's input as decided by Alice. In this case, the communication complexity with a predetermined order of speakers is double that without this restriction, and hence considering such protocols renders $C_{\mathrm{I}}^{\mathsf{KR}}(\varepsilon) \leq \frac{1}{2}$. For further discussion on speaking order impact as well as channel models that allow collisions, see [19]. Note that our definition of the BSC essential interactive capacity is stricter than (4), at least in principle, both since the latter does not consider adaptive input protocols, and also since we measure our blow-up w.r.t. the length of the entire transcript. For this reason, $C_{\mathrm{I}}(\varepsilon) \leq C_{\mathrm{I}}^{\mathsf{KR}}(\varepsilon)$, hence our lower bound applies to $C_{\mathrm{I}}^{\mathsf{KR}}(\varepsilon)$ as well (and also achieves the asymptotic behavior (4) when simulating bit-vs.-bit protocols). Our capacity definition further enjoys the property of being decoupled from any source coding problem such as function communication complexity.

For a fixed nonzero ε, the coding scheme presented in [3] (which precedes [13]) implies that $C_{\mathrm{I}}(\varepsilon) \geq \alpha \cdot C_{\mathrm{Sh}}(\varepsilon)$ for some universal constant α, but the constant has not been computed (and to the best of our knowledge, has not been computed for any scheme hitherto). Both [3] and [13] based their proofs on rewind-if-error coding schemes, i.e., schemes based on a *hierarchical* and *layered* error detection and appropriate retransmissions, which is also the approach we take in this paper.

In [19], Haeupler considered a different physical channel model where Alice and Bob can access the channel simultaneously and have three input symbols (as essentially described above). In this setup, he showed that a rate of $1-O(\sqrt{\varepsilon})$ is achievable for any alternating input protocol, which is higher than the upper bound (5). His results also hold in the more difficult adversrial setting assuming shared randomness, and reduces slightly to $1-O\left(\sqrt{\varepsilon}\log\log\frac{1}{\varepsilon}\right)$ when no randomness is available.

Let us now discuss the issue of randomness resources. The scheme in [3] requires only private randomness, while [13] requires public randomness. It is interesting to note that Schulman's *tree code* scheme [17] is not randomized. However, it is not designed to be rate-efficient, and for example does not achieve the lower bound in [13]. A non-random coding scheme was recently proposed by Gelles et. al. [21] based on a concatenation of a derandomized interactive coding scheme and a tree-code. This scheme achieves a rate $1-O(\sqrt{h(\varepsilon)})$ which is also the rate of the rewind-if-error scheme in this paper in the limite of $\varepsilon \to 0$ as stated in Corrolary 1. The rewind-if-error scheme presented in this paper is inspired by the scheme in [13], yet its error detection mechanism is not based on random hashes, but rather on extended-Hamming codes and randomized (yet structured) error detection. The deterministic coding scheme presented in Section X is not based on derandomization as in [21], but rather on suitably adapting the error detection and using concentration analysis to show that it requires only small number of random bits. These bits are then extracted from the noisy channels in a standard way using a small number of channel uses, which are taken into account in the overall rate calculation. We emphasize that our coding scheme is fully deterministic, namely, the channel inputs generated by Alice and Bob are uniquely determined by the input protocol and the channel noise sequences only.

Other channel models have been addressed in the literature. Much work has been dedicated to the adversarial setting, where the channel is controlled by an adversary with some limited jamming budget, see for example [15], [17]–[19]. It is important to note that the rewind-if-error approach and the randomness extraction ideas we use, do not apply in adversarial settings. More recently, interactive communication over channels with noiseless feedback has been studied in [22].

To summarize the discussion above, there are various distinct setups and sets of assumptions one may wish to consider when studying interactive communication, which can have significant effect on the fundamental limits. Our definition of capacity, and its corresponding lower bound, are based on

the most restrictive set of assumptions: the order of speakers in the input protocol can be adaptive, but is predetermined in the coding scheme; the coding is fixed-length and the blow-up is computed relative to the length of the input protocol; and no private or public randomness are allowed. Consequently, our capacity lower bounds remain valid for any other set of assumptions.

Finally, we note in passing that the current study extends our preliminary results presented in [23] in the following aspects: i) The error detection in the scheme is structured and is not based on random hashes. ii) The rate of the resulting scheme is improved and consequently the lower bound for the ratio between the essential interactive capacity and the Shannon capacity is also improved. iii) The scheme described in this paper deterministic. iv) The results are generalized from the BSC to arbitrary BMS channels.

V. Preliminaries

Let $D(P||Q) \triangleq \sum_{x\in\mathcal{X}} P(x)\log\frac{P(x)}{Q(x)}$ denote the Kullback-Leibler Divergence between the distributions $P(\cdot)$ and $Q(\cdot)$. Let $d(p||q) \triangleq p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$ denote the Kullback-Leibler Divergence between two Bernoulli random variables with probabilities p and q. In the sequel we use $\mathbb{1}(\cdot)$ to denote the indicator function, which equals one if the condition is satisfied and zero otherwise.

The following simple results are used throughout the paper:

Lemma 1 (Repetition coding over BSC)**.** *Let a bit be sent over BSC(ε) using ρ repetitions and decoded by a majority vote (if ρ is even, ties are broken by tossing a fair coin). The decoding error probability P_e can be upper bounded by*

$$P_e \leq \beta^\rho = 2^{-\rho\cdot d(\frac{1}{2}||\varepsilon)},$$

where $\beta \triangleq 2\sqrt{\varepsilon(1-\varepsilon)}$ is the Bhattacharyya parameter respective to the BSC(ε). The induced channel from the input bit to its decoded value is thus a BSC(P_e).

The proof is standard (see for example [8]) and can be regarded as special case of Lemma 8 stated and proved in Section IX. Note that the random tie breaking is done in order to simplify the scheme and its analysis. It does, however, assume private randomness at both parties. In Section X we show how the random tie breaking can be circumvented.

We now introduce two error detection methods that would be used in the coding scheme. The first one assumes the error are generated by BSC's and is based on error correction codes:

Definition 2 (Error detection using an extended-Hamming code)**.** *Let $\mathbf{X}^A$ and $\mathbf{X}^B$ be binary (row) vectors of length k held by Alice and Bob respectively. Let H be the parity check matrix of an extended-Hamming code with parameters $(k, k-\log k-1, 4)$. Let NEQ be a variable set to one if the parties decide that $\mathbf{X}^A \neq \mathbf{X}^B$ and set to zero otherwise, calculated according to the following algorithm:*

1) *Alice calculates her syndrome vector $\mathbf{s}^A = \mathbf{X}^A H^T$*
2) *Bob calculates his syndrome vector $\mathbf{s}^B = \mathbf{X}^B H^T$*
3) *Alice sends $\mathbf{s}^A$ ($1+\log k$ bits) to Bob*
4) *Bob calculates $NEQ = \mathbb{1}\left(\mathbf{s}^A \neq \mathbf{s}^B\right)$*
5) *Bob sends NEQ (1 bit) to Alice*

The overall number of bits communicated between Alice and Bob is $2+\log k$.

The performance of this scheme over a BSC(ε) is given in the following lemma:

Lemma 2. *Assume that*

$$\mathbf{X}^A = \mathbf{X}^B \oplus \mathbf{Z}.$$

where $\mathbf{Z}$ is an i.i.d Bernoulli(ε) vector. The probability of a mis-detected error of the scheme in Definition 2 is given by

$$\begin{aligned}&\Pr\left(NEQ=0, \mathbf{X}^A \neq \mathbf{X}^B\right) \\ &= \frac{1}{2k}\left(1+2(k-1)(1-2\varepsilon)^{\frac{k}{2}} + (1-2\varepsilon)^k\right) \\ &\quad - (1-\varepsilon)^k.\end{aligned} \tag{6}$$

The corresponding probability of a false error detection is

$$\Pr\left(NEQ=1, \mathbf{X}^A = \mathbf{X}^B\right) = 0.$$

Proof. First, it is clear that for any $\mathbf{X}^A = \mathbf{X}^B$ we have $NEQ = \mathbb{1}\left(\mathbf{s}^A \neq \mathbf{s}^B\right) = 0$ with probability one, so the probability of false error detection is $\Pr\left(NEQ = 1, \mathbf{X}^A = \mathbf{X}^B\right) = 0$. For the probability of error mis-detection, note that $\mathbf{s}^A \oplus \mathbf{s}^B = (\mathbf{X}^A \oplus \mathbf{X}^B)H^T = \mathbf{Z}H^T$. Therefore, the event $NEQ = 0$ is identical to the event in which $\mathbf{s}^A \oplus \mathbf{s}^B = \mathbf{Z}H^T = \mathbf{0}^T$, i.e., $\mathbf{Z}$ is a codeword in H. All in all

$$\begin{aligned}
&\Pr\left(NEQ = 0, \mathbf{X}^A \neq \mathbf{X}^B\right) \\
&= \Pr\left(\mathbf{Z}H^T = \mathbf{0}^T, \mathbf{Z} \neq \mathbf{0}^T\right) \\
&= \frac{1}{2k}\left(1 + 2(k-1)(1-2\varepsilon)^{\frac{k}{2}} + (1-2\varepsilon)^k\right) \\
&\quad - (1-\varepsilon)^k, \qquad (7)
\end{aligned}$$

where (7) is standardly calculated using the dual code [24, p. 52]. □

The second error detection scheme is a randomized scheme based on [14, p. 30], which applies for arbitrary vectors. We note that this scheme performs the error detection using hashing, where the hash functions are implemented using polynonmials.

Definition 3 (Randomized error detection using polynomials)**.** *Let $\mathbf{X}^A$ and $\mathbf{X}^B$ be arbitrary binary vectors of length ℓ held by Alice and Bob respectively. Let $\gamma \in \mathbb{N}$, where $\gamma > 1$. Let q be a prime number such that $\gamma\ell \leq q \leq 2\gamma\ell$ (by Bertrand's postulate such a number must exist). Let NEQ^{Poly} be a variable set to one if the parties decides that $\mathbf{X}^A \neq \mathbf{X}^B$ and set to zero otherwise, calculated according to the following algorithm:*

1) *Alice uniformly draws $U \in \mathbb{F}_q$, $U \neq 0$.*
2) *Alice calculates $A(U, \mathbf{X}^A) = \sum_{i=1}^{\ell} X_i^A U^{i-1} (mod\ q)$*
3) *Alice sends Bob U and $A(U, \mathbf{X}^A)$*
4) *Bob calculates $B(U, \mathbf{X}^A) = \sum_{i=1}^{\ell} X_i^B U^{i-1} (mod\ q)$*
5) *Bob calculates $NEQ^{Poly} = \mathbb{1}\left(A(U, \mathbf{X}^A) - B(U, \mathbf{X}^A) \neq 0\right)$*
6) *Bob sends NEQ^{Poly} to Alice*

All in all, Alice needs to send at most $\lceil \log 2\gamma\ell \rceil$ bits for the representation of U, and at most $\lceil \log 2\gamma\ell \rceil$ bits for the representation of $A(U, \mathbf{X}^A)$. Bob sends Alice one bit.

Lemma 3. *The error detection scheme of Definition 3 obtains an error mis-detection probability of*

$$\Pr\left(NEQ^{Poly} = 0 \mid \mathbf{X}^A \neq \mathbf{X}^B\right) \leq \frac{1}{\gamma},$$

and a false error detection probability of

$$\Pr\left(NEQ^{Poly} = 1 \mid \mathbf{X}^A = \mathbf{X}^B\right) = 0.$$

Proof. Note that $A(U, \mathbf{X}^A)$ and $B(U, \mathbf{X}^B)$ are the evaluation at point U of two polynomials over $\mathbb{F}_q$ whose (binary) coefficients are the elements of $\mathbf{X}^A$ and $\mathbf{X}^B$ respectively. Clearly, if $\mathbf{X}^A = \mathbf{X}^B$, then $NEQ = 0$ for every value of U hence $\Pr\left(NEQ = 1 \mid \mathbf{X}^A = \mathbf{X}^B\right) = 0$. On the other hand, if $\mathbf{X}^A \neq \mathbf{X}^B$, $A(U, \mathbf{X}^A) - B(U, \mathbf{X}^A) = 0$ implies that U is a root of the polynomial

$$\sum_{i=1}^{\ell} (X_i^A - X_i^B) U^{i-1} (\text{mod } q).$$

Since the degree of the polynomial is at most ℓ, there are at most $\ell - 1$ such roots, so

$$\Pr\left(NEQ = 0 \mid \mathbf{X}^A \neq \mathbf{X}^B\right) \leq \frac{\ell - 1}{q} < \frac{\ell}{\gamma\ell} = \frac{1}{\gamma}.$$

□

VI. The Lower Bound in the BSC Case

In the following sections we prove the lower bound on the essential interactive capacity to the BSC case, which is then extended to BMS in Section IX. The BSC version of the bound is stated in the following theorem:

Theorem 2. *For any BSC with crossover probability $0 \leq \varepsilon \leq 1/2$, Shannon capacity $C_{\mathrm{Sh}}(\varepsilon)$ the and essential interactive capacity $C_{\mathrm{I}}(\varepsilon)$ the following bound holds:*

$$\frac{C_{\mathrm{I}}(\varepsilon)}{C_{\mathrm{Sh}}(\varepsilon)} \geq 0.0302.$$

This bound is derived by using a rewind-if-error scheme for a small ε, whose rate appears in Theorem 3, and then leveraging it to a general BSC using repetition coding via Lemma 4.

Theorem 3. *The transcript of any protocol with n bit-vs.-bit order of speakers (i.e. Alice sends a bit*

on odd times and Bob sends a bit on even times), can be reliably simulated over BSC(ε) (i.e. with a vanishing error as $n \to \infty$ for a fixed ε) at the rate specified in (8)*, where*

$$P_{e1} \leq \frac{1}{2k}\left(1+2(k-1)(1-2\varepsilon)^{\frac{k}{2}}+(1-2\varepsilon)^{k}\right) - (1-\varepsilon)^{k}+(3+\log k)\beta^{5}. \quad (9)$$

Let $0 < \varepsilon < \frac{1}{16}$ and $\beta \triangleq 2\sqrt{\varepsilon(1-\varepsilon)}$. k can be any integer power of two satisfying $k \leq \frac{1}{8\varepsilon}$.

An example for $R_{BSC}(\varepsilon,k)$ with $k=9$ is depicted in Figure 1. Using this theorem, $C_{\mathrm{I}}(\varepsilon) \geq \max_k R_{BSC}(\varepsilon,k)$ for protocols with a bit-vs.-bit order of speakers and $C_{\mathrm{I}}(\varepsilon) \geq \frac{1}{2}\max_k R_{BSC}(\varepsilon,k)$ for protocols with a general (possibly adaptive) order of speakers.

The proof of Theorem 3 is by the construction and analysis of a rewind-if-error scheme and appears in Sections VII and VIII. We note that the presented scheme is randomized and in Section X we explain how to modify it to be deterministic. It is also in place to note that the error probability of this scheme decays polynomially in n, as can be seen in the analysis of the error event.

The following corollary proved in Appendix A states that the scheme obtains the rate lower bound $1-O(\sqrt{h(\varepsilon)}$ from [13]:

Corollary 1. *For $\varepsilon \to 0$*

$$\max_k R_{BSC}(\varepsilon,k) \geq 1-O(\sqrt{h(\varepsilon)})$$

As stated before, the presented rewind-if-error scheme is designed for BSC with a sufficiently small ε. For larger values of ε, the channel can be converted to a BSC(δ') with $\delta' \leq \delta < \varepsilon$ using $\rho(\varepsilon,\delta)$ repetitions followed by a majority vote according to Lemma 1. The following lemma bounds the essential interactive capacity by using an interactive coding scheme augmented by a repetition code:

Lemma 4. *For every $0<\varepsilon<\frac{1}{2}$ and $0<\delta<\frac{1}{2}$*

$$\frac{C_{\mathrm{I}}(\varepsilon)}{C_{\mathrm{Sh}}(\varepsilon)} \geq \frac{C_{\mathrm{I}}(\delta)}{\log\frac{1}{\delta}+1}.$$

Proof. Let ρ be the smallest integer such that $\beta^{\rho} \leq \delta$, where $\beta \triangleq 2\sqrt{\varepsilon(1-\varepsilon)}$ is the Bhattacharyya parameter of the BSC(ε) as above. By Lemma 1, using ρ repetitions, the BSC(ε) can be converted to a BSC(δ') with $\delta' \leq \delta$. Normalizing by $C_{\mathrm{Sh}}(\varepsilon)$ and noting that $C_{\mathrm{I}}(\delta) \leq C_{\mathrm{I}}(\delta')$ we obtain

$$\frac{C_{\mathrm{I}}(\varepsilon)}{C_{\mathrm{Sh}}(\varepsilon)} \geq \frac{C_{\mathrm{I}}(\delta)}{\rho(\varepsilon,\delta)C_{\mathrm{Sh}}(\varepsilon)}.$$

By the definition of ρ in Lemma 1:

$$\rho \leq \rho(\varepsilon,\delta) \triangleq \frac{\log\frac{1}{\delta}}{\log\frac{1}{\beta}}+1,$$

where '$+1$' accounts for rounding to the nearest larger integer. Furthermore,

$$\begin{aligned}\rho(\varepsilon,\delta)C_{\mathrm{Sh}}(\varepsilon) &= \left(\frac{\log\frac{1}{\delta}}{\log\frac{1}{\beta}}+1\right)C_{\mathrm{Sh}}(\varepsilon) \quad (10)\\ &\leq \frac{I(X;Y)}{L(X;Y)}\log\frac{1}{\delta}+I(X;Y),\end{aligned}$$

where $X \sim \text{Bernoulli}\left(\frac{1}{2}\right)$ is the input of a BSC(ε) channel and Y is its respective output,

$$I(X;Y) = D\left(P_{XY}||P_XP_Y\right) = C_{\mathrm{Sh}}(\varepsilon)$$

is the mutual information between X and Y and

$$L(X;Y) = D\left(P_XP_Y||P_{XY}\right) = d\left(\tfrac{1}{2}||\varepsilon\right) = \log\frac{1}{\beta}$$

is the *lautum information* between X and Y [25]. Using the facts that for the BSC, $L(X;Y) \geq I(X;Y)$ [25, Theorem 12] and that trivially $I(X;Y) \leq 1$, concludes the proof. □

Theorem 2 now follows by using $\frac{1}{2}R_{BSC}(\delta,k)$

$$R_{BSC}(\varepsilon,k) \triangleq \frac{1-k\varepsilon-(3+\log k)\beta^5-\frac{k^2}{k-1}\left(P_{e1}+3\beta^7 k\log k\frac{2-\beta^2 k}{(1-\beta^2 k)^2}\right)-3\beta^7\log k\frac{2-\beta^2}{(1-\beta^2)^2}}{1+\frac{5(3+\log k)}{k}+3\log k\left[\frac{3(2k-1)}{k(k-1)^2}+\frac{4k}{(k-1)^3}+\frac{4k-2}{k(k-1)^2}\right]} \quad (8)$$

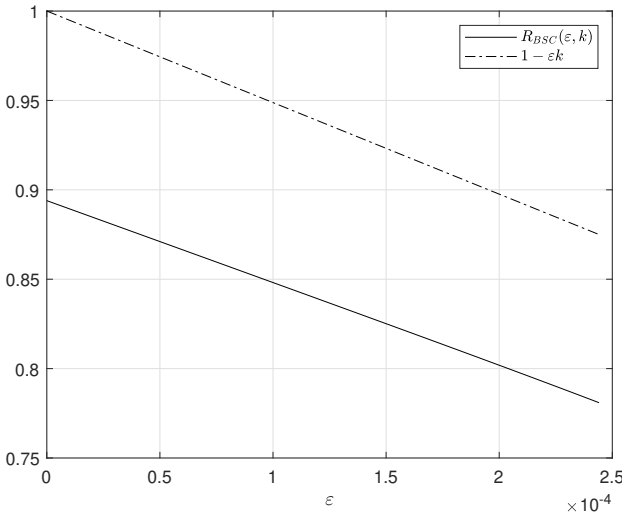

Fig. 1. An example for the rate $R_{BSC}(\varepsilon, k)$ from Theorem 3 with $k = 2^9$. The maximal channel crossover probability ε is $\frac{1}{8k}$ as required by the theorem. It is observable that in this regime, $R_{BSC}(\varepsilon, k)$ is almost linear in ε. This is since the denominator is constant and the dominant term in the numerator is $1 - k\varepsilon$. The function $1 - k\varepsilon$ is also plotted as reference.

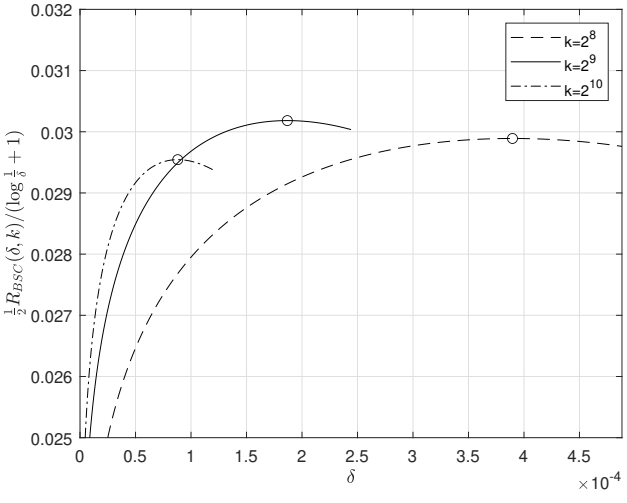

Fig. 2. Calculating the bound of Theorem 2 using (11) and various combinations of k and δ.

from Theorem 3 as a lower bound to $C_{\mathrm{I}}(\delta)$, where the $\frac{1}{2}$ factor is used in for the symmetrization of the order of speakers. Then, applying Lemma 4, gives:

$$\frac{C_{\mathrm{I}}(\varepsilon)}{C_{\mathrm{Sh}}(\varepsilon)} \geq \frac{\frac{1}{2}R_{BSC}(\delta,k)}{\log\frac{1}{\delta}+1}. \tag{11}$$

The bound in (11) is then tightened by scanning through various values of δ, k as seen in Figure 2. The combination of $k = 2^9$ and $\delta = 0.0001842$ gives the value of the lower bound in Theorem 2.

VII. Description of the coding scheme for the BSC

The *rewind-if-error* scheme is based on two concepts: uncoded transmission and retransmissions based on error detection. The uncoded transmission is motivated by the fact that in a general interactive protocol, even in a noise-free environment, the parties cannot predict the transcript bits to be output by their counterpart, and hence might not always know some of their own future outputs. For this reason, long blocks of bits, which are essential for efficient block codes, cannot be generated.

The concept of retransmissions based on error detection can be viewed as an extension of the classic example of the one-way BEC with feedback [8, p. 506]. In this simple setup, channel errors occur independently with probability ϵ and errors are detected and marked as erasures, whose locations are immediately revealed to both parties. The coding scheme is simply resending the erased bits, yielding an average rate of $1 - \epsilon$, which is exactly Shannon's capacity for the BEC. In addition, since all the channel errors are marked as erasures, the probability of decoding error is zero.

When performing interactive communication over a BSC, channel errors are not necessarily marked as erasures and perfect feedback is not present. However, the fact that the parties have (a noisy) two-way communication link, enables them to construct a coding scheme in a similar spirit as follows. The parties start by simulating the transcript in a window (or a *block*) of k consecutive bits, operating as if the channel is error free. The probability of error in the window can be upper bounded using the union bound by $k\varepsilon$, and this number is assumed to be small. Next, the parties exchange bits in order to decide if the window is

correct, i.e., no errors occurred, which would lead to the simulation of the consecutive window, or incorrect, i.e., some errors occurred, which would lead to retransmission (i.e. re-simulation of the window).

Unfortunately, error detection using less than k bits of communication has an inherent failure probability. In addition, performing the error detection over a noisy channel can cause further errors, including a disagreement between the parties regarding the mere presence of the errors. For this purpose, the error detection is done in a *hierarchical* and *layered* fashion. Namely, after k windows are simulated, error detection is applied on all of them, including on the outcome of the previous error detection, possibly initiating their entire retransmission. After k^2 windows are simulated, error detection is applied on all of them, and so on. An illustrated example for this concept for $k = 4$ is given in Table I.

We are now ready to describe the coding scheme. We note that it can be viewed both as a sequential algorithm and as a recursive algorithm. For sake of clarity and simplicity of exposition, we chose the sequential interpretation for the description and the recursive interpretation for the analysis.

A. Building blocks

In the sequel we assume that the order of speakers is bit. vs. bit, namely, Alice speaking at odd times and Bob speaking at even times. We denote the input of a the channel by X_i and its corresponding output by Y_i. We denote by i the time index used for the protocol simulation. $i \in \{0, T-1\}$, where T denotes the number of times the channels are used for the simulation of the protocol, excluding the overhead required for the calculation of the rewind bits. In other words, for the sake of simplicity, the instances in which the channels are used for error detection are counted and indexed separately.

The following notions are used as the building blocks of the scheme:

- The *uncoded simulation* of the transcript is a sequence of bits, generated by the parties and the channel, using the transmission functions in $\boldsymbol{\pi}$ and disregarding the channel errors. Alice's and Bob's uncoded simulation vectors are for odd i: $\mathbf{X}_i^A \triangleq (X_1, Y_2, \ldots, X_i)$, and $\mathbf{X}_i^B \triangleq (Y_1, X_2, \ldots, Y_i)$ respectively. For even i they are $\mathbf{X}_i^A \triangleq (X_1, Y_2, \ldots, Y_i)$, and $\mathbf{X}_i^B \triangleq (Y_1, X_2, \ldots, X_i)$ respectively.
- The *cursor* variables indicate the time indexes of the transmission functions (i.e. the appropriate function in $\boldsymbol{\pi}$) used by Alice or Bob in the previous transmission. We denote Alice's and Bob's cursors by j^A and j^B respectively. We note that j^A and j^B are random variables and may not be identical.
- The *rewind bits* are the result of the error detection procedure and are calculated at predetermined points throughout the scheme. They determine whether the simulation of the transcript should proceed forward, or rewind. We recall that T denotes the number of times the channels are used for the simulation of the protocol, excluding the overhead required for the calculation of the rewind bits. We define the number of *layers* by $L = \log_k T$, so that $T = k^L$. We then separate the rewind bits into layers : $l = 1, \ldots, L$. At layer l there are k^{L-l} rewind bits, denoted by $b_l^A(1), ..., b_l^A(k^{L-l})$ for Alice and $b_l^B(1), ..., b_l^B(k^{L-l})$ for Bob. The value of Alice's and Bob's rewind bits might differ in the general case. The rewind bits $b_l^A(m)$ and $b_l^B(m)$ are calculated after exactly mk^l bits of uncoded simulation, and are calculated according to their respective *rewind windows*. In the sequel we use the term *active* to denote that a rewind bit is set to one, and *inactive* if it is set to zero.
- The *rewind window* $w\left[b_l^A(m)\right]$ of Alice (resp. $w\left[b_l^B(m)\right]$ of Bob) contains the bits according to which $b_l^A(m)$ (resp. $b_l^A(m)$) is calculated. It contains the uncoded simulation bits of the respective party, between times $(m-1)k^l + 1$ and mk^l. In addition it contains all the rewind bits of levels $1 \leq \tilde{l} < l$ the party has calculated between these times.

We note, that at every point of the simulation, having the uncoded simulation bits and the rewind bits calculated so far, both parties can calculate

their cursors j^A and j^B and their simulations of the transcript. We denote these simulation vectors by: $\hat{\boldsymbol{\tau}}_A$ and $\hat{\boldsymbol{\tau}}_B$ for Alice and Bob respectively. We are now ready to introduce the coding scheme.

The coding scheme

The coding scheme is elaborated in Algorithm 1. Note that this is the scheme as implemented at Alice's. The coding scheme implemented at Bob's side is obtained by respectively replacing j^A, $\mathbf{X}^A$, $\hat{\boldsymbol{\tau}}_A$, $b_l^A(m)$, "if j^A is odd", $X_i = \phi_{j^A}^{\text{Alice}}(\hat{\boldsymbol{\tau}}_A^{j^A-1})$ by j^B, $\mathbf{X}^B$, $\hat{\boldsymbol{\tau}}_B$, $b_l^B(m)$, "if j^B is even", $X_i = \phi_{j^B}^{\text{Bob}}(\hat{\boldsymbol{\tau}}_B^{j^B-1})$.

Calculation of the rewind bits

For the first layer, $l = 1$, the rewind bits are calculated using the algorithm for error detection using an extended-Hamming code, described in Definition 2. The reason for the choice of this procedure is the fact that in the first layer the difference between $\mathbf{X}^A$ and $\mathbf{X}^B$ is only the channel noise, which is i.i.d. Bernoulli(ε), and the fact that the extended-Hamming code is a good error detection code for such a noise. In particular, this code is *proper* [24], which means that the probability of error mis-detection is monotonically increasing for $0 < \varepsilon < 1/2$. As the probability of mis-detection for $\varepsilon = \frac{1}{2}$ is equal to that of random hashing with the same number of bits, for $\varepsilon < \frac{1}{2}$ we obtain favorable performance without randomness. The details of the calculation are elaborated in Algorithm 2.

For all other layers, $l > 1$, the procedure is implemented according to the polynomial based randomized error detection scheme from Definition 3. We start by assuming that the parties agree on the prime number q_l for every layer $l > 1$. We also assume for simplicity of exposition, that for every rewind window, the parties commonly and independently draw a test point U using a common random string. We denote the set comprising all the test points used by the scheme by $\mathcal{U}$, which contains $|\mathcal{U}| = O(n)$ elements. In Section X we show how the common randomness assumption can be relaxed. The details of the calculation are elaborated in Algorithm 3.

Initialization
- $i = 0$ the channel-use index
- $j^A = 0$ the cursor variable
- $\mathbf{X}_0^A = \emptyset$ the uncoded simulation vector
- $\hat{\boldsymbol{\tau}}_A = \emptyset$ the transcript simulation vector

while $i \leq T$
- uncoded simulation of k bits:
- **for** $\ell = 0$ **to** k
  - $i = i + 1$
  - $j^A = j^A + 1$
  - **if** j^A *is odd*
    - $X_i = \phi_{j^A}^{\text{Alice}}(\hat{\boldsymbol{\tau}}_A^{j^A-1})$ produce a transcript bit
    - $\mathbf{X}_i^A = (\mathbf{X}_{i-1}^A, X_i)$
    - $\hat{\boldsymbol{\tau}}_A^{j^A} = (\hat{\boldsymbol{\tau}}_A^{j^A-1}, X_i)$
  - **else**
    - receive Y_i, the transcript bit produced by Bob
    - $\mathbf{X}_i^A = (\mathbf{X}_{i-1}^A, Y_i)$
    - $\hat{\boldsymbol{\tau}}_A^{j^A} = (\hat{\boldsymbol{\tau}}_A^{j^A-1}, Y_i)$
- check if a rewind window is full and operate accordingly:
- **for** $l = 1$ **to** L
  - **if** i *mod* $k^l = 0$
    - $m = i/k^l$
    - the rewind window $w\left[b_l^A(m)\right]$ is full
    - calculate the rewind bit $b_l^A(m)$ by Algorithm 2 if $l = 1$ or Algorithm 3 if $l > 1$
    - **if** $b_l^A(m) = 1$ *rewind*
      - rewind j^A to the value it had at the beginning of $w\left[b_l^A(m)\right]$
      - delete the values of $w\left[b_l^A(m)\right]$ from $\hat{\boldsymbol{\tau}}_A$
      - set the values corresponding to $w\left[b_l^A(m)\right]$ in $\mathbf{X}_i^A$ to zero
    - **else**
      - do nothing

Algorithm 1: The coding scheme as implemented by Alice.

Input:
$\mathbf{X}^A = w\left[b_1^A(m)\right]$ - Alice's rewind window, k bits row-vector
$\mathbf{X}^B = w\left[b_1^B(m)\right]$ - Bob's rewind window, k bits row-vector
H - the parity check matrix of a $(k, k-\log k-1, 4)$ extended-Hamming code
Output:
$b_1^A(m)$ - Alice's rewind bit
$b_1^B(m)$ - Bob's rewind bit
Algorithm
Alice: calculate $\mathbf{s}^A = \mathbf{X}^A \cdot H^T$
Alice: send $\mathbf{s}^A$ to Bob over the channel using $\tilde{a}$ repetitions per bit
Bob: decode $\hat{\mathbf{s}}^A$ using a majority vote per bit on the channel respective inputs
Bob: calculate $\mathbf{s}^B = \mathbf{X}^B \cdot H^T$
Bob: $b_1^B(m) = \mathbb{1}\left(\hat{\mathbf{s}}^A \neq \mathbf{s}^B\right)$
Bob: send $b_1^B(m)$ to Alice over the channel using $\tilde{a}$ repetitions per bit
Alice: Set $b_1^A(m)$ according to a majority vote per bit on the channel respective input

Algorithm 2: Calculating of the rewind bits at $l = 1$

Let us now bound the number of bits required for this procedure. First, we generously bound the number of bits in a rewind window of layer l, which contains all the uncoded simulation bits and the nested rewind bits of the previous layers, by $2k^l$. For layer l, the parties set q_l to be the first prime number between $2k^{2+l}$ and $4k^{2+l}$. Therefore, a number in $\mathbb{F}_{q_l}$ can be represented by no more than $2 + (2+l)\log k$ bits. All in all the procedure described above required $3 + (2+l)\log k$ bits for layer l. For simplicity of calculation, from this point on, we bound this number by

$$3 + (2+l)\log k < 3l\log k, \tag{12}$$

which applies for any $l \geq 2$ and $k \geq 4$.

VIII. Analysis of the coding scheme : A Proof of Theorem 3

We start by giving the following notation:

Input:
$\mathbf{X}^A = w\left[b_1^A(m)\right]$ - Alice's rewind window, k^l bits row-vector
$\mathbf{X}^B = w\left[b_1^B(m)\right]$ - Bob's rewind window, k^l bits row-vector
H - the parity check matrix of a $(k, k-\log k-1, 4)$ extended-Hamming code
Output:
$b_1^A(m)$ - Alice's rewind bit
$b_1^B(m)$ - Bob's rewind bit
Algorithm
Alice & Bob: uniformly draws $U \in \mathbb{F}_q$, $U \neq 0$.
Alice: calculates $A(U, \mathbf{X}^A) = \sum_{i=1}^{\ell} X_i^A U^{i-1} (\text{mod } q)$
Alice: send the bits representing $A(U, \mathbf{X}^A)$ over the channel to Bob using $a + 2l$ repetitions per bit
Bob: decode $\tilde{A}(U, w\left[b_l^A(m)\right])$ using a majority vote per bit on the channel respective inputs
Bob: calculate $B(U, \mathbf{X}^A) = \sum_{i=1}^{\ell} X_i^B U^{i-1} (\text{mod } q)$
Bob: $b_l^B(m) = \mathbb{1}\left(\tilde{A}(U, \mathbf{X}^A) \neq B(U, \mathbf{X}^B)\right)$
Bob: send $b_l^B(m)$ to Alice over the channel using $a + 2l$ repetitions per bit
Alice: set $b_l^A(m)$ according to a majority vote per bit on the channel respective input

Algorithm 3: Calculating of the rewind bits at $l > 1$

- T is the number of times the channels are used for the protocol simulation, including retransmissions and excluding the overhead required for the transmission of the rewind bits.
- $j \triangleq \min\{j^A, j^B\}$ is the minimum between Alice's and Bob's cursor at any moment
- $j(T), j^A(T), j^B(T)$ denote the respective values of j, j^A, j^B at the end of the simulation
- $\hat{\boldsymbol{\tau}}_A^{j(T)}$ and $\hat{\boldsymbol{\tau}}_B^{j(T)}$ denote the first $j(T)$ bits of Alice's and Bob's simulations of the transcript respectively, at the end of the simulation. We

Start the simulation: Initialize the cursors: $j^A = j^B = 0$

	$w[b_1(1)]$	$b_1(1)$
A	$0,0,1,1$	0
B	$0,0,1,1$	0

End of $w[b_1(1)]$: No errors, continue. $j^A = j^B = 4$

	$w[b_1(1)]$	$b_1(1)$	$w[b_1(2)]$	$b_1(2)$
A	$0,0,1,1$	0	$1,0,0,1$	0
B	$0,0,1,1$	0	$1,0,0,1$	0

End of $w[b_1(2)]$: No errors, continue. $j^A = j^B = 8$

	$w[b_1(1)]$	$b_1(1)$	$w[b_1(2)]$	$b_1(2)$	$w[b_1(3)]$	$b_1(3)$
A	$0,0,1,1$	0	$1,0,0,1$	0	$0,\mathbf{0},0,0$	1
B	$0,0,1,1$	0	$1,0,0,1$	0	$0,\mathbf{1},0,0$	1

End of $w[b_1(3)]$: An error occurred and was detected by both parties: $b_1^A(3) = b_1^B(3) = 1$
Both parties zero the rewind window and rewind the cursors to the value it had before the window started: $j^A = j^B = 8$

	$w[b_1(1)]$	$b_1(1)$	$w[b_1(2)]$	$b_1(2)$	$w[b_1(3)]$	$b_1(3)$	$w[b_1(4)]$	$b_1(4)$
A	$0,0,1,1$	0	$1,0,0,1$	0	$0,0,0,0$	1	$0,1,1,1$	$\mathbf{1}$
B	$0,0,1,1$	0	$1,0,0,1$	0	$0,0,0,0$	1	$0,1,1,1$	$\mathbf{0}$

End of $w[b_1(4)]$: There are no errors so Bob calculates $b_1^B(3) = 0$ and continues ($j^B = 12$).
However due to an error in communicating $b_1^B(3)$, Alice decodes $b_1^A(3) = 1$, zeros the window and rewinds the cursor ($j^A = 8$)

	$w[b_2(1)]$								
	$w[b_1(1)]$	$b_1(1)$	$w[b_1(2)]$	$b_1(2)$	$w[b_1(3)]$	$b_1(3)$	$w[b_1(4)]$	$b_1(4)$	$b_2(1)$
A	$0,0,1,1$	0	$0,0,0,0$	0	$0,0,0,0$	0	$0,\mathbf{0},\mathbf{0},\mathbf{0}$	$\mathbf{1}$	1
B	$0,0,1,1$	0	$1,0,0,1$	0	$0,0,0,0$	0	$0,\mathbf{1},\mathbf{1},\mathbf{1}$	$\mathbf{0}$	1

End of $w[b_2(1)]$. Calculate $b_2(1)$.
The errors are detected so $b_2^A(1) = b_2^B(1) = 1$, and the cursors are rewound to the beginning of the window : $j^A = j^B = 0$.

	$w[b_2(1)]$										
	$w[b_1(1)]$	$b_1(1)$	$w[b_1(2)]$	$b_1(2)$	$w[b_1(3)]$	$b_1(3)$	$w[b_1(4)]$	$b_1(4)$	$b_2(1)$	$w[b_1(5)]$	$b_1(5)$
A	$0,0,0,0$	0	$0,0,0,0$	0	$0,0,0,0$	0	$0,0,0,0$	0	1	$0,0,1,1$	0
B	$0,0,0,0$	0	$0,0,0,0$	0	$0,0,0,0$	0	$0,0,0,0$	0	1	$0,0,1,1$	0

End of $w[b_1(5)]$ The first four bits of the protocol are re-simulated. No errors. $j^A = j^B = 4$.

TABLE I

EXAMPLE FOR A REWIND-IF-ERROR CODING SCHEME WITH $k = 4$. DETECTED ERROR ARE IN ***bold***, ZEROED BITS ARE IN *blue*.

also assume that if $j^A(T) > n$ or $j^B(T) > n$ then the parties proceed the protocol by transmitting zeros
- We denote $b_l(m) \triangleq b_l^A(m) \vee b_l^B(m)$. Namely, $b_l(m)$ it is defined as the disjunction between Alice's and Bob's respective rewind bits

The following two error events will be analyzed

- $\mathcal{E}_1$ is the event in which $j(T) < n$
- $\mathcal{E}_2$ is the event in which either $\hat{\boldsymbol{\tau}}_A^{j(T)} \neq \boldsymbol{\tau}^{j(T)}$ or $\hat{\boldsymbol{\tau}}_B^{j(T)} \neq \boldsymbol{\tau}^{j(T)}$

The simulation error event is included in $\mathcal{E}_1 \cup \mathcal{E}_2$ and we would like its respective probability to vanish with n.

We start by analyzing $\Pr(\mathcal{E}_1)$ and do it by lower bounding $j(T)$. We recall that by construction of the scheme, $b_l^A(m) = 1$ (resp. $b_l^B(m) = 1$) will rewind j^A (resp. j^B) to the value it had at the beginning of the rewind window. Namely j^A (resp. j^B) will be reduced by at most k^l. It is now instrumental to use the definitions of j and $b_l(m)$ and observe that if either $b_l^A(m) = 1$ or $b_l^B(m) = 1$ (namely, if $b_l(m) = 1$) then the minimal among j^A and j^B (namely, j) will be reduced by at most k^l. Recalling that $T = k^L$ we can now write

$$\begin{aligned} j(T) &\geq T - \sum_{l=1}^{L} \sum_{m=1}^{k^{L-l}} b_l(m) k^l \\ &= T \left(1 - \sum_{l=1}^{L} \overline{b_l} \right), \end{aligned} \tag{13}$$

where

$$\overline{b_l} \triangleq \frac{\sum_{m=1}^{k^{L-l}} b_l(m)}{k^{L-l}} \tag{14}$$

denotes the average number of active (i.e., nonzero) rewind bits at level l. We note that by construction of the scheme (including its use of randomness), the processes of the error generation and detection are identical for all blocks at level l. For this reason, the probability of having an active rewind bit is also identical for all the blocks at level l. We denote this probability by

$$P_{b_l} = \Pr(b_l(1) = 1) = ... = \Pr(b_l(k^{L-l}) = 1).$$

Taking the expectation over (13) yields

$$\mathbb{E}j(T) \geq T \left(1 - \sum_{l=1}^{L} P_{b_l} \right).$$

In order to proceed with the calculation of P_{b_l}, we define P_{el} as the probability that either $b_l^A(m)$ or $b_l^B(m)$ differ from the error indicator $\mathbb{1}\left(w\left[b_l^A(m)\right] \neq w\left[b_l^A(m)\right]\right)$. This probability does not depend on m due to the same considerations as above.

The following lemma bounds P_{el}:

Lemma 5. *For* $l = 1$

$$\begin{aligned} P_{e1} \leq \frac{1}{2k} &\left(1 + 2(k-1)(1-2\varepsilon)^{\frac{k}{2}} + (1-2\varepsilon)^k\right) \\ &- (1-\varepsilon)^k + (3 + \log k)\beta^{\tilde{a}}, \end{aligned}$$

and for $l > 1$

$$\begin{aligned} P_{el} \leq \\ k^{-l} \left(k P_{e1} + 3\beta^{a+4} k^2 \log k \frac{2 - \beta^2 k}{(1-\beta^2 k)^2} \right). \end{aligned} \tag{15}$$

Proof. For the first layer

$$\begin{aligned} P_{e1} \leq \\ \Pr\left(NEQ = 0, \mathbf{X}^A \neq \mathbf{X}^B\right) + (3 + \log k)\beta^{\tilde{a}}, \end{aligned} \tag{16}$$

where $\Pr\left(NEQ = 0, \mathbf{X}^A \neq \mathbf{X}^B\right)$ is the error misdetection probability of the extended-Hamming code based error detection scheme of Definition 2 as given in (6). $\beta^{\tilde{a}}$ is the probability of error in the decoding of a bit sent with $\tilde{a}$ repetitions according to Lemma 1. The multiplication by $(3 + \log k)$ accounts for the union bound over the number of bits used for the error detection: $2 + \log k$ bits sent from Alice to Bob ($1 + \log k$ required for the description of the syndrome according to Lemma 2 and an additional bit reserved for avoiding the random tie breaking as described in Subsection X-C) and a single bit fed back from Bob to Alice.

The key idea in the analysis of the scheme for $l > 1$ is regarding the calculation of the rewind bits as a *layered* recursive process. Namely, we observe that by construction, a rewind window at level l comprises k rewind windows of level

$l-1$. In addition, the polynomial based randomized error detection of Definition 3 uses independent test points for every layer and hence is independent between layers. Having this notion we can write the following recursion formula:

$$P_{el} \leq k^{-2} k P_{el-1} + (2 + (2+l)\log k)\beta^{a+2l} \quad (17)$$

where kP_{el-1} is the union bound over th error events of the previous level. The multiplication by k^{-2} accounts for the probability of error misdetection according to Lemma 3 with the setting $\gamma = k^{-2}$ and ℓ as the number of bits in the appropriate rewind window $w\left[b_l^A(m)\right]$ (or $w\left[b_l^B(m)\right]$). As described above, for the error detection, Alice should send Bob a number in $\mathbb{F}_q$ and Bob should reply with a single bit (we assume that the set of test points $\mathcal{U}$ is jointly drawn by the parties using common randomness). We recall that the number of bits required for the error detection scheme of Definition 3 is generously bounded by $3l\log k$ due to (12). All in all, we can rewrite (17) as

$$P_{el} \leq k^{-1} P_{el-1} + 3\beta^a (\log k) l \beta^{2l} \quad (18)$$

Solving the recursion of (18) with the initial condition in (16) we can bound P_{el} as follows:

$$\begin{aligned} P_{el} &\leq k^{1-l} P_{e1} + 3\beta^a (\log k) \sum_{j=2}^{l} j \beta^{2j} k^{j-l} \\ &\leq k^{1-l} P_{e1} + 3\beta^a (\log k) k^{-l} \sum_{j=2}^{\infty} j (\beta^2 k)^j \quad (19) \\ &= k^{1-l} P_{e1} + 3\beta^a (\log k) k^{-l} (\beta^2 k)^2 \frac{2-\beta^2 k}{(1-\beta^2 k)^2} \\ &= k^{-l} \left(k P_{e1} + 3\beta^{a+4} k^2 \log k \frac{2-\beta^2 k}{(1-\beta^2 k)^2} \right). \end{aligned}$$

We note that the assumption in Theorem 3 that $\varepsilon < 1/(8k)$ ensures that $\beta^2 k < 1$ ensuring that the infinite sum in (19) converges. □

We are now ready to bound P_{b_l}. We recall that it is defined as the probability that either $b_l^A(m) = 1$ or $b_l^B(m) = 1$, and is independent of m due to the symmetry of the scheme. For $l = 1$ we use the union bound over the probability of an erroneous bit and a communication error:

$$P_{b_1} \leq k\varepsilon + (3 + \log k)\beta^{\tilde{a}} \triangleq \overline{P}_{b_1}.$$

Similarly, for $l > 1$ we take the union bound over the probability of error P_{el-1}, in one of the k blocks in the layer $l-1$ and a communication error:

$$\begin{aligned} P_{b_l} &\leq k P_{el-1} + 3\beta^a (\log k) l \beta^{2l} \\ &\leq k^{2-l} \left(k P_{e1} + 3\beta^{a+4} k^2 \log k \frac{2-\beta^2 k}{(1-\beta^2 k)^2} \right) \\ &\quad + 3\beta^a (\log k) l \beta^{2l} \\ &\triangleq \overline{P}_{b_l}. \end{aligned} \quad (20)$$

Let us now bound the average rewind by

$$\mathbb{E}j(T) \geq T\left(1 - \sum_{l=1}^{\infty} \overline{P}_{b_l}\right) = T\zeta. \quad (21)$$

where

$$\begin{aligned} \zeta &\triangleq 1 - \sum_{l=1}^{\infty} \overline{P}_{b_l} \\ &= 1 - k\varepsilon - (3 + \log k)\beta^{\tilde{a}} \\ &\quad - \frac{k^2}{k-1} \left(P_{e1} + 3\beta^{a+4} k \log k \frac{2-\beta^2 k}{(1-\beta^2 k)^2} \right) \\ &\quad - 3\beta^{a+4} \log k \frac{2-\beta^2}{(1-\beta^2)^2}. \end{aligned}$$

Setting

$$T = \frac{n}{1 - \sum_{l=1}^{\infty} \overline{P}_{b_l} - \xi} = \frac{n}{\zeta - \xi} \quad (22)$$

for some $0 < \xi < \zeta$ will therefore ensure that $\mathbb{E}j(T) \geq n$. The following lemma ensures that $\Pr(\mathcal{E}_1)$ also vanishes in n:

Lemma 6. *For any $\xi > 0$ and T that satisfies* (22)*:*

$$\lim_{n\to\infty} \Pr(\mathcal{E}_1) = \lim_{n\to\infty} \Pr\left(j(T) < n\right) = 0.$$

The proof is in Appendix B. It is based on the fact that due to (21) and (22) we have $\mathbb{E}j(T) \geq (1+\eta)n$ for some $\eta > 0$ and using standard concentration techniques. We note that the proof assumes the number of test points in $\mathcal{U}$ is $|\mathcal{U}| = O(\sqrt{n})$, whereas so far we assumed that every use of the error detection procedure of Definition 3 uses a different test point (i.e. $|\mathcal{U}| = O(n)$).

Since $|\mathcal{U}| = O(\sqrt{n})$ is restrictive, Lemma 6 also holds for the current description of the scheme. The motivation for reducing $|\mathcal{U}|$ is changing the common randomness to private randomness, which is extracted from the channel, and is elaborated in Section X.

The following lemma ensures $\Pr(\mathcal{E}_2)$ vanishes in n:

Lemma 7. *For any $\xi > 0$ and T that satisfies* (22)

$$\lim_{n\to\infty} \Pr(\mathcal{E}_2) = 0.$$

Proof. We remind the reader that P_{el} is defined as the probability that either $b_l^A(m)$ or $b_l^B(m)$ differ from the error indicator $\mathbb{1}\left(w\left[b_l^A(m)\right] \neq w\left[b_l^A(m)\right]\right)$. Namely, it is the probability of an undetected error, or a falsely detected error, in the simulation of a block in layer l at least at one party. Since L is the final layer, and due to the recursive structure of the error detection, P_{eL} therefore upper bounds the respective probability at the end of the coding scheme. The error event related to P_{eL} includes $\mathcal{E}_2$ and therefore $\Pr(\mathcal{E}_2) \leq P_{eL}$. Rewriting (15) and setting $l = L = \log_k T = \log_k(n/(\zeta - \xi))$ we obtain:

$$\begin{aligned}&\Pr(\mathcal{E}_2)\\ &\leq \frac{\zeta-\xi}{n}\left(kP_{e1} + 3\beta^{a+4}k^2\log k\frac{2-\beta^2 k}{(1-\beta^2 k)^2}\right).\end{aligned}$$

Therefore $\lim_{n\to\infty}\Pr(\mathcal{E}_2) = 0$. □

Let us now bound N, the total number of channel uses consumed by the scheme:

$$\begin{aligned}N &\leq T + \tilde{a}(3+\log k)k^{L-1}\\ &+ 3\log k\sum_{l=2}^{\infty} l(a+2l)k^{L-l}\\ &\leq T\left(1 + \tfrac{\tilde{a}(3+\log k)}{k}\right.\\ &\left.+ 3\log k\left[\tfrac{a(2k-1)}{k(k-1)^2} + \tfrac{4k}{(k-1)^3} + \tfrac{4k-2}{k(k-1)^2}\right]\right), \quad (23)\end{aligned}$$

where $\tilde{a}(3+\log k)k^{L-1}$ is the number of channel uses required for the error detection at the first layer, and $3\log k\sum_{l=2}^{\infty} l(a+2l)k^{L-l}$ is the number of channel uses required for the error detection in all other layers. Using (22) and (23) we can bound the total rate of the scheme by the term in (24). Since this term holds for and $\xi > 0$, we can take the limit $\xi \to 0$. Setting $a = 3$ and $\tilde{a} = 5$ (which are the results of an exhaustive search over various possible values) provides (8) and conclude the proof of Theorem 3.

IX. Generalization to binary memoryless symmetric channels

In Definition 1 we defined a *binary memoryless symmetric* (BMS) channel as a collection of BSC's with various crossover probabilities.

We now extend the notion of repetition coding of Lemma 1 to BMS channels.

Definition 4. *[ρ-repetition channel] Let $P^{(\rho)}_{\tilde{Y}|\tilde{X}}$ be the ρ-repetition channel corresponding to a BMS($P_{Y|X}$) channel. It is obtained by using the bit $\tilde{X}$ as the input of BMS($P_{Y|X}$), ρ consecutive times, hence producing the series of channel outputs $Y_1, ..., Y_\rho$. The output of $P^{(\rho)}_{\tilde{Y}|\tilde{X}}$ is then calculated using the following equation*

$$\tilde{Y} = \underset{x\in\{0,1\}}{\operatorname{argmax}} \prod_{i=1}^{\rho} P_{Y|X=x}(Y_i), \qquad (25)$$

where ties are broken by drawing a $\mathrm{Bernoulli}(1/2)$ *random variable* [2].

We note that like in the BSC case, we randomly break the ties in order to facilitate the analysis and later explain in Subsection X-C how this random procedure can be circumvented. The following lemma bounds the decoding error of the ρ-repetition channel.

Lemma 8. *For any BMS($P_{Y|X}$) channel with Shannon capacity $C_{\mathrm{Sh}}(P_{Y|X}) = C$ the corresponding ρ-repetition channel $P^{(\rho)}_{\tilde{Y}|\tilde{X}}$ is a* $\mathrm{BSC}(\delta)$ *with $\delta \leq \beta^\rho$, where $\beta = 2\sqrt{h^{-1}(1-C)\cdot(1-h^{-1}(1-C))}$ is the Bhattacharyya parameter of a BSC(ε) with capacity C.*

Proof. We start by defining the log-likelihood ratio:

$$\Lambda \triangleq \ln\left[\frac{\prod_{i=1}^{\rho} P_{Y_i|X=0}(Y_i)}{\prod_{i=1}^{\rho} P_{Y_i|X=1}(Y_i)}\right],$$

[2]If Y is continuous, replace $P_{Y|X=x}$ with the conditional density.

and use Λ to rewrite the maximum-likelihood decision rule of (25) as:

$$\tilde{Y} = \begin{cases} 0 & \text{if } \Lambda > 0 \\ 1 & \text{if } \Lambda < 0 \\ W & \text{if } \Lambda = 0, \end{cases} \tag{26}$$

where W is a $\mathrm{Bernoulli}(1/2)$ random variable, drawn independently between uses of $P^{(\rho)}_{\tilde{Y}|\tilde{X}}$. Using the sufficient statistic $g(Y_i) = (X \oplus Z_{Ti}, T_i)$ from Definition 1, it is easy to show that the log-likelihood function Λ can be written as

The (symmetric) decision error probability can now be upper bounded by

$$\begin{aligned}
&\delta = \Pr(\tilde{Y} \neq \tilde{X}) \\
&= \Pr(\tilde{Y} \neq \tilde{X} \mid \tilde{X} = 0) \\
&\leq \Pr\left((-1)^{\tilde{X}} \sum_{i=1}^{\rho} (1 - 2Z_{Ti}) \ln \tfrac{1-T_i}{T_i} \leq 0 \;\middle|\; \tilde{X} = 0 \right) \qquad (27) \\
&= \Pr\left(\sum_{i=1}^{\rho} (1 - 2Z_{Ti}) \ln \tfrac{1-T_i}{T_i} \leq 0 \right). \qquad (28)
\end{aligned}$$

We note that the inequality in (27) implies that the event of a *tie* (i.e., $\Lambda = 0$) is regarded as an error in probability one, where in fact, due to the random tie breaking, it is an error with probability half. We now recall the Chernoff bound for a sum of i.i.d. random variables $A_1, ..., A_\rho$:

$$\Pr\left(\sum_{i=1}^{\rho} A_i \leq a \right) \leq e^{sa} \left[\mathbb{E} e^{-sA_i} \right]^{\rho}.$$

for any $s > 0$. Applying this bound to (28) with $A_i = (1 - 2Z_{Ti}) \ln \frac{1-T_i}{T_i}$, $a = 0$ and $s = 1/2$ yields

$$\delta \leq \boldsymbol{\beta}^{\rho} \tag{29}$$

where $\boldsymbol{\beta}$ is defined as the Bhattacharyya parameter of the channel $P_{Y|X}$, which is equal to:

$$\begin{aligned}
\boldsymbol{\beta} &= \mathbb{E}_{T,Z_T} \left(\left(\sqrt{\tfrac{T}{1-T}} \right)^{1-2Z_T} \right) \\
&= \mathbb{E}_T \left(\mathbb{E}_{Z_T|T} \left(\left(\sqrt{\tfrac{T}{1-T}} \right)^{1-2Z_T} \;\middle|\; T \right) \right) \\
&= \mathbb{E} \left(2\sqrt{T(1-T)} \right).
\end{aligned}$$

It was shown by Guillén i Fàbregas et. al. [11] that among all BMS channels $P_{Y|X}$ with capacity C, the Bhattacharyya parameter is maximized by a BSC. Their proof is based on the fact that the function $x \mapsto \sqrt{h^{-1}(x) \cdot (1 - h^{-1}(x))}$ is concave, and therefore:

$$\begin{aligned}
\boldsymbol{\beta} &= \mathbb{E}[2\sqrt{T(1-T)}] \\
&= 2\mathbb{E}\left[\sqrt{h^{-1}(h(T)) \cdot (1 - h^{-1}(h(T)))} \right] \\
&\leq 2\sqrt{h^{-1}(\mathbb{E}\left[h(T)\right]) \cdot (1 - h^{-1}(\mathbb{E}\left[h(T)\right]))} \qquad (30) \\
&= 2\sqrt{h^{-1}(1-C) \cdot (1 - h^{-1}(1-C))} \qquad (31) \\
&= 2\sqrt{\varepsilon \cdot (1-\varepsilon)} \qquad (32) \\
&= \beta
\end{aligned}$$

where in (30) we used Jensen's inequality, in (31) we used the the fact that capacity of a BMS channel is $C = 1 - \mathbb{E}[h(T)]$, and in (32) we used the capacity of the BSC(ε) $C = 1 - h(\varepsilon)$. Combining (29) and (32) concludes the proof of the lemma. □

We are now ready to prove Theorem 1, which is a generalization of Theorem 2 to BMS channels.

Proof of Theorem 1. We follow the same lines as the in proof of Lemma 4 and start by converting the BMS($P_{Y|X}$) channel to a BSC(δ') with $0 < \delta' \leq \delta$.

$$R_{BSC}(\varepsilon, k) \geq \frac{1 - k\varepsilon - (3 + \log k)\beta^{\tilde{a}} - \frac{k^2}{k-1}\left(P_{e1} + 3\beta^{a+4} k \log k \frac{2-\beta^2 k}{(1-\beta^2 k)^2} \right) - 3\beta^{a+4} k^2 \log k \frac{2-\beta^2 k}{(1-\beta^2 k)^2} - \xi}{1 + \frac{\tilde{a}(3+\log k)}{k} + 3 \log k \left[\frac{a(2k-1)}{k(k-1)^2} + \frac{4k}{(k-1)^3} + \frac{4k-2}{k(k-1)^2} \right]}. \tag{24}$$

According to Lemma 8 this can be done using

$$\rho(P_{Y|X},\delta) \triangleq \frac{\log\frac{1}{\delta}}{\log\frac{1}{\beta}} + 1.$$

repetitions where $\beta = 2\sqrt{\varepsilon(1-\varepsilon)}$ is the Bhattacharyya parameter of a BSC(ε) with capacity $C_{\mathrm{Sh}}(\varepsilon) = C_{\mathrm{Sh}}(P_{Y|X})$. We then apply an interactive coding scheme for the BSC(δ) with rate $R(\delta)$. After normalizing by $C_{\mathrm{I}}(P_{Y|X})$ the following bound it obtained:

$$\frac{C_{\mathrm{I}}(P_{Y|X})}{C_{\mathrm{Sh}}(P_{Y|X})} \geq \frac{R(\delta)}{\rho(P_{Y|X},\delta)C_{\mathrm{Sh}}(P_{Y|X})}. \quad (33)$$

Bounding the denominator of the right hand term in (33):

$$\rho(P_{Y|X},\delta)C_{\mathrm{Sh}}(P_{Y|X}) = \left(\frac{\log\frac{1}{\delta}}{\log\frac{1}{\beta}} + 1\right) C_{\mathrm{Sh}}(P_{Y|X})$$

$$= \left(\frac{\log\frac{1}{\delta}}{\log\frac{1}{\beta}} + 1\right) C_{\mathrm{Sh}}(\varepsilon),$$

which is exactly (10). The rest of the proof is as in Lemma 4, and using the same coding scheme to obtain the same numeric value in the lower bound as in Theorem 2. □

For completeness, we now show that not only $\frac{C_{\mathrm{I}}(P_{Y|X})}{C_{\mathrm{Sh}}(P_{Y|X})} \geq 0.0302$ for any BMS channel, but also the ratio $\frac{C_{\mathrm{I}}(P_{Y|X})}{C_{\mathrm{Sh}}(P_{Y|X})}$ tends to one as the BMS channel becomes cleaner, similarly to the BSC case.

Corollary 2. *For any sequence in C of BMS channels $P^C_{Y|X}$ with $C_{\mathrm{Sh}}(P^C_{Y|X}) = C$, we have*

$$\lim_{C\to 1} \frac{C_{\mathrm{I}}(P_{Y|X})}{C} = 1.$$

Proof. We start by proving that without repetitions a BMS($P_{Y|X}$) channel can be reduced to BSC(ε) with $\varepsilon \leq \frac{1-C_{\mathrm{Sh}}(P_{Y|X})}{2}$. As in [26], the proof is by noting that $h(t) \geq 2t$ for any $t \in [0, 1/2]$ and therefore:

$$\begin{aligned} C_{\mathrm{Sh}}(P_{Y|X}) &= 1 - \mathbb{E}h(T) \\ &\leq 1 - \mathbb{E}2T \\ &= 1 - 2\varepsilon. \end{aligned}$$

The corollary now follows by taking the lower bound for $R_{BSC}(\varepsilon, k)$ in Corollary 1 as a lower bound to $C_{\mathrm{I}}(P_{Y|X})$. □

X. A deterministic coding scheme

The coding scheme described throughout this paper uses randomness for two purposes: the randomized polynomial based error detection procedure described in Definition 3, and the random tie breaking in the repetition decoding described in Lemma 1 and Lemma 8. In this section we show how the requirements for randomness can be relaxed using a few simple adaptations of the coding scheme.

A. On the Randomness Requirements of the Error Detection Scheme in Definition 3

We start by recalling that the scheme from Definition 3 requires a random generation of a test point U taken from a finite field. We note that original scheme from [14, p. 30] requires only private randomness. Namely, the test point U should be drawn by Alice party and conveyed to Bob. However, so far we assumed that all the test points used by the scheme (denoted by $\mathcal{U}$) are jointly drawn by both parties using a shared random string (i.e., *public randomness*). This choice was made in order to save the communication overhead of conveying the test points from one party to the other, which is prone to reduce the overall rate of the interactive communication scheme.

The first step in modifying the communication scheme to private randomness is showing the number of random test points can be reduced, without affecting the overall rate. We start showing that $|\mathcal{U}|$, the number of random test points required for all the error detections in the interactive coding scheme can be reduced to $o(n)$. This way, if only private randomness is used, $\mathcal{U}$ can be reliably conveyed from one party to the other without affecting the total rate. In Subsection X-B we show how $\mathcal{U}$ can be generated using randomness extracted from the channel, removing the requirement for private randomness.

We start by noting that by construction of error detection scheme, using independently drawn test

points for its different actuations, will make their corresponding error mis-detection events statistically independent. It is now in place to discuss the amount of statistical independence required by the coding scheme. In (17) we assumed that the probability of error mis-detection is independent between layers. That might imply that using $|\mathcal{U}| = L$ is satisfactory. In fact, if one is concerned only with the average rate of the coding scheme, using only $|\mathcal{U}| = L$ will lead to the same average rate of Theorem 3.

However, we recall that we defined rate not in the average sense, but rather, we required the reconstruction of the transcript with high probability after a predetermined simulation length. To illustrate this delicate difference, consider the example of the one-way BEC with feedback. In this example, all the erased bits are retransmitted. So, using the channel n times will result in $n(1-\epsilon)$ bits decoded with zero error, where ϵ is the erasure probability. This means that the average rate is $1-\epsilon$, which is exactly the Shannon capacity of the BEC(ϵ). However, it is interesting to note that since the erasures are drawn i.i.d., for $n \to \infty$ the rate will concentrate around its average and the probability of decoding less than $n(1-\epsilon-\xi)$ bits will vanish in n for any $\xi > 0$. This means, that this simple scheme also achieves Shannon's capacity in a stricter deterministic sense - namely, for $n \to \infty$ a number of information bits respective to Shannon's capacity could be reliably transmitted with a vanishing error probability using a fixed number of channel uses.

For our scheme, the convergence to the average rate is stated in Lemma 6. The concept of the proof appearing in Appendix B is similar to that of the BEC with feedback. We regard the rewind bits as the counterparts of the erasures in the BEC and show that actual number of rewind bits in every layer, concentrates around its average. A delicate issue in the analysis is the independence of the rewind bits in our scheme. In the first layer, the rewind bits are calculated according to Definition 2. This is a deterministic scheme that is based only on the vectors of channel errors, which are i.i.d between different blocks. Therefore, the rewind bits are indeed i.i.d. For higher layers, the scheme in Definition 3 is used. As explained in the proof of Lemma 3, the rewind bit is calculating according to

$$\mathbb{1}\left(\sum_{i=1}^{\ell}(X_i^A - X_i^B)U^{i-1}(\text{mod } q) \neq 0\right).$$

While it is tempting to assume that $X_i^A - X_i^B$ is exactly the vector of i.i.d channel errors, we note that the "$-$" operation is done over $\mathbb{F}_q$ and not over $\mathbb{F}_2$. This means, that the event of error mis-detection depends not only on the channel error vector, but also on the vectors related to the transcript: X_i^A, X_i^B. Since the transcript might be dependent between consecutive blocks, the corresponding rewind bits might also be statistically dependent, if the same value of U is used for both blocks.

One way of breaking this dependence is drawing independent U for every error detection in every layer. As stated before, if common randomness is used, this procedure is feasible, but when using only private randomness it might cause a decrease of the total rate. We recall that in every layer $1 < l \leq L$, there are k^{L-l} blocks for which error detection is applied using Definition 3. In our modification of the coding scheme for private randomness we assume that only $k^{\lceil (L-l)/2 \rceil}$ independent test points are used, such that the test point is changed every $k^{\lfloor (L-l)/2 \rfloor}$ blocks. In Appendix B we prove that this reduced number of independent test points still ensures a slower, yet fast enough, concentration.

Let us now bound the total number of bits required for the description of $\mathcal{U}$ denoted by n_U. We recall that the number of bits required for the error detection at layer $1 < l \leq L$ is bounded by $3l \log k$ by (12). So, the overall number of bits can be upper bounded by

$$\begin{aligned} n_U &\leq \sum_{l=2}^{L} 3l(\log k)k^{\lceil (L-l)/2 \rceil} \\ &\leq 3k^{L/2+1} \log k \sum_{l=2}^{\infty} lk^{-l/2} \\ &= O(k^{L/2} \log k) \\ &= O(\sqrt{n}) \end{aligned}$$

These bits can be conveyed from Alice to Bob

before the beginning of the simulation using a block code with some constant positive rate R_U below Shannon's capacity, requiring $\frac{n_U}{R_U} = O(\sqrt{n})$ channel uses. However, an error in the decoding of $\mathcal{U}$ might occur, which might cause a failure in the simulation of the entire transcript. We denote this error event by $\mathcal{E}_3$ and add it to the previously defined error events $\mathcal{E}_1$ and $\mathcal{E}_3$. The probability of $\mathcal{E}_3$ can be upper bounded by an error exponent yielding:

$$\Pr(\mathcal{E}_3) \leq e^{-O(\sqrt{n})}$$

so clearly $\lim_{n\to\infty} \Pr(\mathcal{E}_3) = 0$ making this error event negligible. We should also add $\frac{n_U}{R_U}$ to the total number of channel uses of the scheme in (23). But since $\frac{n_U}{R_U} = O(\sqrt{n})$, N would change only by $O(\sqrt{n})$, which would not affect the asymptotic value of the rate from Theorem 3.

B. Extracting randomness from the channel

In the previous subsection we showed that the error detection procedure of Definition 3 can be implemented using private randomness requiring $n_U \leq O(\sqrt{n})$ random bits for the entire coding scheme, which were assumed to be drawn by Alice. Our coding scheme can however, be made explicit by extracting the random bits from the channel. While a randomness extraction procedure with optimal efficiency was presented by Elias in [4], we use von-Neumann's suboptimal scheme [5] due to its simplicity of analysis and the vanishing effect of its suboptimality on the total rate.

Lemma 9. *The coding scheme can be made explicit by extracting the randomness from the channel with an overhead of*

$$n_R = O(\sqrt{n})$$

channel uses and an additional error probability

$$\Pr(\mathcal{E}_4) \leq e^{-O(\sqrt{n})}.$$

Proof. Bob sends Alice n_R zeros and Alice receives a noise vector $Z_1, ..., Z_{n_R}$ whose elements are i.i.d Bernoulli(ε). Alice then divides the noise elements into pairs. For the pairs 00 and 11, Alice does nothing. For the pairs 01 or 10 Alice extracts a single random bit valued 0 or 1 respectively. Clearly if a bit was extracted, it is 0 or 1 with equal probability. We now define W_i as a Bernoulli r.v. that is set to one if a random bit was extracted:

$$W_i = \mathbb{1}\left(Z_{2i-1}Z_{2i} = 01 \vee Z_{2i-1}Z_{2i} = 10\right),$$

such that $\Pr(W_i = 1) = 2\varepsilon(1-\varepsilon)$. Therefore, the (random) number of extracted bits is

$$N_R = \sum_{i=1}^{n_R/2} W_i,$$

and the probability of failure in the random bit extraction is

$$\Pr(\mathcal{E}_4) = \Pr(N_R < n_U).$$

We now set

$$n_R = \frac{n_U}{\varepsilon(1-\varepsilon)(1-\delta)} = O(\sqrt{n})$$

for some fixed $0 < \delta < 1$. Using the multiplicative form of Chernoff's bound

$$\begin{aligned}\Pr(\mathcal{E}_4) &= \Pr\left(\sum_{i=1}^{n_R/2} W_i < (1-\delta)\mathbb{E}\sum_{i=1}^{n_R/2} W_i\right) \\ &\leq e^{-\frac{\delta^2 n_U}{2(1-\delta)}} \\ &= e^{-O(\sqrt{n})}\end{aligned}$$

□

Using Lemma 9, the explicit scheme that extracts the randomness from the channel has a vanishing error probability with the same rate in as in Theorem 3.

C. Treating ties as erasures

We start this discussion by observing a simple example of a *tie*, which is the erasure event in the BEC. Clearly, if the channel output is an erasure, i.e., $Y = \mathsf{E}$, then $\Pr(Y = \mathsf{E} \mid X = 0) = \Pr(Y = \mathsf{E} \mid X = 1)$ and a tie occurs. Suppose now, that we would like to adapt the coding scheme of Theorem 3, which gives a rate $R_{BSC}(\delta, k)$ for a BSC(δ), for a BEC(δ). Randomly breaking the tie, i.e., uniformly drawing $Y = 0$ or $Y = 1$ in the case of $Y = \mathsf{E}$ will reduce the BEC(δ) to a BSC($\delta/2$)

X \ Y	0	E	1
0	$1-\delta$	ϵ	$\delta-\epsilon$
1	$\delta-\epsilon$	ϵ	$1-\delta$

TABLE II
THE TRANSITION MATRIX $P_{Y|X}$ OF A BSEC($\delta-\epsilon,\epsilon$)

and the coding scheme designated for a BSC could be applied. However, we note that the erasure event in the BEC(δ) has the same probability of the error event in the BSC(δ), which is to be detected in the error detection phase of the rewind-if-error scheme. However, since the erasure is naturally detected by its receiver without requiring an error detection procedure, the rewind-if-error for the BSC could potentially be used, without requiring randomness, and with an improved efficiency.

We can now extend the notion of treating ties as erasures to the general case of a BMS channel. Before we proceed it is instrumental to define *binary channel with symmetric error and erasure*, BSEC($\delta-\epsilon,\epsilon$), whose transition matrix $P_{Y|X}$ appears in Table II. It is clear from the definition that $\delta \in [0,1/2]$ and $\epsilon \in [0,\delta]$, where $\epsilon = 0$ for a BSC(δ) and $\epsilon = \delta$ for a BEC(ϵ). In addition, it is easy to see that for any $\epsilon \in [0,\delta]$, the capacity of the BSEC($\delta-\epsilon,\epsilon$) is

$$(1-\epsilon)\left(1-h\left(\frac{1-\delta}{1-\epsilon}\right)\right),$$

which can be proved by analysis to be strictly larger than $C_{\mathrm{Sh}}(\delta)$ for every $0<\epsilon\leq\delta$.

We now give a non-random version of Definition 4 and Lemma 8, in which ties are marked as erasures:

Definition 5. *[ρ-repetition channel with erasures] Let $P^{(\rho,\mathsf{E})}_{\tilde{Y}|\tilde{X}}$ be the ρ-repetition channel with erasure, corresponding to a BMS($P_{Y|X}$) channel, obtained by transmitting ρ repetitions of the bit $\tilde{X}$ through BMS($P_{Y|X}$) channel and taking*

$$\tilde{Y} = \begin{cases} 0 & \text{if} \quad \prod_{i=1}^{\rho} P_{Y_i|X=0}(Y_i) > \prod_{i=1}^{\rho} P_{Y_i|X=1}(Y_i) \\ 1 & \text{if} \quad \prod_{i=1}^{\rho} P_{Y_i|X=0}(Y_i) < \prod_{i=1}^{\rho} P_{Y_i|X=1}(Y_i) \\ \mathsf{E} & \text{if} \quad \prod_{i=1}^{\rho} P_{Y_i|X=0}(Y_i) = \prod_{i=1}^{\rho} P_{Y_i|X=1}(Y_i) \end{cases}$$

Lemma 10. *For any BMS($P_{Y|X}$) channel with Shannon capacity $C_{\mathrm{Sh}}(P_{Y|X}) = C$ the corresponding ρ-repetition with erasure channel $P^{(\rho,\mathsf{E})}_{\tilde{Y}|\tilde{X}}$ is a BSEC($\delta-\epsilon,\epsilon$) with $\epsilon \in [0,\delta]$ and $\delta \leq \beta^{\rho}$ where β is as in Lemma 8.*

Proof. The proof follows the same lines as the proof of Lemma 8 by making two observations. The first is by noting that in Definition 1 it was implied that an erasure event in a BMS channel corresponds to the statistic $g(Y) = (T, X \oplus Z_T)$ with $T = 1/2$ and $X \oplus Z_T$, which is a Bernoulli($1/2$) random bit. In Definition 5, as well as in the standard BEC definition, such a bit is not produced. However, we note that in the log-likelihood ratio function used for the decision (26), the value of the random bit is not used. The second observation is by noting that in Lemma 8, ties were pessimistically regarded as errors with probability one, where in fact, the random tie breaking reduces their respective error probability to half. Therefore, marking ties as erasures, the aggregate probability of erasure and error is δ and the induced channel is a BSEC($\delta-\epsilon,\epsilon$) with δ as in Lemma 8 and $\epsilon \in [0,\delta]$. □

We are now ready to present the rewind-if-error coding scheme, without tie breaking. We note that ties can appear in two contexts: i) If the original BMS channel had an erasure event (i.e., the probability of $T = 1/2$ is strictly positive). ii) If the BMS channel was reduced to BSC using Lemma 8 and ties occurred in the decoding. We note that ties cannot occur in the repetition coding used for the transmission of the error detection bits in the BSC scheme, since the number of repetitions is always odd.

For for contexts the rewind-if-error scheme can be modified as follows: when a party receives an erasure, it uses the zero value in order to calculate its next bit of the transcript. Then, at the end of the corresponding rewind window, the standard error detection procedure is bypassed and an error is

announced. If the erasure was detected by Bob, he simply sets the rewind bit to one and sends it to Alice. If it was detected by Alice, she signals a designated symbol to Bob, indicating the erasure. We note that in the first layer an additional bit was reserved for this purpose. In higher layers, the bound in (12) ensures that the extra symbol could be signaled without requiring additional bits.

For the sake of completeness, the issue of erasures should also be discussed in the context of randomness extraction in Subsection X-B. Here, we note that if the channel used for randomness extraction can be reduced to a BSEC($\delta-\epsilon,\epsilon$), with $\epsilon < \delta$, Lemma 9 could still be used, changing n_R only by a constant factor and leaving it in an order of magnitude of $O(\sqrt{n})$. In the extreme case $\epsilon = \delta$ (a *pure* BEC), Lemma 9 could not be used. However, in this case all the errors in the scheme in all layers (including the errors of the repetition used for the error detection bits) are marked as erasure. Therefore, the random error detection procedure of Lemma 3 need not be used, and random bits need not be extracted from the channel.

XI. Concluding Remarks

In this paper we revisited the problem of interactive communication over noisy channels originally introduced by Schulman [3], and studied the problem from an information- and communication-theoretic perspective. We started by defining the interactive channel capacity with respect to a protocol and not with respect to a distributed computing problem. As a consequence, our definitions do not use the notion of communication complexity. We then presented a structured and deterministic rewind-if-error coding scheme, and used it to calculate a lower bound for the ratio between the Shannon capacity and the essential interactive capacity of every BMS channel. To the best of our knowledge, this is the first time that a numerical value is attached to this ratio.

We note that the current value of the lower bound can likely be further improved using different coding schemes. A nontrivial upper bound on the ratio between the Shannon capacity and the essential interactive capacity for a fixed channel (i.e., not in the limit of a very clean channel) remains an intriguing open question even in the simplest binary symmetric case.

Appendix A
Proof of Corollary 1

We begin by writing (8) as

$$R_{BSC}(\varepsilon,k)=\frac{1-A(\varepsilon,k)}{1+B(\varepsilon,k)}$$

where

$$\begin{aligned}A(\varepsilon,k) &\triangleq k\varepsilon+(2+\log k)\beta^{\tilde{a}}\\ &+\frac{k^2}{k-1}\left(P_{e1}+3\beta^{a+4}k\log k\frac{2-\beta^2k}{(1-\beta^2k)^2}\right)\\ &+3\beta^{a+4}k^2\log k\frac{2-\beta^2k}{(1-\beta^2k)^2}+\xi\end{aligned}$$

and

$$\begin{aligned}B(\varepsilon,k) &\triangleq \frac{\tilde{a}(2+\log k)}{k}\\ &+3\log k\left[\frac{a(2k-1)}{(k-1)^2}+\frac{4k}{(k-1)^3}+\frac{4k-2}{k(k-1)^2}\right]\\ &+o(1).\end{aligned}$$

Using the inequality $1/(1+x)<1-x$ for $x>0$ and the fact that $A(\varepsilon,k)\geq 0$, $B(\varepsilon,k)\geq 0$ gives:

$$\begin{aligned}R_{BSC}(\varepsilon,k) &\geq 1-A(\varepsilon,k)-B(\varepsilon,k)+A(\varepsilon,k)B(\varepsilon,k)\\ &\geq 1-A(\varepsilon,k)-B(\varepsilon,k).\end{aligned}$$

We use the definitions $\beta=2\sqrt{\varepsilon(1-\varepsilon)}$, $a=3$ and $\tilde{a}=5$ and assume from this point on that $k\to\infty$ and $\varepsilon=o(1/k)$. Neglecting all high order terms we obtain:

$$B(\varepsilon,k)=O\left(\frac{\log k}{k}\right)$$

and

$$A(\varepsilon,k)=k\varepsilon+O(k)P_{e1}+\xi+o(1).$$

We now recall (9)

$$\begin{aligned}P_{e1} &\leq \frac{1}{2k}\left(1+2(k-1)(1-2\varepsilon)^{\frac{k}{2}}+(1-2\varepsilon)^k\right)\\ &-(1-\varepsilon)^k+(2+\log k)\beta^{\tilde{a}}\end{aligned}$$

$$= O(k\varepsilon^2).$$

and set

$$\xi = k^{-2},$$

which ensures that Lemma 6 holds (see (44)) obtaining

$$R_{BSC}(\varepsilon, k) \geq$$
$$1 - \left(k\varepsilon + O(k^2\varepsilon^2) + k^{-2} + o(1) + O\left(\frac{\log k}{k}\right)\right).$$

Finally, setting $\varepsilon = \frac{\log k}{k^2}$ as in [13] gives

$$\begin{aligned} R_{BSC}(\varepsilon, k) &\geq 1 - O\left(\frac{\log k}{k}\right) \\ &= 1 - O\left(\sqrt{-\varepsilon \log \varepsilon}\right) \\ &= 1 - O\left(\sqrt{h(\varepsilon)}\right). \end{aligned}$$

APPENDIX B
PROOF OF LEMMA 6

We would like to prove that

$$\lim_{n\to\infty} \Pr(\mathcal{E}_1) = \lim_{n\to\infty} \Pr\left(j^A(T) < n\right) = 0.$$

We start by recalling (13)

$$j^A(T) \geq T\left(1 - \sum_{l=1}^{L} \overline{b_l}\right)$$

The probability of the complementary event is:

$$\Pr\left(j^A(T) \geq n\right) \geq \Pr\left(1 - \sum_{l=1}^{L} \overline{b_l} \geq \frac{n}{T}\right). \quad (34)$$

By (22) we have

$$\frac{n}{T} = 1 - \sum_{l=1}^{\infty} \overline{P}_{b_l} - \xi \leq 1 - \sum_{l=1}^{L} \overline{P}_{b_l} - \xi,$$

so we can further bound (34) by

$$\begin{aligned} &\Pr\left(j^A(T) \geq n\right) \\ &\geq \Pr\left(1 - \sum_{l=1}^{L} \overline{b_l} \geq 1 - \sum_{l=1}^{L} P_{b_l} - \xi\right) \\ &= 1 - \Pr\left(\sum_{l=1}^{L} \overline{b_l} > \sum_{l=1}^{L} P_{b_l} + \xi\right). \end{aligned}$$

Therefore $\Pr(\mathcal{E}_1) \leq \Pr\left(\sum_{l=1}^{L} \overline{b_l} > \sum_{l=1}^{L} P_{b_l} + \xi\right)$ and the lemma can be proved by proving

$$\lim_{T\to\infty} \Pr\left(\sum_{l=1}^{L} \overline{b_l} > \sum_{l=1}^{L} P_{b_l} + \xi\right) = 0.$$

We start by observing that

$$\begin{aligned} &\Pr\left(\sum_{l=1}^{L} \overline{b_l} > \sum_{l=1}^{L} P_{b_l} + \xi\right) \\ &\leq \Pr\left(\bigcup_{l=1}^{L} \left[\overline{b_l} > P_{b_l} + \frac{\xi}{L}\right]\right) \\ &\leq \sum_{l=1}^{L} \Pr\left(\overline{b_l} > P_{b_l} + \frac{\xi}{L}\right) \\ &= S_1 + S_2, \end{aligned}$$

where $S_1 \triangleq \sum_{l=1}^{\lfloor \frac{3}{4}L \rfloor} \Pr\left(\overline{b_l} > P_{b_l} + \frac{\xi}{L}\right)$ and $S_2 \triangleq \sum_{l=\lfloor \frac{3}{4}L \rfloor + 1}^{L} \Pr\left(\overline{b_l} > P_{b_l} + \frac{\xi}{L}\right)$.

Starting with S_1, by the definition $\overline{b_l}$ in (14), the l'th summand of S_1 is:

$$\begin{aligned} &\Pr\left(\overline{b_l} > P_{b_l} + \frac{\xi}{L}\right) \quad (35) \\ &= \Pr\left(\sum_{m=1}^{k^{L-l}} b_l^A(m) > k^{L-l}(P_{b_l} + \tfrac{\xi}{L})\right). \end{aligned}$$

We recall that $b_l^A(m)$ are Bernoulli(P_{b_l}) r.v.'s with limited independence. The following straightforward generalization of the Chernoff-–Hoeffding Theorem is now useful:

Lemma 11. *Let $X_1, ..., X_n$ be a series of Bernoulli(p) r.v.'s, divided into groups of ℓ elements. We assume that all distinct groups statistically independent but the r.v.'s within every group might be statistically dependent. Namely, let $i, \tilde{i} \in \{1, ..., n/\ell\}$ and $j, \tilde{j} \in \{1, ..., \ell\}$. It is given that $X_{(i-1)\ell+j}$ and $X_{(\tilde{i}-1)+\tilde{j}\ell}$ are statistically independent for every $i \neq \tilde{i}$ and every $j, \tilde{j}$ but might be statistically dependent for $i = \tilde{i}$ and some $j \neq \tilde{j}$. Then, for every $0 < \epsilon < 1 - p$:*

$$\Pr\left(\sum_{i=1}^{n} X_i \geq n(p + \epsilon)\right) \leq e^{-\frac{n}{\ell} 2\epsilon^2}. \quad (36)$$

Proof. We begin with the standard derivation of the Chernoff bound for $\sum_{i=1}^{n} X_i$:

$$\begin{aligned}
&\Pr\left(\sum_{i=1}^{n} X_i \geq n(p+\epsilon)\right) \\
&\leq \min_{t>0} e^{-tn(p+\epsilon)} \mathbb{E}\left(e^{t\sum_{i=1}^{n} X_i}\right) \\
&\leq \min_{t>0} e^{-tn(p+\epsilon)} \mathbb{E}\left(e^{t\sum_{i=1}^{n/\ell}\sum_{j=1}^{\ell} X_{(i-1)\ell+j}}\right) \\
&= \min_{t>0} e^{-tn(p+\epsilon)} \mathbb{E}\left(\prod_{i=1}^{n/\ell}\prod_{j=1}^{\ell} e^{tX_{(i-1)\ell+j}}\right) \\
&= \min_{t>0} e^{-tn(p+\epsilon)} \prod_{i=1}^{n/\ell} \mathbb{E}\left(\prod_{j=1}^{\ell} e^{tX_{(i-1)\ell+j}}\right) \quad (37)
\end{aligned}$$

where in (37) we used the independence assumptions of groups of length ℓ. We now prove the following bound for the first group, $i = 1$

$$\mathbb{E}\left(\prod_{j=1}^{\ell} e^{tX_j}\right) \leq \mathbb{E}\left(e^{\ell t X_1}\right). \quad (38)$$

The proof is based on using Hölder's inequality iteratively. We start by recalling Hölder's inequality for the expectation of real valued non-negative random variables, $W, V \in \mathbb{R}$, $W, V \geq 0$ and $p > 1$:

$$\mathbb{E}(W \cdot V) \leq \left(\mathbb{E}\left(W^{\frac{p}{p-1}}\right)\right)^{\frac{p-1}{p}} \left(\mathbb{E}\left(V^p\right)\right)^{\frac{1}{p}}. \quad (39)$$

Using (39) for $\mathbb{E}\left(\prod_{j=1}^{\ell} e^{tX_j}\right)$ with $W = \prod_{j=1}^{\ell-1} e^{tX_j}$, $V = e^{tX_\ell}$ and $p = \ell$ gives

$$\begin{aligned}
&\mathbb{E}\left(\prod_{j=1}^{\ell} e^{tX_j}\right) \\
&\leq \left(\mathbb{E}\prod_{j=1}^{\ell-1} e^{\frac{\ell}{\ell-1}tX_j}\right)^{\frac{\ell-1}{\ell}} \left(\mathbb{E}\left(e^{\ell t X_\ell}\right)\right)^{\frac{1}{\ell}}. \quad (40)
\end{aligned}$$

Using (39) for $\mathbb{E}\left(\prod_{j=1}^{\ell-1} e^{\frac{\ell}{\ell-1}tX_j}\right)$ with $W = \prod_{j=1}^{\ell-2} e^{\frac{\ell}{\ell-1}tX_j}$, $V = e^{\frac{\ell}{\ell-1}tX_{\ell-1}}$ and $p = \ell - 1$ gives

$$\begin{aligned}
&\mathbb{E}\left(\prod_{j=1}^{\ell-1} e^{\frac{\ell}{\ell-1}tX_j}\right) \\
&\leq \left(\mathbb{E}\prod_{j=1}^{\ell-2} e^{\frac{\ell}{\ell-2}tX_j}\right)^{\frac{\ell-2}{\ell-1}} \left(\mathbb{E}\left(e^{\ell t X_{\ell-1}}\right)\right)^{\frac{1}{\ell-1}} \quad (41)
\end{aligned}$$

Plugging (41) into (40) and taking into account that X_ℓ and $X_{\ell-1}$ have the same marginal distribution as X_1 gives:

$$\begin{aligned}
&\mathbb{E}\left(\prod_{j=1}^{\ell} e^{tX_j}\right) \quad (42) \\
&\leq \left(\mathbb{E}\prod_{j=1}^{\ell-2} e^{\frac{\ell}{\ell-2}tX_j}\right)^{\frac{\ell-2}{\ell}} \left(\mathbb{E}\left(e^{\ell t X_1}\right)\right)^{\frac{2}{\ell}}.
\end{aligned}$$

We now implement this process iteratively on the left hand term the upper bound in (42) for $p = \ell - 2$ to $p = 2$ finally giving (38).

We now notice that (38) depends only on the marginal distribution of a single sample, which is assumed to be Bernoulli(p), so it should hold for all groups $i \in \{1, ..., n/\ell\}$. Therefore we can use (38) for all the elements in the outer product in (37) giving:

$$\begin{aligned}
&\Pr\left(\sum_{i=1}^{n} X_i \geq n(p+\epsilon)\right) \\
&\leq \min_{t>0} e^{-tn(p+\epsilon)} \left(\mathbb{E}\left(e^{t\ell X_1}\right)\right)^{n/\ell} \\
&\leq \left(\min_{t\ell>0} e^{-t\ell(p+\epsilon)} \mathbb{E}\left(e^{t\ell X_1}\right)\right)^{n/\ell} \\
&= e^{-\frac{n}{\ell} d_n((p+\epsilon)||p)}. \quad (43)
\end{aligned}$$

where (43) is by the standard minimization of the Chernoff bound and $d_n(p||q) \triangleq p \ln \frac{p}{q} + (1-p) \ln \frac{1-p}{1-q}$ is Kullback-Leibler Divergence between two Bernoulli random variable with probabilities p and q, which is now calculated with respect to the natural logarithm basis. Finally, by Pinsker's inequality we bound the divergence by $d_n(p+\epsilon||p) \geq 2\epsilon^2$ and obtain (36). □

We can now use Lemma 11 to bound (35). Recalling the discussion from Subsection X-A, at every layer $1 < l \leq L$, there are k^{L-l} blocks for which error detection is applied using Definition 3. We assume that only $k^{\lceil (L-l)/2 \rceil}$ independent test points are used, which are changed every $k^{\lfloor (L-l)/2 \rfloor}$

blocks. So, we can use Lemma 11 on (35) where the number of independent groups is $\frac{n}{\ell} = k^{\lceil (L-l)/2 \rceil}$ yielding:

$$\begin{aligned}
&\Pr\left(\sum_{m=1}^{k^{L-l}} b_l^A(m) > k^{L-l}(P_{b_l} + \frac{\xi}{L})\right) \\
&\leq e^{-k^{\lceil (L-l)/2 \rceil}\frac{2\xi^2}{L^2}} \\
&\leq e^{-k^{(L-l)/2}\frac{2\xi^2}{L^2}}
\end{aligned}$$

Summing all the element is of S_1 yields:

$$S_1 \leq \sum_{l=1}^{\lfloor \frac{3}{4}L \rfloor} e^{-k^{(L-l)/2}\frac{2\xi^2}{L^2}} \leq \frac{3}{4}L \cdot e^{-k^{L/8}\frac{2\xi^2}{L^2}}. \quad (44)$$

The second transition is by using the maximal summand obtained at $l = \lfloor \frac{3}{4}L \rfloor$. Recalling that $L = \log_k T$, it is clear that $\lim_{T\to\infty} S_1 = \lim_{L\to\infty} S_1 = 0$.

Proceeding with S_2:

$$\begin{aligned}
S_2 &= \sum_{l=\lfloor \frac{3}{4}L \rfloor + 1}^{L} \Pr\left(\overline{b_l} > P_{b_l} + \frac{\xi}{L}\right) \\
&\leq \sum_{l=\lfloor \frac{3}{4}L \rfloor + 1}^{L} \Pr\left(\overline{b_l} > 0\right).
\end{aligned}$$

Observe that if $\overline{b_l} > 0$ then at least one rewind bit at level l is set to one. So, we can use the union bound and obtain

$$\Pr\left(\overline{b_l} > 0\right) \leq k^{L-l} P_{b_l}. \quad (45)$$

Recalling (20)

$$\begin{aligned}
P_{b_l} &\leq k^{2-l}\left(kP_{e1} + 3\beta^{a+4}k^2 \log k \frac{2-\beta^2 k}{(1-\beta^2 k)^2}\right) \\
&\quad + 3\beta^a (\log k) l \beta^{2l}
\end{aligned}$$

we can further bound (45) by

$$\begin{aligned}
&\Pr\left(\overline{b_l} > 0\right) \leq \quad (46) \\
&k^L\left(k^{-2l}\left(k^3 P_{e1} + 3\beta^{a+4}k^4 \log k \frac{2-\beta^2 k}{(1-\beta^2 k)^2}\right)\right. \\
&\left. + 3\beta^a (\log k) l \left(\frac{\beta^2}{k}\right)^l\right).
\end{aligned}$$

Observing that the bound in (46) is monotonically decreasing in l for a sufficiently large l we can bound the summands of S_2 by the term obtained at $l = 3/4L$, yielding:

$$\begin{aligned}
S_2 &\leq \frac{L}{4}k^{-L/2}\left(k^3 P_{e1} + 3\beta^{a+4}k^4 \log k \frac{2-\beta^2 k}{(1-\beta^2 k)^2}\right) \\
&\quad + \frac{9}{16}\beta^a (\log k) L^2 \left(\left(\frac{\beta^2}{k}\right)^{3/4} k\right)^L.
\end{aligned}$$

It is clear that the left hand term is monotonically decreasing in L. Analyzing the right hand term, we use the definition of β and we observe that

$$\begin{aligned}
\left(\frac{\beta^2}{k}\right)^{3/4} k &= \left(\beta^6 k\right)^{1/4} < \left(2^6 \varepsilon^3 k\right)^{1/4} \\
&< (2^6/(8k)^3 k)^{1/4} < (2^3 k^2)^{-1/4} < 1
\end{aligned}$$

where the third transition is due to the assumption that $\varepsilon < 1/(8k)$ in Theorem 3. All in all, setting $L = \log_k T$ guarantees that $\lim_{T\to\infty} S_2 = 0$, which concludes the proof of Lemma 6.

REFERENCES

[1] C. E. Shannon, "Two-way communication channels," in *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

[2] R. Gelles, "Coding for interactive communication: a survey," 2019. [Online]. Available: http://www.eng.biu.ac.il/~gellesr/survey.pdf

[3] L. J. Schulman, "Communication on noisy channels: A coding theorem for computation," in *Proceedings., 33rd Annual Symposium on Foundations of Computer Science*. IEEE, 1992, pp. 724–733.

[4] P. Elias, "The efficient construction of an unbiased random sequence," *The Annals of Mathematical Statistics*, pp. 865–870, 1972.

[5] J. Von Neumann, "Various techniques used in connection with random digits," *Appl. Math Ser*, vol. 12, no. 5, pp. 36–38, 1951.

[6] F. Leighton and R. Rivest, "Estimating a probability using finite memory," *IEEE Transactions on Information Theory*, vol. 32, no. 6, pp. 733–742, 1986.

[7] R. G. Gallager, *Low-Density Parity-Check Codes*. MIT Press, 1963.

[8] ——, *Information Theory and Reliable Communication*. New York: John Wiley & Sons, 1968.

[9] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge university press, 2008.

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

[11] A. Guillén i Fàbregas, I. Land, and A. Martinez, "Extremes of error exponents," *IEEE Transactions on Information Theory*, vol. 59, no. 4, pp. 2201–2207, 2013.

[12] R. Pedarsani, S. H. Hassani, I. Tal, and E. Telatar, "On the construction of polar codes," in *ISIT*. IEEE, 2011, pp. 11–15.

[13] G. Kol and R. Raz, "Interactive channel capacity," in *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM, 2013, pp. 715–724.

[14] E. Kushlevitz and N. Nisan, *Communication Complexity*. Cambridge University Press, 1997.

[15] S. Agrawal, R. Gelles, and A. Sahai, "Adaptive protocols for interactive communication," in *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2016, pp. 595–599.

[16] V. Dani, T. P. Hayes, M. Movahedi, J. Saia, and M. Young, "Interactive communication with unknown noise rate," *Information and computation*, vol. 261, pp. 464–486, 2018.

[17] L. J. Schulman, "Coding for interactive communication," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1745–1756, 1996.

[18] M. Ghaffari, B. Haeupler, and M. Sudan, "Optimal error rates for interactive coding i: Adaptivity and other settings," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, 2014, pp. 794–803.

[19] B. Haeupler, "Interactive channel capacity revisited," in *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*. IEEE, 2014, pp. 226–235.

[20] A. C.-C. Yao, "Some complexity questions related to distributive computing (preliminary report)," in *Proceedings of the eleventh annual ACM symposium on Theory of computing*. ACM, 1979, pp. 209–213.

[21] R. Gelles, B. Haeupler, G. Kol, N. Ron-Zewi and A. Wigderson, "Explicit capacity approaching coding for interactive communication," *IEEE Transactions on Information Theory*, vol. 64, pp. 6546 – 6560, October 2018.

[22] R. Gelles and B. Haeupler, "Capacity of interactive communication over erasure channels and channels with feedback," *SIAM Journal on Computing*, vol. 46, no. 4, pp. 1449–1472, 2017.

[23] A. Ben-Yishai, Y-H Kim, O. Ordentlich and O. Shayevitz, "The interactive capacity of the binary symmetric channel is at least 1/40 the shannon capacity," in *ISIT*, 2019.

[24] T. Kløve and V. Korzhik, *Error detecting codes: general theory and their application in feedback communication systems*. Springer Science & Business Media, 2012, vol. 335.

[25] D. P. Palomar and S. Verdú, "Lautum information," *IEEE transactions on information theory*, vol. 54, no. 3, pp. 964–975, 2008.

[26] M. Hellman and J. Raviv, "Probability of error, equivocation, and the chernoff bound," *IEEE Transactions on Information Theory*, vol. 16, no. 4, pp. 368–372, 1970.