The Interactive Capacity of the Binary Symmetric Channel is at Least $1/40$ the Shannon Capacity

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

***Abstract*—We define the interactive capacity of the binary symmetric channel (BSC) as the maximal rate for which any interactive protocol can be fully and reliably simulated over a pair of BSC's. We show that this quantity is at least $1/40$ of the BSC Shannon capacity, uniformly for all channel crossover probabilities. Our result is based on a public-coin rewind-if-error coding scheme in the spirit of Kol & Raz 2013 [1].**

I. Introduction

Let a (binary) protocol $\boldsymbol{\pi} = (\pi_1, \pi_2, ..., \pi_n)$ of length $|\boldsymbol{\pi}| = n$, be interactively generated by Alice and Bob, where Alice speaks at the (predetermined) time points $A \subseteq \{1, ..., n\}$, and Bob speaks at the (predetermined) time points $B = \{1, ..., n\} \setminus A$. The bits of the protocol are generated by the parties using a set of (possibly stochastic) transmission functions

$$\pi_i = \phi_i(\boldsymbol{\pi}^{i-1}).$$

The transmission functions are predetermined by the parties before transmission, but Alice's functions are unknown to Bob and vice versa. In what follows, we conveniently identify the functions ϕ_i with their outputs π_i. We therefore think of $\boldsymbol{\pi}$ as both the protocol itself and the binary vector of its transcript.

Suppose now that Alice and Bob can only communicate through a pair of independent BSC with a known crossover probabilities $0 < \varepsilon < \frac{1}{2}$ (BSC(ε)). Below we denote the input, output and noise of the channels by X, $\widetilde{X}$ and Z respectively (with the appropriate time index). The channel input-to-output relation is thus

$$\widetilde{X} = X \oplus Z,$$

where $Z \sim$ Bernoulli(ε) and $\oplus$ is addition over $\mathbb{F}_2$. The Shannon capacity of the BSC(ε) is

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

where $h(\varepsilon) \triangleq -\varepsilon \log \varepsilon - (1-\varepsilon)\log(1-\varepsilon)$ is the binary entropy function, and $\log(x) \triangleq \log_2(x)$.

Alice and Bob wish to simulate $\boldsymbol{\pi}$ reliably by communicating over these noisy channels, with an error probability that vanishes with n. To that end, they use an (n, N) coding scheme Σ that operates on the protocol $\boldsymbol{\pi}$ and uses the channel N times. It consists of a partition $\tilde{A} \cup \tilde{B} = \{1, ..., N\}$ where $\tilde{A}$ (resp. $\tilde{B}$) is the set of time indices where Alice (resp. Bob) speaks. At time $j \in \tilde{A}$ Alice sends some function of her protocol functions $\{\pi_i\}_{i\in A}$ and of everything she has received so far from Bob. At time $j \in \tilde{B}$ Bob sends some function of his protocol functions $\{\pi_i\}_{i\in B}$ and of everything he has received so far from Alice. The rate of the scheme is $R = \frac{n}{N}$ bits per channel use.

When the communication terminates, Alice and Bob produce estimates $\hat{\boldsymbol{\pi}}_A(\Sigma)$ and $\hat{\boldsymbol{\pi}}_B(\Sigma)$, respectively, of the clean protocol $\boldsymbol{\pi}$. The error probability attained by Σ when simulating $\boldsymbol{\pi}$ is therefore

$$P_e(\Sigma, \boldsymbol{\pi}) \triangleq \Pr\left(\hat{\boldsymbol{\pi}}_A(\Sigma) \neq \boldsymbol{\pi} \vee \hat{\boldsymbol{\pi}}_B(\Sigma) \neq \boldsymbol{\pi}\right).$$

A rate R is called *achievable* if there exists a sequence Σ_n of (n, N_n) coding schemes with rates $\frac{n}{N_n} \geq R$, such that

$$\lim_{n\to\infty} \sup_{\pi:|\pi|=n} P_e(\Sigma_n, \boldsymbol{\pi}) = 0,$$

where the supremum is taken over all length n protocols (collection of transmission functions) and speaking orders. Accordingly, we define the interactive capacity $\mathsf{C_I}(\varepsilon)$ of the BSC(ε) as the supremum of all achievable rates for a given crossover probability ε.

We note that the above capacity definition does not use the notion of communication complexity as is usually done in the literature (see Subsection I-A). A trivial upper bound for the interactive capacity is $\mathsf{C_I}(\varepsilon) \leq \mathsf{C_{Sh}}(\varepsilon)$, simply since the Shannon capacity is the maximal possible rate in the special (non-interactive) case

A. Ben-Yishai and O. Shayevitz are 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. O. Ordentlich is with the Hebrew University of Jerusalem, Israel. 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 grant no. ISF 1791/17. The work of O. Shayevitz was supported by an ERC grant no. 639573.

where Alice and Bob generate their bits uniformly at random, independently from each other. In the general interactive case however, the parties cannot predict future transmissions hence cannot reliably convey the protocol over the noisy channels using block codes as is done in the classical Shannon setup.

Our main result is a lower bound for the ratio between the interactive capacity and the Shannon capacity.

Theorem 1. $\inf_{0\leq\varepsilon<1/2}\dfrac{\mathsf{C}_{\mathsf{I}}(\varepsilon)}{\mathsf{C}_{\mathsf{Sh}}(\varepsilon)} \geq 0.0261.$

The theorem is proved by an analysis of a "rewind-if-error scheme" in the spirit of [1]. We analyze the rate of the scheme for a fixed small ε and account for the rate loss incurred by a repetition code used for channels with larger crossover probabilities. In the coding scheme, and in the rest of the paper, it is assumed that the order of speakers is alternating, namely, that A and B are the sets of odd and even number between 1 and n respectively. We account for the general case of non-alternating order of speakers by standardly adding dummy transmission and reducing the rate by a factor of at most two. Namely, for protocols with alternating speakers, we prove a version of Theorem 1 with a ratio of 0.0523 instead of 0.0261.

A. Related work

The interactive communication problem introduced by Schulman [2], [3] is motivated by the communication complexity problem [4] in which a binary function f is to be computed with negligible error by two parties, Alice and Bob, each having only part of its input. For this purpose they exchange bits using an interactive protocol, whose minimal length is the *communication complexity* of f denoted by $CC(f)$. In the interactive communication setup [2], Alice and Bob must achieve their goal by communicating through a pair of independent BSC(ε). The minimal expected length of an interactive protocol attaining this goal is now denoted by $CC_{\varepsilon}(f)$.

In [1], Kol and Raz defined the interactive capacity as

$$\mathsf{C}_{\mathsf{I}}^{\mathsf{KR}}(\varepsilon) \triangleq \lim_{n\to\infty}\min_{f:CC(f)=n}\frac{n}{CC_{\varepsilon}(f)}. \tag{1}$$

They proved that $\mathsf{C}_{\mathsf{I}}^{\mathsf{KR}}(\varepsilon)\geq 1-O(\sqrt{h(\varepsilon)})$ in the limit of $\varepsilon\to 0$ under the additional assumption that the order of speakers has a fixed period. For a fixed nonzero ε, the coding scheme presented in [2] (which precedes [1]) uses error detection and retransmissions to achieve a constant fraction of the Shannon capacity, i.e., showing that $\mathsf{C}_{\mathsf{I}}^{\mathsf{KR}}(\varepsilon)=\Omega(\mathsf{C}_{\mathsf{Sh}}(\varepsilon))$, but the constant has not been computed.

We note that our interactive capacity definition is stricter than (1), at least in principle, as it requires reconstruction of the entire protocol transcript and not only the computation of the function, hence $\mathsf{C}_{\mathsf{I}}(\varepsilon)\leq\mathsf{C}_{\mathsf{I}}^{\mathsf{KR}}(\varepsilon)$. Our definition enjoys the property of being decoupled from any source coding problem such as function communication complexity.

There have been works in the literature that assume the order of speakers in the noiseless and simulating protocols can be adaptive and determined on the fly, which can have a positive impact on the capacity [5]. This however creates a nonzero probability of speaker *collision*, which requires to modify the definition of the physical channel from a pair of independent BSCs to a full two-way channel. Hence, the predetermined order of speakers assumed in this paper is necessary and cannot be dispensed of in the BSC case under consideration. Another issue is the type of randomness used by the coding scheme, which can be either public, private or none. In [2] only private randomness was assumed, while [1] assumes public randomness. In this paper we assume public randomness as well. Clearly, the use of randomness does not affect Shannon's capacity which is taken as reference.

II. Preliminaries

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 variable with probabilities p and q. The following two basic lemmas are used several times in our analysis:

Lemma 1 (Repetition over BSC)**.** *Let a bit be sent over BSC(ε) using ρ repetitions and decoded by 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*

$$P_e\leq(2\sqrt{\varepsilon(1-\varepsilon)})^{\rho}=2^{-\rho D(\frac{1}{2}||\varepsilon)}.$$

The proof is by the Chernoff bound ([6] for example).

Lemma 2 (Error detection by hashing)**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be discrete alphabets. Let f be a mapping from $\mathcal{A}$ to $\mathcal{B}$, generated by independently and uniformly drawing a member in $\mathcal{B}$ for every member in $\mathcal{A}$. Then for every $x\neq y\in\mathcal{A}$*

$$\Pr(f(x)=f(y))=|\mathcal{B}|^{-1}.$$

where the probability is with respect to the randomness in the generation of f.

The proof is a straightforward consequence of uniformity and independence of the generation of f.

III. Main results

Theorem 2. *For any ε*

$$\frac{\mathsf{C}_{\mathsf{I}}(\varepsilon)}{\mathsf{C}_{\mathsf{Sh}}(\varepsilon)}\geq\max_{0<\delta<\frac{1}{2}}\frac{\mathsf{C}_{\mathsf{I}}(\delta)}{\log\frac{1}{\delta}+1}.$$

Proof. By Lemma 1, we may transform a BSC(ε) to a BSC(δ'), $\delta' \leq \delta < \varepsilon$ using

$$\rho(\varepsilon, \delta) \triangleq \frac{\log \frac{1}{\delta}}{D(\frac{1}{2}||\varepsilon)} + 1$$

repetitions over the BSC(ε), followed by majority decoding with symmetric tie breaking. Thus, any interactive capacity achieving coding scheme for BSC(δ), can be used to construct a reliable coding scheme for BSC(ε), with rate $\mathsf{C_I}(\delta)/\rho(\varepsilon, \delta)$. In particular, for any $\delta < \varepsilon$, $\mathsf{C_I}(\varepsilon) \geq \mathsf{C_I}(\delta)/\rho(\varepsilon, \delta)$. Dividing by $\mathsf{C_{Sh}}(\varepsilon)$ we obtain $\mathsf{C_I}(\varepsilon)/\mathsf{C_{Sh}}(\varepsilon) \geq \mathsf{C_I}(\delta)/(\rho(\varepsilon, \delta)\mathsf{C_{Sh}}(\varepsilon))$. Let us proceed by upper bounding the denominator:

$$\begin{aligned}\mathsf{C_{Sh}}(\varepsilon)\rho(\varepsilon, \delta) &= \log \frac{1}{\delta} \frac{\mathsf{C_{Sh}}(\varepsilon)}{D(\varepsilon||\frac{1}{2})} + 1\mathsf{C_{Sh}}(\varepsilon) \\ &\leq \log \frac{1}{\delta} \frac{D(\varepsilon||\frac{1}{2})}{D(\frac{1}{2}||\varepsilon)} + 1.\end{aligned}$$

We are left with proving that for all $0 < \varepsilon < \frac{1}{2}$, $D(\varepsilon||\frac{1}{2}) \leq D(\frac{1}{2}||\varepsilon)$. We prove it by analysis: replace ε with $\eta \triangleq \frac{1}{2} - \varepsilon$, so the inequality to be proved becomes $\Delta(\eta) \triangleq D\left(\frac{1}{2} - \eta||\frac{1}{2}\right) - D\left(\frac{1}{2}||\frac{1}{2} - \eta\right) \leq 0$. Now, it is possible to show by differentiation that $\Delta(0) = \frac{d}{d\eta}\Delta(0) = 0$ and $\frac{d^2}{d^\eta}\Delta(0) < 0, \forall 0 < \varepsilon < \frac{1}{2}$. □

Having Theorem 2 at hand, we can take any coding scheme with a known rate for a relatively clean channel, and obtain a lower bound to the interactive capacity of a noisier channel. As stated above, we would take a coding scheme designed for an alternating order of speakers and standardly modify other orders of speakers by adding dummy transmissions and dividing the overall rate by a two. A possible coding scheme is the *rewind-if-error* scheme from [1] whose rate is proved to be $1 - O(\sqrt{h(\varepsilon)})$. However, since to the best of our knowledge, the explicit rate as a function of ε of this scheme (and to the best of our knowledge, of any other scheme) does not appear in the literature, such a calculation is necessary. The remainder of this paper is dedicated for the presentation and analysis of a simple *rewind-if-error* scheme in the spirit of [1], whose rate is given in the following theorem:

Theorem 3. *For any $\xi > 0$ and an integer $k < (2\varepsilon)^{-1}$, there exist a reliable rewind-if error coding scheme with rate:*

$$R(\varepsilon) \geq \frac{1 - \frac{k^2}{k-1}\left(\varepsilon + \frac{\alpha k \beta^{2+a}}{1-\beta^2 k}\right) - \frac{\alpha\beta^{2+a}}{1-\beta^2}\left(1 - \frac{k^2}{1-k\beta^2}\right) - \xi}{1 + \alpha\left(\frac{a}{k-1} + \frac{2k}{(k-1)^2}\right)}. \tag{2}$$

where $a = 3$, $\beta \triangleq 2\sqrt{\varepsilon(1-\varepsilon)}$ and $\alpha \triangleq \lceil 2\log k \rceil + 1$.

The proof consumes most of the remainder of this paper. Theorem 1 now follows by taking (2) as a lower bound for $\mathsf{C_I}(\delta)$, setting $k = 512$, $\delta = \log k/k^2$ using Theorem 2, and dividing the result by two for the general case of non-alternating speakers.

IV. Description of the coding scheme

The *rewind-if-error* scheme is based on uncoded transmission combined with error detection and retransmissions. The parties simulate the protocol in a window of k consecutive transmissions, considering the channel to be error free. They then try to detect errors in this window, and initiate its retransmission in case they were found. Assuming that $k\varepsilon$ is sufficiently small, a substantial number of windows will be error free. The error detection procedure is based on hashes, and is performed over a noisy channel, so it might fail. In order to mitigate the effect of the failures, the process is repeated in a nested fashion. Namely, after k windows are simulated, error detection is applied on all of them, possibly initiating their entire retransmission. After k^2 windows are simulated, error detection is applied on all of them, and so on.

For the sake of clarity of exposition, we give a sequential (rather then a recursive) description of the scheme. We do, however, use its recursive interpretation in the analysis. We start by defining the following notions, which are used as the building blocks of the scheme:

- The *uncoded simulation* is a sequence of bits, generated by the parties and the channel, using the transmission functions and disregarding the channel errors. Alice's and Bob's uncoded simulation vectors are for odd i: $\mathbf{X}_i^A \triangleq (X_1, \widetilde{X}_2, \ldots, X_i)$, and $\mathbf{X}_i^B \triangleq (\widetilde{X}_1, X_2, \ldots, \widetilde{X}_i)$ respectively. For even i they are $\mathbf{X}_i^A \triangleq (X_1, \widetilde{X}_2, \ldots, \widetilde{X}_i)$, and $\mathbf{X}_i^B \triangleq (\widetilde{X}_1, X_2, \ldots, X_i)$ respectively.
- The *cursor* variable indicates the time index of the transmission function which was used by Alice or Bob in the last 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.
- *Rewind bits* are the result of the error detection procedure and are calculated at predetermined points throughout the scheme. These bits indicate whether the protocol should proceed forward, or rewind backward. We denote $T = k^L$ and separate the rewind bits into levels : $l = 1, \ldots, L$. At level 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 bit $b_l^A(m)$ and $b_l^B(m)$ are calculated after exactly mk^l bits of uncoded simulation, and are calculated according to their respective *rewind windows*.

- The *rewind window* $w\left[b_l(m)\right]$ (of either Alice or Bob) contains the bits according to which $b_l(m)$ 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 $1 < i < l$ that were 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 estimates for the protocol: $\hat{\boldsymbol{\pi}}_A$ and $\hat{\boldsymbol{\pi}}_B$. We are now ready to introduce the coding scheme:

Initialization: $i = 0$. $j^A = j^B = 0$. $\mathbf{X}_0^A = \mathbf{X}_0^B = \emptyset$, $\hat{\boldsymbol{\pi}}_A = \hat{\boldsymbol{\pi}}_B = \emptyset$, where $\emptyset$ denotes an empty vector.

Iteration:

- Simulate the protocol for consecutive k times, disregarding the channel errors, as follows. The parties start by advancing i and their respective cursors, j^A, j^B by one. At odd $1 \leq i \leq T$ Alice sends $X_i = \phi_{j^A}(\hat{\boldsymbol{\pi}}_A^{j^A-1})$, and at even $1 \leq i \leq T$, Bob sends $X_i = \phi_{j^B}(\hat{\boldsymbol{\pi}}_B^{j^B-1})$.[1] At odd i Alice updates her uncoded simulation vector by $\mathbf{X}_i^A = (\mathbf{X}_{i-1}^A, X_i)$ and her protocol estimate by $\hat{\boldsymbol{\pi}}^{j^A} = (\hat{\boldsymbol{\pi}}^{j^A-1}, X_i)$ whereas Bob updates $\mathbf{X}_i^B = (\mathbf{X}_{i-1}^B, \widetilde{X}_i)$ and $\hat{\boldsymbol{\pi}}^{j^B} = (\hat{\boldsymbol{\pi}}^{j^B-1}, \widetilde{X}_i)$. The update for even i is done similarly with appropriate alterations.
- For $l = 1$ to L, if $i = mk^l$ for some integer m, then rewind window $w\left[b_l(m)\right]$ has ended. Alice computes her rewind bit $b_l^A(m)$ according to the procedure explained in the sequel. If $b_l^A(m) = 0$ she does nothing. If $b_l^A(m) = 1$ she *rewinds* j^A to the value it had at the beginning of $w\left[b_l(m)\right]$ and deletes the corresponding values from $\hat{\boldsymbol{\pi}}_A$. She also sets all the bits of $w\left[b_l(m)\right]$ (in her uncoded simulation vector) to zero, so they will not be redetected as errors in the future. Bob does the same with the appropriate replacements.

$b_l^A(m)$ and $b_l^B(m)$ are computed as follows for every $w\left[b_l^A(m)\right]$ and $w\left[b_l^B(m)\right]$:

1) Alice and Bob randomly draw a hash function according to some shared random string. The input to the hash functions are the bits in the respective rewind window. The cardinality of the hash alphabet is set to be $|\mathcal{B}| = k^2$ for all $1 \leq l \leq L$.
2) Alice calculates the value of the hash according to $w\left[b_l^A(m)\right]$. The hash value is represented by $\lceil 2\log k \rceil$ bits that are sent over the channel using $a + 2l$ repetitions for every bit.
3) Bob decodes the value of the hash bits sent from Alice using a majority vote and compares them to their value calculated on his respective rewind window $w\left[b_l^B(m)\right]$. He sets $b_l^B(m) = 0$ if all decoded hash bits agree, and $b_l^B(m) = 1$ otherwise. Then he sends $b_l^B(m)$ to Alice using $a + 2l$ repetitions.
4) Alice decodes by a majority vote and obtains $b_l^A(m)$.

V. Analysis of the Coding Scheme

We denote by $j^A(T)$ and $j^B(T)$ the final values of Alice's and Bob's cursors, respectively. We assume that if any of the cursor exceeds n, the protocol is extended by zeros. The simulation is successful if both of these conditions are satisfied: (i) $j^A(T) \geq n$ (ii) $\hat{\boldsymbol{\pi}}_A^n = \hat{\boldsymbol{\pi}}_B^n = \boldsymbol{\pi}$. The purpose of the following analysis is to bound the rate of the scheme for which conditions (i) and (ii) hold with high probability.

Let us start by analyzing $j^A(T)$. We recall that by construction of the scheme, $b_l^A(m) = 1$ will rewind j^A to the value it had at the beginning of the rewind window. This means that j^A will be reduced by at most k^l, so:

$$j^A(T) \geq T - \sum_{l=1}^{L} \sum_{m=1}^{k^{L-l}} b_l^A(m) k^l = T\left(1 - \sum_{l}^{L} \overline{b_l}\right) \tag{3}$$

where

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

denotes the average number of non-zero rewind bits at level l. We note that at level l, all the rewind bits are *independently and identically distributed* (i.i.d), as they are determined only by the channel's symmetric noise and the independently generated hashes. So we denote

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

Taking the expectation over (3) yields

$$\mathbb{E} j^A(T) \geq T\left(1 - \sum_{l=1}^{L} P_{b_l}\right). \tag{4}$$

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}:

[1] It will become apparent in the sequel, that for an even k, by construction of the protocol both j^A and j^B are even when i is even, and odd when i is odd.

Lemma 3.

$$P_{el} \leq k^{-l}\left(\varepsilon + k\alpha\beta^{a+2}\frac{1-(\beta^2 k)^l}{1-\beta^2 k}\right), \tag{5}$$

where $\beta \triangleq 2\sqrt{\varepsilon(1-\varepsilon)}$ *and* $\alpha \triangleq \lceil 2\log k\rceil + 1$.

Proof. The key idea in the analysis is regarding the calculation of the rewind bits as a *layered* recursive process. Namely, we observe that by construction, a rewind window of level l comprises k rewind windows for level $l-1$. Having this notion we can write the following recursion formula:

$$P_{el} \leq k^{-2}kP_{el-1} + \alpha\beta^{a+2l}.$$

Namely, kP_{el-1} is the union bound over the error events of the previous level, and k^{-2} account for the probability of not detecting an error by Lemma 2. The $\alpha\beta^{a+2l}$ term is an upper bound for the probability of communication error (i.e. that at least one bit used for the calculation of the rewind bit was wrongfully decoded). It uses the union bound and Lemma 1. Setting the initial condition of the recursion to be $P_{e0} = \varepsilon$ and solving the recursion gives (5). □

We now bound P_{b_l} by the union bound over the probability of a detected error in level $l-1$ or a communication error in the calculation of the rewind bit. We also generously assume that every communication error causes Alice's rewind bit to be one. All in all:

$$P_{b_l} \leq (1-k^{-2})kP_{el-1} + \alpha\beta^{a+2l} \leq kP_{el-1} + \alpha\beta^{a+2l}. \tag{6}$$

where $(1-k^{-2})$ accounts for the probability that the mismatch in the rewind windows was detected. Plugging (5) in (6) we obtain $P_{b_l} \leq \overline{P}_{b_l}$:

$$\overline{P}_{b_l} = k^{2-l}\left(\varepsilon + k\alpha\beta^{a+2}\frac{1-(\beta^2 k)^{l-1}}{1-\beta^2 k}\right) + \alpha\beta^{a+2l} \tag{7}$$

We can now bound $\mathbb{E}j^A(T)$ using (4), (7) and the fact that $k\beta^2 < 2\varepsilon k < 1$:

$$\begin{aligned}\mathbb{E}j^A(T) &\geq T\left(1-\sum_{l=1}^{L}P_{b_l}\right) \geq T\left(1-\sum_{l=1}^{\infty}\overline{P}_{b_l}\right) \\ &\geq T\left(1-\tfrac{k^2}{k-1}\left(\varepsilon+\tfrac{\alpha k\beta^{2+a}}{1-\beta^2 k}\right) - \tfrac{\alpha\beta^{2+a}}{1-\beta^2}\left(1-\tfrac{k^2}{1-k\beta^2}\right)\right)\end{aligned} \tag{8}$$

We are now ready to set the relation between n and T :

$$T = \frac{n}{\left(1-\frac{k^2}{k-1}\left(\varepsilon+\frac{\alpha k\beta^{2+a}}{1-\beta^2 k}\right) - \frac{\alpha\beta^{2+a}}{1-\beta^2}\left(1-\frac{k^2}{1-k\beta^2}\right) - \xi\right)}. \tag{9}$$

Condition (i) now holds from the following lemma:

Lemma 4. *For any* $\xi > 0$ *and* T *set according to* (9)*:*

$$\lim_{n\to\infty}\Pr\left(j^A(T) \geq n\right) = 1.$$

This result follows since $\mathbb{E}j^A(T) \geq n$ (by (8) and (9)) and by using standard concentration techniques. The details are omitted due to lack of space.

Condition (ii) is validated in the following lemma

Corollary 1.

$$\lim_{n\to\infty} P_e = 0.$$

Proof. Recall that $P_e \leq P_{eL}$. By Lemma 3, $P_{eL} = O(k^{-L}) = O(T^{-1})$. Now observe that by (9) $n \to \infty$ implies that $T \to \infty$. □

We are now ready to calculate the rate of the coding scheme. N is the total number of channel uses consumed by the coding scheme Σ including the overhead required for the calculation of the rewind bits:

$$\begin{aligned} N &= T + \sum_{l=1}^{L}(a+2l)(2\log k + 1)k^{L-l} \\ &\leq T\left(1+\sum_{l=1}^{\infty}(a+2l)(2\log k+1)k^{-l}\right) \\ &= T\left(1+\alpha\left(\frac{a}{k-1}+\frac{2k}{(k-1)^2}\right)\right). \end{aligned} \tag{10}$$

Using (9), (10) and the rate definition $R = n/N$ yields (2) and concludes the proof of Theorem 3.

Acknowledgment

We thank the anonymous reviewers for their helpful comments that have significantly improved the presentation of the paper.

References

[1] 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.

[2] 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.

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

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

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

[6] R. G. Gallager, *Information Theory and Reliable Communication*. New York: John Wiley & Sons, 1968.