Electronic Colloquium on Computational Complexity, Report No. 95 (2017)

Constant-Rate Interactive Coding Is Impossible, Even In Constant-Degree Networks*

Ran Gelles†
Bar-Ilan University
ran.gelles@biu.ac.il

Yael T. Kalai
Microsoft Research
yael@microsoft.com

Abstract

Multiparty interactive coding allows a network of n parties to perform distributed computations when the communication channels suffer from noise. Previous results (Rajagopalan and Schulman, STOC '94) obtained a multiparty interactive coding protocol, resilient to random noise, with a blowup of $O(\log(\Delta+1))$ for networks whose topology has a maximal degree Δ. Vitally, the communication model in their work forces all the parties to send one message at every round of the protocol, even if they have nothing to send.

We re-examine the question of multiparty interactive coding, lifting the requirement that forces all the parties to communicate at each and every round. We use the recently developed information-theoretic machinery of Braverman et al. (STOC '16) to show that if the network's topology is a cycle, then there is a specific "cycle task" for which any coding scheme has a communication blowup of $\Omega(\log n)$. This is quite surprising since the cycle has a maximal degree of $\Delta = 2$, implying a coding with a *constant blowup* when all parties are forced to speak at all rounds.

We complement our lower bound with a matching coding scheme for the "cycle task" that has a communication blowup of $\Theta(\log n)$. This makes our lower bound for the cycle task tight.

*A preliminary version of this work appeared in the proceedings of the 8th Innovations in Theoretical Computer Science (ITCS'17) conference.

†Part of this work was done while at Princeton University. Supported in part by NSF grant CCF-1149888.

ISSN 1433-8092

1 Introduction

In multiparty interactive communication, n parties, connected via some arbitrary network $G = (V, E)$, try to compute some function f of their private inputs by communicating messages over the network. *Coding for interactive communication* asks for coding schemes that succeed to compute any such function even when the communication may be noisy. A fundamental question in this field is finding the maximal *rate* such coding schemes can achieve[1], that is, what is the minimal amount of redundancy coding schemes must add in order to successfully compute any function f despite the noise.

The work of Rajagopalan and Schulman [RS94] gave an initial answer to this question, assuming stochastic noise (e.g., when each bit is being flipped independently with some fixed probability $\varepsilon < 1/2$): Let Δ be the maximal degree in G, then any (noiseless) protocol χ can be simulated with high probability over the noisy network by a protocol χ' with communication complexity $\mathsf{CC}(\chi') = \mathsf{CC}(\chi) \cdot O(\log(\Delta + 1))$. That is, for constant-degree networks such as the line or the cycle, the rate, $\mathsf{CC}(\chi)/\mathsf{CC}(\chi')$, is a constant bounded away from 0 while for highly-connected graphs such as the star or the complete graph, the rate goes to zero when n tends to infinity, i.e., the rate is $\Theta(1/\log n)$. The work of Alon et al. [ABE+16] shows that coding schemes with constant (non-zero) rate also exist for the complete graph, and other highly-connected graphs, hinting that it may be possible to achieve a constant rate coding scheme for any network G. This hope was terminated by Braverman et al. [BEGH16], showing that a rate of $\Theta(\log\log n/\log n)$ is maximal for a specific task over the star network.

All the above works assume that the communication over the network is performed in rounds, where at every round *all the parties "speak"*, that is, $2|E|$ symbols are being communicated—one symbol over each channel of the network. A natural question to ask is: Why is such an assumption justifiable? One interpretation is that these previous works try to optimize the *round complexity*, as opposed to the communication complexity, hence the assumption that all parties send a message to all other connected parties in each round.

In this work, our goal is to optimize the *communication complexity* (as opposed to round complexity), and we ask whether similar bounds on the rate follow if we don't force all parties to speak at every round.

1.1 Our Results

Surprisingly, we show that the rate of coding schemes when G is a cycle (assuming channels with large alphabets) is at most $O(1/\log n)$. This corresponds to a lower bound of $\Omega(\log n)$ on the communication blowup. Informally, our main theorem is the following:

Theorem 1.1 (main, informal)**.** *Let G be the cycle graph with n parties. Then, for any constant $\varepsilon < 1/2$ there exists a task whose communication complexity over the noiseless G is d while any coding scheme over any noisy graph G' (with noise parameter ε) that succeeds with high probability has communication complexity $\Omega_\varepsilon(d \log n)$.*

The above theorem is quite surprising in light of the result of Rajagopalan and Schulman [RS94]: the maximal degree in the cycle graph is $\Delta = 2$, therefore the coding scheme of [RS94] (in the "everybody speaks" model) has a constant rate which is independent of the size of the network! In hindsight, the reason for this discrepancy is simple: the fact that everybody speaks in the model of [RS94] implies an inherent blowup in the communication of $O(n)$, which allows the parties to overcome errors. Indeed, assume that the "relevant" information for computing the function f

[1]The rate of a coding scheme is the ratio between the communication of a protocol that performs over a noiseless network, to the communication of the coding scheme for the same task, over the noisy network.

progresses along the cycle: first p_1 sends a message to p_2 (while all the other parties have nothing to send in the meantime), only then p_2 has a message to send to p_3 and so on. While the "relevant" information is limited to a single edge on the network at any round, the fact that all the parties *must* speak at every round multiplies the effective communication by n both for the noiseless and noisy protocols, hence, it cancels out in the rate. On the other hand, this superfluous redundancy gives the parties the opportunity to correct previous errors in rounds where they are supposed to be idling if we weren't to force all the parties to speak at every round, and charge the parties according to the communication that actually happened.

For our lower bound we don't restrict the topology G' of the noisy graph, and allow any party to communicate with any other party (since anyways we count the actual communication, allowing the coding scheme to utilize any underlying graph just makes our lower bound stronger). Our lower bound actually works when the noise erases symbols instead of corrupting symbols (again, making the result stronger). The only "restrictive" assumption we have on the coding scheme is a fixed speaking order, independent of the inputs and the noise; see the "Communication Model" paragraph for a discussion regarding this assumption.

The Cycle Task The noiseless task we use for Theorem 1.1 is an analog of the "pointer jumping" task over a cycle (see formal description in Section 2.1). Every party begins with a 2^n-ary tree of depth d, where each edge is labeled by a single bit. Each party begins at the root of its own tree, and the goal is to travel down the tree until it reaches a leaf.

It is most convenient to describe this task via the protocol that solves it. The parties are activated in a cyclic order (first p_1, then p_2, etc.). When p_i is activated, it receives a message of the form $(b_1, \ldots, b_n)$ from p_{i-1}, corresponding to the labels of the edges traversed by the parties in the previous n rounds (padding with zeros as necessary in the first $n-1$ rounds). Upon receiving this message $\ell = (b_1, \ldots, b_n)$ from p_{i-1}, p_i moves down from its current node to its ℓ-th child. Denoting by b the label of the edge it just took, p_i communicates to p_{i+1} the string $(b_2, \ldots, b_n, b)$. This process continues until all parties reach a leaf at depth d in their input tree. The output is the path each party took along its tree.

In addition to the lower bound on the communication blowup, we show a coding scheme that successfully computes the cycle task over a noisy network with rate $\Theta(1/\log n)$, matching the rate of our lower bound for the cycle task (up to a constant).

Theorem 1.2 (upper bound, informal)**.** *For any constant $\varepsilon < 1/2$, there exists a coding scheme that solves the cycle task of depth d over noisy channels with large alphabet and error parameter ε. The coding scheme obtains a rate of $\Theta_\varepsilon(1/\log n)$ and a success probability of $1 - 2^{-\Omega_\varepsilon(d \log n)}$.*

Communication Model For our communication model, we assume that protocols have a *fixed* order of speaking. That is, we can assume that the protocol works in rounds so that the party that speaks at round i is determined in advance, independently of inputs and noise. This assumption is not without loss of generality, but we claim here that lifting this assumptions trivializes the model.

A completely unrestricted model would let the parties determine, at any round, whether they speak or not (cf. adaptive protocols for the two party case [AGS16]; see also [GHS14]). Such a model trivializes coding in the multiparty scenario, as parties can "encode" information via the path that the message is sent through: say p_1 wants to send a single bit to p_2. If the bit is 0, then p_1 sends the message directly. If it is 1, it can send the bit through p_n (who will relay it to p_2). Now, even if noise occurs[2], p_2 can figure out the bit in certainty by the identity of the sender.

[2] As long as we do not allow a stronger type of noise, i.e., insertions and deletions, see [BGMO16].

Another model, which is not completely unrestricted but still trivializes coding in our scenario, is described in [JKL15]. There, parties are allowed to decide whether to send a message or not (and to whom) according to the transcript so far. On its surface, this restriction avoids the "path encoding" described above, as parties are not allowed to change the delivery path according to their inputs. Nevertheless, such a model still enables error correction via "path selection", since the transcript depends on the both the inputs and the noise. To give a simple example, assume a noiseless protocol in which the parties speak in order (p_1 sends a bit to p_2, then p_2 sends a bit to p_3, and so on). Such a protocol can be easily simulated over a noisy network in the [JKL15] model: After p_i sends a bit to p_{i+1} the latter sends the bit back either directly (if it was a 0), or through p_{i-1} (if it was a 1); note that this decision is made as a function of the observed (possibly noisy) transcript, and thus it is allowed in that model. Now p_1 knows if its original bit reached p_{i+1} correctly or not and either retransmits the bit, or sends a message to p_{i+2} (who forwards it to p_{i+1}) to indicate that the bit was transferred correctly, and the simulation can move on to simulating the next bit of the noiseless protocol. In other words, this model reduces bit flips into erasures, and performing error correction from erasures with rate $1 - \varepsilon$ is fairly simple if the model allows changing the order of the speaking according to the observed noise.

To conclude, we show that there is a strong relation between the order of speaking and the obtained coding rate. On one hand, allowing the order of speaking to change adaptively, allows trivial coding schemes. On the other hand, fixing the order of speaking allows us to show an $\Omega(\log n)$ lower bound on the blowup for the cycle task. It is however possible that worse rates are possible for other tasks. In fact, we conjecture that the blowup can get as high as $\Omega(n)$ in specific situations, as a function of the "mismatch" between the order of speaking in the noiseless protocol and the coding scheme.

Conjecture 1.3. *There exists a topology G and a noiseless protocol χ with a fixed order of speaking for which any coding scheme χ' with a fixed order of speaking has a rate of at most $O(1/n)$.*

Our findings are reminiscent of the two-party case: if the simulation has a fixed order, the order of speaking in the original scheme determines the maximal rate of the coding; specifically, it is conjectured that there exists a protocol whose simulation has rate bounded away from 1. On the other hand, if the simulation is allowed to be adaptive, better rates (that approach 1) can be achieved. See discussion in [KR13, Hae14].

On Binary vs. Large Alphabet While our main result (Theorem 1.1) assumes that the parties communicate symbols from a large alphabet, we also obtain a lower-bound for the case where the parties communicate bits, i.e., use a binary alphabet. Typically, constructing coding schemes over the binary alphabet is harder than constructing such schemes over a large alphabet. However, our result is a lower-bound rather than a coding scheme, and it is not necessarily so that the binary-case is stronger (nor is more difficult to obtain).

Nevertheless, the setting of binary channels and the setting of large-alphabet channels seem *incomparable*, since the alphabet constraint applies both to the original (noiseless) protocol and to the coded (noisy) protocol. We elaborate on this topic in Section 9.

We extend our lower bound result also to the case where the noiseless protocol and the coding scheme are binary. Specifically, we show a lower bound of $\tilde{\Omega}(\log n)$ on the blowup of the communication for binary coding scheme over the star network (where the $\tilde{\Omega}$ notation means neglecting $\log \log n$ terms). Informally, the theorem is the following.

Theorem 1.4 (binary case, informal)**.** *Let G be the star graph with n parties. Then, for any constant $\varepsilon < 1/2$ there exists a task whose communication complexity over the noiseless G is d while*

any coding scheme (with fixed order) over any noisy graph G' (with noise parameter ε) that succeeds with high probability has communication complexity $\tilde{\Omega}_\varepsilon(d \log n)$.

We stress that the above theorem is incomparable to the result of [BEGH16]: in our model parties may speak in an arbitrary (but fixed) order and are not forced to speak at every round. The task in consideration is the generalized jumping pointer described in [BEGH16]. The proof of Theorem 1.4 follows by combining the techniques developed in this paper for the cycle task with the techniques of [BEGH16] in quite a straightforward way, and we omit the details here.

1.2 Overview of our Techniques

The proof of our lower bound uses techniques from [BEGH16] for bounding the progress of a coding scheme χ' in simulating a noiseless protocol χ. As in [BEGH16], we use the notion of *cutoff* (Definition 4.1) that measures for any partial transcript of χ', how many cycles of the noiseless protocol χ are still not-simulated: when the cutoff is k, then the last $d-k$ cycles of χ are not simulated by the given transcript. More accurately (but still very informally) the transcript gives very little of information about the labels $\{b_i\}$ of the last $d-k$ cycles.

We show that any coding scheme that solves the cycle task with high probability must produce transcripts whose cutoff is $\approx d$, in expectation. Then, we show that for any segment in which the coding scheme communicates $O(n \log n)$ symbols, the cutoff advances by at most $O(1)$ cycles in expectation. Namely, let π be some fixed previous communication (including erasures), and let Π^{new} be the random variable describing the next $O(n \log n)$ symbols communicated by the coding scheme χ' (including erasures), then

$$\mathbb{E}[\mathsf{cutoff}(\pi \circ \Pi^{new}) \mid \mathsf{cutoff}(\pi) = k] \leq k + O(1).$$

In order for χ' to achieve an expected cutoff of $\approx d$, which is crucial for being correct with high probability, the coding scheme must communicate at least $\Omega(dn \log n)$ symbols, yielding a rate of $O(1/\log n)$.

The reason for the restricted progress in the cutoff is that many parties do not send any useful information in the segment Π^{new}, and that the next "move" (in the input tree) of each party depends on the moves of *all* the parties in the previous cycle. This means that most parties are missing a lot of crucial information in order to advance more than a constant number of levels in their input tree. Bounding the exact information sent by the parties (and thus the expected increase in the cutoff) is performed via the machinery of [BEGH16].

Showing that many parties give no information in any segment of $O(n \log n)$ rounds in our setting is a main technical difference from [BEGH16]. In the model of [BEGH16] all parties speak at every round, thus when the coding scheme communicates $O(n \log n)$ symbols we know that this communication is evenly spread—every party communicates exactly $O(\log n)$ symbols. In our setting, it is possible that the communication is evenly spread, but it is also possible that all $O(n \log n)$ symbols are communicated by a single party (or any other pattern in between). In the latter case, even if the noise targets the single party that speaks, that party could still convey $O(n \log n)$ bits of information by encoding its message via a standard error-correction code. Nevertheless, we show that there is a large set of parties that do not communicate any information in the new segment: either they don't speak at all, or they speak very little and their entire communication is completely erased by the noise. Furthermore, previous communication of these parties contains very little information on their labels in the last $d-k$ cycles to begin with.

The existence of this set of "erased" parties implies that the non-erased parties in this segment don't know how to proceed in their input tree, and their communication in that segment is "irrelevant"

to the progress of the protocol, even if it is not erased by the noise. Indeed, assume a party's current node in its input tree is given, and assume that the party doesn't know which of its children it should go to next. The best that a party can do is to send all the labels below its current node. However, due to the fact that each node has 2^n children, that party cannot communicate more than $O(1)$ levels below its current node even if it gets to speak all the $O(n \log n)$ symbols in the next segment.

Naturally, the actual proof is more complex, since the party has some prior information about the children it should go (due to communication in previous rounds). This means that the children are not equiprobable and the party can communicate more information about (the labels of) more probable children. Still, since the arity of the input tree is so large and since the information on the next children it should take is rather little, the party will be able to communicate information on the labels of only $O(1)$ levels below its current node (in expectation).

1.3 Other Related Work

The field of coding for interactive communication was initiated by Schulman [Sch92, Sch96] who formalized the question for the two-party case and developed basic techniques used for solving this task, either when the noise is stochastic (where each bit is flipped with some constant probability) or adversarial (where any subset of up to ε-fraction of the bits can be flipped). Later works in the two-party setting improve on the computational efficiency, success probability, and achievable rate of coding schemes. We refer the reader to [Gel15] for a survey on interactive coding.

As mentioned above, the interactive coding in the multiparty case was initiated by Rajagopalan and Schulman [RS94] for the random noise case. Efficiency for this setting is obtained by Gelles, Moitra and Sahai [GMS14]. The works of Alon et al. [ABE+16] and of Braverman et al. [BEGH16] identify the maximal rate obtainable over the complete graph and the star (and provide efficient schemes that obtain such a rate).

The work of Gallager [Gal88] considers the case where all the parties share a noisy *broadcast channel*, and show a coding scheme with blowup of $\Theta(\log \log n)$ for the task where each party begins with a single input bit and needs to learn all the input bits of all the other parties. Goyal, Kindler, and Saks [GKS08] prove that a blowup of $\Theta(\log \log n)$ is tight for the case of learning all parties' input bit over a noisy broadcast channel, i.e., they prove a blowup lower bound that matches the blowup of Gallager's scheme.

The case of multiparty interactive coding assuming *adversarial* noise is considered by Jain, Kalai and Lewko [JKL15] providing a coding scheme for topologies that have a star as a subgraph, that withstands $O(1/n)$-fraction of adversarial noise and blows up the communication by only a constant. The work of Hoza and Schulman [HS16] provides a coding scheme for any topology $G = (V, E)$ that withstands $O(1/n)$-fraction of noise and obtains a rate of $\Theta(n/|E| \log n)$.

1.4 Paper Outline

We begin by defining the setting and the communication task we wish to solve (Section 2). In Section 3 we give our upper bound, showing a coding scheme for the cycle task with blowup $O(\log n)$. The more complex lower bound is presented in Sections 4–8: In Section 4 we define the notion of cutoff and prove that any successful coding protocol must produce a transcript with a large cutoff value. In Section 5 we formalize the claim that the cutoff increases by only a constant for every $O(n \log n)$ noisy transmissions of any given coding scheme; this implies the lower bound. The detailed proof of this claim is given in sections 6–8; technical preliminaries and proofs of several technical lemmas appear in Appendices A and B, respectively.

2 Preliminaries: Notations, Model, Coding Schemes

Notations For $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, 2, \ldots, n\}$. The log() function is taken to base 2. For two strings a, b we denote their concatenation by $a \circ b$.

Given any tree $\mathcal{T}$ of depth N, we denote its first k levels by $\mathcal{T}^{\leq k}$ and its $N-k$ last levels by $\mathcal{T}^{>k}$. Given a path $z = (e_1, e_2, \ldots)$, we denote by $\mathcal{T}[z]$ the subtree of $\mathcal{T}$ rooted at the end of the path that begins at the root of $\mathcal{T}$ and follows the edge-sequence z. The above notation composes for sets of trees, e.g., if $\vec{\mathcal{T}} = (\mathcal{T}_1, \mathcal{T}_2, ...)$ is an array of trees and $\vec{z} = (z_1, z_2, \ldots)$ is an array of paths, then we let $\vec{\mathcal{T}}^{\leq k}$ denote the array $(\mathcal{T}_1^{\leq k}, \mathcal{T}_2^{\leq k}, ...)$ and $\vec{\mathcal{T}}[\vec{z}]$ the array $(\mathcal{T}_1[z_1], \mathcal{T}_2[z_2], ...)$, etc.

As a rule, we use small letters to denote specific values (e.g., the input x_i given to party i), and capital letters to denote the corresponding random variables (i.e., X_i for the random variable describing the input of the i-th party, when the inputs are drawn from some given distribution).

Multiparty Interactive Communication and Protocols We assume an undirected network $G = (V, E)$ of $n = |V|$ parties, $p_1, \ldots, p_n$, where p_i is connected to p_j if and only if $(i, j) \in E$. Each party is given an input x_i, and is assumed to output $f_i(x_1, \ldots, x_n)$ at the end of the protocol.

A *protocol* dictates to each party what is the next symbol to send and over which channel, given the party's input, the round number, and all the symbols that the party has received so far. After a fixed and predetermined number of rounds, the protocol terminates and each party outputs a value as a function of its input and observed transcript. We assume that the order of speaking is *fixed* and is independent of the party's inputs and the noise. That is, it is determined in advance which channel is utilized at each round.

Noisy and Noiseless Networks For the noiseless network, we focus on the *cycle network*. In the cycle, each party p_i is connected to p_{i-1} and p_{i+1} (all indices are modulo n).

For showing lower bounds over the noisy network we allow the parties to utilize the complete graph, avoiding any limitation on the protocol (since limiting the connectivity may harm the rate artificially). For our upper bound (coding scheme) the underlying topology is still the complete graph, however, the specific scheme we show does not need to communicate over all possible links—it communicates only over the cycle subgraph.

In a noisy network, each channel is assumed to suffer from random noise. For our lower bound we will assume each channel is a large-alphabet *erasure channel* EC_ε with erasure probability ε.

Definition 2.1. *For $\varepsilon \in [0, 1]$ and a finite set Σ, the erasure channel over alphabet Σ is a random function $\mathsf{EC}_\varepsilon : \Sigma \to \Sigma \cup \{\perp\}$ which turns each input symbol into an erasure mark $\perp$ with probability ε, or otherwise keeps the symbol intact. When a channel is accessed multiple times, each instance is independent.*

When considering upper bounds (coding schemes), channels with random noise are too weak (i.e., they can be reduced to erasure channels with high probability). Therefore, for our coding scheme we will assume a stronger type of noise we name semi-adversarial. Here, the transmissions that will be corrupted are determined in a random manner, however the received symbol of a corrupted transmission is determined *adversarially*; see discussion in Section 3.

Definition 2.2. *For $\varepsilon \in [0, 1]$ and a finite set Σ, the semi-adversarial noisy channel over alphabet Σ is a random function $\mathsf{SAC}_\varepsilon : \Sigma \to \Sigma$ which corrupts any input symbol with probability ε, independently per instance. Once a symbol is corrupted, it may turn into any symbol in Σ, determined adversarially by the channel (possibly, all the corrupted symbols are chosen in a dependent manner).*

Communication Complexity For any protocol χ communicating symbols from an alphabet Σ, denote by $|\chi|$ the maximum number of symbols communicated by any execution of χ. Since we assume the order of speaking is fixed regardless of the inputs (and noise), each execution of χ has exactly $|\chi|$ number of symbols communicated. We define the communication complexity of χ, denoted by $\mathsf{CC}(\chi)$, by

$$\mathsf{CC}(\chi) = |\chi| \cdot \log |\Sigma|.$$

2.1 The Cycle Task: Problem Statement

In this section we define the cycle task and discuss a simple protocol that solves it over the noiseless cycle network.

Recall we have n parties $\{p_1, \ldots, p_n\}$ where each p_i receives the input x_i. We assume each input x_i is a labeled $|\Sigma|$-ary tree of depth d, where $\Sigma = \{0,1\}^n$ and each edge in the tree is labeled by a single bit.

The output of p_i is a simple root-to-leaf path (of length d) denoted by path_i, and the complete task output is denoted by $\mathsf{path} = (\mathsf{path}_1, \ldots, \mathsf{path}_n)$. We define the output in an inductive manner. For $i \in [n]$ and $j \in [d]$, let $\mathsf{path}_i(j) \in \{0,1\}^n$ denote the (index of the) j-th edge of path_i. Moreover, let $b_i(j) \in \{0,1\}$ denote the label of the edge that corresponds to $\mathsf{path}_i(j)$. For the induction basis, assume $b_i(j) = 0$ for all $i \in [n]$ and $j \leq 0$.

For $j \geq 1$, and for $i \in [n]$ we define $\mathsf{path}_i(j)$ as a function of $\{\mathsf{path}_{i'}(j')\}_{(j',i')<(j,i)}$, where $(x,y) < (u,v)$ holds if $x < u$ or if both $x = u$ and $y < v$; note that this implies a total order on pairs (j,i). The value of $\mathsf{path}_i(j)$ is given by the labels $b_{i'}(j')$ for the $n-1$ pairs (j',i') preceding (j,i) according to the total order we defined. Namely,

$$\mathsf{path}_i(j) = (b_{i+1}(j-1), b_{i+2}(j-1) \ldots, b_n(j-1), b_1(j), \ldots, b_{i-2}(j), b_{i-1}(j)).$$

Note that the cycle task can be solved by a simple protocol as described in Section 1. The protocol works in "cycles" where each such cycle means repeating the following process for $p_1, p_2, \ldots, p_n$ in order. During the j-th cycle p_i sends to p_{i+1} the value of $\mathsf{path}_i(j)$ along with the label $b_i(j)$ of the edge it just took. Now p_{i+1} can infer the value of $\mathsf{path}_{i+1}(j)$, and obtain the bit $b_{i+1}(j)$ labeling that edge in its input x_{i+1}. It follows that after d such "cycles" all parties reach a leaf at level d in their input, and can output path_i. Assuming the parties communicate symbols from Σ, the protocol communicates dn symbols[3] and has a communication complexity of dn^2 bits. It can be verified that the communication complexity of solving the cycle task is $\Theta(dn^2)$.

For our lower bound, we assume that the inputs $X = (X_1, \ldots, X_n)$ are sampled so that each label is uniform in $\{0,1\}$. We are looking for coding schemes that solve the above task with high probability over the inputs X, the noise and the randomness of the coding scheme.

3 Upper Bound: A coding scheme with blowup $\Theta(\log n)$ for the cycle task

In this section we provide a coding scheme for the cycle task that achieves a communication blowup of $\Theta(\log n)$ with respect to the communication complexity of solving the cycle task over noiseless channels. The key idea behind our scheme is repeating each communicated symbol $\Theta(\log n)$ times. This in turn reduces the probability that the symbol is decoded incorrectly at the recipient to be polynomially small in the number of parties. Then, the event of an error is rare enough that standard

[3] In fact, it is enough to use $\Sigma = \{0,1\}^{n-1}$. We will neglect this issue as it doesn't change the asymptotic behaviour of the communication complexity, nor the asymptotic rate of related coding schemes.

interactive-coding techniques that recover from small amount of errors (e.g., [RS94]) succeed with overwhelming probability.

When considering random noise over large alphabet, notice that the analog of the binary-symmetric-channel—a channel that uniformly picks the corrupted symbol—is too weak. Indeed, the parties could use only a small fraction of the symbol space in order to "catch" errors with high probability, thus essentially reducing the noise model into the case of erasures, while keeping the asymptotic rate the same up to a constant (see, for instance, the blueberry code technique in [FGOS15]).

Hence, our upper bound is defined in the somewhat stronger noise-model, which we call *semi-adversarial*, formally defined in Definition 2.2. In this noise model, each symbol is corrupted with probability ε, independently across different symbols. However, once a symbol is corrupted, the output symbol of the channel is chosen *adversarially*, in a worst case manner.

Our main theorem for this section is the following

Theorem 3.1. *For any $\varepsilon < 1/2$, there exists a coding scheme with fixed order that solves the cycle task over a noisy network where each communication channel is a SAC_ε, with rate $\Theta_\varepsilon(1/\log n)$ and success probability at least $1 - 2^{-\Omega_\varepsilon(d \log n)}$.*

3.1 Coding Scheme Construction

The construction of our coding scheme utilizes a primitive known as tree codes (see [Sch96]; also see [Gel15]). Let us first recall Hamming distance.

Definition 3.2. *The Hamming distance $\Delta(s, s')$ of two strings $s = s_1 \dots s_m$ and $s' = s'_1 \dots s'_m$ of length m, is the number of positions i such that $s_i \neq s'_i$.*

A tree code is defined as follows.

Definition 3.3. *A β-ary tree code of depth γ, distance α and alphabet σ is a prefix code $\mathsf{TC} : [\beta]^{\leq\gamma} \to \sigma^{\leq\gamma}$ that satisfies the following. For any two strings $x, y \in [\beta]^\ell$ of the same length $\ell \leq \gamma$ whose first difference is at the i-th coordinate,*

$$\Delta(\mathsf{TC}(x), \mathsf{TC}(y)) \geq \alpha(\ell - i + 1),$$

where $\Delta(\cdot, \cdot)$ is the Hamming distance.

Schulman [Sch96] showed that infinite-depth tree codes exist, and described the tradeoff between their distance and arity, and their alphabet size.

Lemma 3.4 ([Sch96])**.** *For any fixed $\beta \in \mathbb{N}$ and $\alpha \in (0, 1)$, there exists a finite alphabet σ of size $|\sigma| = \beta^{O(1/(1-\alpha))}$ which suffices to construct a β-ary tree code with distance α and any depth.*

Our coding scheme is denoted by χ' and described in Algorithm 1. The scheme builds on tools from [RS94], and adapts them to our communication-model in which the parties are not forced to speak at every round. Let χ be the noiseless protocol for the cycle task described in Section 2.1. Our coding scheme χ' simulates χ step by step, sending each symbol that χ sends using two levels of coding (tree code TC and repetition code REP). The repetition code sends each symbol for $k = \Theta_\varepsilon(\log n)$ repetitions, so that decoding via a simple majority fails at the recipient with probability at most n^{-10}. The tree codes ensures that the recipient correctly decodes an increasing prefix of the communication. While we cannot guarantee that a party correctly decodes *all* the symbols sent to it so far, symbols that were sent earlier in the protocol will be decoded correctly

with an increasing probability. The party can then verify that symbols it has already sent during previous rounds are consistent with its current understanding of the decoded incoming transmissions. In case they are not, the party transmits a special $\mathcal{B}$ symbol whose meaning at the recipient is to "delete" the last (non-$\mathcal{B}$) symbol it has received. By sending multiple $\mathcal{B}$ symbols, the party can delete any incorrect suffix of its outgoing transmissions, until they become consistent with its (current view of its) incoming transmissions.

One special case can happen when a party has communicated a future symbol due to some decoding error (e.g., the other parties where going back), however, in a later round it finds out that the communicated symbol is indeed the correct one. In this case, the party sends a $\mathcal{H}$ symbol which means "hold": don't go back, but also no new symbol is added.

In the coding scheme χ' the parties communicate over channels with alphabet of size $(|\Sigma| + 2)$ that corresponds to all the symbols of χ and the additional "back" symbol $\mathcal{B}$ and "hold" symbol $\mathcal{H}$. We assume a tree code with input alphabet $O(\Sigma)$ (specifically, a $(|\Sigma| + 2)$-ary tree), distance $\alpha > \varepsilon$, and output alphabet of size $|\Sigma'| = |\Sigma|^{O_\varepsilon(1)}$. Such a tree code exists due to Lemma 3.4.

3.2 Coding Scheme Analysis

We now prove that the coding scheme χ' of Algorithm 1 solves the cycle task with high probability over noisy networks, and satisfies the conditions of Theorem 3.1.

Proof. (**Theorem 3.1.**) First, let us analyze the obtained rate. It is easy to verify that

$$\mathsf{CC}(\chi') = 100d \cdot n \cdot O_\varepsilon(\log n) \times \log |\Sigma'|.$$

Recall that $\mathsf{CC}(\chi) = \Theta(dn \log |\Sigma|)$ to get that the rate is

$$\frac{\mathsf{CC}(\chi)}{\mathsf{CC}(\chi')} = \frac{dn \log |\Sigma|}{100dn \cdot O_\varepsilon(\log n) \cdot O_\varepsilon(1) \log |\Sigma|} = \Theta_\varepsilon\left(\frac{1}{\log n}\right).$$

Next we show that χ' simulates χ with high probability. For any $j \in [d']$ let $\psi(j)$ be the potential of the protocol at the end of cycle j defined as follows. For party p_i, let $r'_i(j)$ and $s'_i(j)$ be the parsed incoming and outgoing messages of p_i up to cycle j of the simulation χ', respectively; set $T_i(j) = s'_i(j)$ be the simulated outgoing transcript of party p_i up to cycle j. This transcript can be compared to the correct transcript p_i sees in χ. Usually, T_i agrees with the correct transcript up to some point (this will be called the correct prefix), and possibly differ from the correct transcript beyond it (this will be the incorrect suffix). let $\psi^+(j)$ be minimal length of correct prefix across all the parties, and let $\psi^-(j)$ be length of the longest suffix some party p_u holds, where we begin to count the suffix length for p_u starting at position $\psi^+(j)$ in $T_u(j)$. The potential at the end of the j-th cycle of the simulation is then

$$\psi(j) = \psi^+(j) - \psi^-(j).$$

We denote by *k-error* the event that at least $\lceil k/2 \rceil$ repetitions of a single symbol transmission are corrupted, i.e., that the receiver decodes the repetition code to a different symbol than the one sent. The correctness of the coding scheme follows from the following two claims, relating the potential to the noise pattern observed throughout the coding scheme.

Claim 3.5. *After every* 3 *cycles in which all the parties decode the correct incoming messages, the potential increases by at least* 1*.*

Algorithm 1 The coding scheme χ' for the Cycle Task

Let χ be the noiseless protocol that solves the cycle task. In particular, for any $i \in [n]$ and any $r \in \Sigma^{\leq d}$ let $\chi_i(r)$ be the next symbol p_i sends if its incoming transcript is r. We assume $\chi_i(r) = 0$ for any r whose length is larger than d. Set $k = \Theta_\varepsilon(\log n)$.

1: Init: for all $i \in [n]$, $y_i, s_i \leftarrow \emptyset$.

2: **repeat** for $d' = 100d$ times
3: **for** $i = 1$ to n (sequentially) **do** for party p_i:
4: Let $\Upsilon \in (\Sigma')^k$ be the last k symbols received from p_{i-1}
5: $y_i \leftarrow y_i \circ \mathsf{REP}^{-1}(\Upsilon)$ ▷ Decode the repetition code and store the received symbol
6: $r \leftarrow \mathsf{TC}^{-1}(y_i)$ ▷ Decode y_i via the tree code
7: $r' \leftarrow \mathsf{Parse}(r)$

The function $\mathsf{Parse}(r)$ is defined in the following manner: Process the string r symbol-by-symbol in order. When processing a symbol from Σ, copy it to the output register. When processing a $\mathcal{H}$ symbol, ignore it and move on. When processing a $\mathcal{B}$ symbol, delete the last non-deleted symbol in the output register. For instance, the string '$a\mathcal{H}bd\mathcal{B}ccc\mathcal{B}d\mathcal{B}\mathcal{B}d$' is parsed to the string '$abcd$'.

▷ Check Consistency according to χ
8: $s' = \mathsf{Parse}(s_i)$
9: **if** $|s'| < |r'|$ and for all $j \leq |s'|$, $s'(j) = \chi_i(r'_1 \cdots r'_j)$ **then**
10: $\sigma \leftarrow \chi_i(r'_1, \ldots, r'_{|s'|+1})$ ▷ Consistent: the next symbol is set according to χ
11: **else if** $|s'| = |r|$ and for all $j \leq |s'|$, $s'(j) = \chi_i(r'_1 \cdots r'_j)$ **then**
12: $\sigma \leftarrow \mathcal{H}$ ▷ Consistent: correct next message was already sent; hold
13: **else**
14: $\sigma \leftarrow \mathcal{B}$ ▷ Inconsistent: send a "back" symbol
15: **end if**
16: $s_i \leftarrow s_i \circ \sigma$
17: Let $\tilde{\sigma}$ be the last symbol of $\mathsf{TC}(s_i)$ ▷ Encode σ via the tree code
18: Send $\mathsf{REP}(\tilde{\sigma})$ to p_{i+1} ▷ Encode $\tilde{\sigma}$ via a repetition code (k symbols)
19: **end for**
20: **end repeat**

Ramark: The first party p_1 behaves slightly different in order to initiate the cycle. We assume that p_1 begins with a "received" transcript that contains the symbol 0^d in a "hardcoded" manner. Specifically, for p_1 Line 7 in the above algorithm is always replaced with $r' \leftarrow 0^d \circ \mathsf{Parse}(r)$. Indeed, on the first step of the protocol p_1 receives no incoming messages, therefore $y_1 = \emptyset$ and $r' = 0^d \circ \emptyset$, so p_1 generates the first message $\sigma = \chi_1(0^d)$.

Proof. If all the parties posses correct transcripts T_i of the same length, i.e., $\psi^- = 0$, then it is clear that during the next cycle the potential increases by one.

If all T_i are correct, but they are not of the same length, parties with $|T_i| = \psi^+$ will add a correct symbol to T_i, while parties with $|T_i| > \psi^+$ will either remove a symbol, add a symbol or hold. In all cases ψ^+ increases by one while ψ^- either remains the same (if a symbol was added by a party with $|T_i| > \psi^+$) or otherwise decreases. Hence, in all cases, the potential ψ increases by at least 1 at the end of the cycle.

Now consider the case where at the end of the $(j-1)$-th cycle, some of the T_i's are incorrect. For this analysis assume that after the $(j+2)$-th cycle, the T_i's are still not all correct; we later consider the case that they become correct during these three cycles. Since we assume all decodings are correct in the next 3 cycles, for any $1 < i \leq n$ and for $j' \in \{j, j+1, j+2\}$ it holds that $r'_i(j') = s'_{i-1}(j')$ and for the first player we have $r'_1(j') = 0^d \circ s'_n(j'-1)$. Let P be the set of parties that find an inconsistency in their T_i during the $j+2$-th cycle, and let $p_k \in P$ be the first such party. Note that $P \neq \emptyset$ or otherwise all T_i's are correct before the end of the third cycle, which we assumed is not the case.

We distinguish three cases.

1. For any party $p_i \in P$: It is clear that P_i sends a $\mathcal{B}$ in the j-th cycle so it effectively removes a symbol from its T_i.

2. Consider $p_k, \ldots, p_n$: Denote by $m_{\geq k}(j') = max_{i \geq k}|T_i(j')|$ the maximal length of T_i of the parties $p_k, \ldots, p_n$ at the end of cycle j'. We claim that $m_{\geq k}(j) < m_{\geq k}(j-1)$. Let p_i be the first party (with $i \geq k$) whose T_i is maximal, $|T_i| = m_{\geq i}(j-1)$. It is clear that this party removes one symbol from its T_i: either it is p_k who removes a symbols due to item 1 above, or otherwise, it holds that $T_{i-1}(j-1)$ is *not* maximal. In this case, note that $T_{i-1}(j)$ cannot be maximal (i.e., it cannot be that p_{i-1} added a symbol to its transcript on the j-th cycle). This holds since all the parties before p_{i-1} (up to p_k) have a transcript's length strictly less than $m_{\geq k}(j-1)$ when it is $p_{i'-1}$ turn. hence $T_{i-1} < T_i$ and due to the correct decoding we have $r'_i < s'_i$ so p_i removes a symbol from its T_i (Line 14). A similar argument holds for the next party that holds a transcript of maximal length, and so on. Hence, $m_{\geq k}(j) < m_{\geq k}(j-1)$.

   The same argument applies to all three cycles, and we have $m_{\geq k}(j+2) \leq m_{\geq k}(j-1) - 3$.

3. Now consider $p_1, \ldots, p_{k-1}$. Similar to the above case, let $m_{<k}(j') = \max_{i<k} |T_i(j')|$ be the maximal length of T_i of the parties $p_1, \ldots, p_{k-1}$ at the end of cycle j'. There are three cases here to consider.

   (a) The first case is when $T_1(j-1) \leq T_n(j-1)$ in which case p_1 potentially adds a symbol to its transcript, thus potentially $m_{<k}(j) = m_{<k}(j-1) + 1$

   (b) The second case is when $T_1(j-1) = T_n(j-1) + 1$ in which case p_1 potentially holds and it can be that $m_{<k}(j) = m_{<k}(j-1)$.

   (c) In the last case, it holds that $m_{<k}(j) < m_{<k}(j-1)$ by a similar argument to item 2 above.

   It follows that, in each of the 3 cycles, either $m_{<k}$ decreases by 1, or otherwise it must be that $m_{<k} \leq m_{\geq k} + 2$, that is, $m_{<k}$ is bounded by $m_{\geq k}$, maybe up to an additive constant of 2. Therefore, if we look at the maximal transcript length $m = \max(m_{<k}, m_{\geq k})$ of three consecutive cycles, we have that $m_{\geq k}$ decreases in each of these cycles, and $m_{<k}$ is bounded by $m_{\geq k} + 2$, then it must hold that $m(j+2) < m(j-1)$, which proves the claim.

Finally, in the above we assumed that at least one of T_i's remain incorrect at the end of the third cycle. If during the analysis above we reached a situation where all the T_i's are correct, it means that at this point ψ^- is 0 (which already implies the potential has increased), and from that point and on ψ^+ (and the potential) can only increase.

□

Claim 3.6. *During any given consecutive n steps of the for-loop (Line 3), the potential decreases by at most* 3*.*

Proof. Each of r'_i, s'_i either grows by at most one symbol (correct or not) or reduces by at most one symbol (correct or not), so the worst case is when ψ^+ decreases by one (since some party deleted a symbol from its correct prefix) and ψ^- increases by two (since some party added an incorrect symbol, and the 'correct prefix' is now shorter by 1 due to the change in ψ^+), thus ψ changes by at most 3. □

It follows from the above two claims, that the coding scheme is successful in simulating the first d cycles of χ as long as there are less than $8d$ cycles in which an incorrect decoding happened somewhere over the cycle—in that case the potential at the end of the coding scheme will be at least $\psi(d') \geq d$ implying that d cycles of χ are correctly simulated: A very conservative analysis shows that $8d$ cycles with incorrect decodings can reduce the potential by at most $8d \times (-3)$. Out of the other cycles (where there is no decoding error), there are at least $(100 - 3 \times 8)d = 76d$ cycles which are consecutive to another two cycles with no decoding error. Each triplet of cycles with no decoding error increases the potential by at least 1, so the potential at the end is at least

$$\psi(d') \geq 8d \times (-3) + \frac{76d}{3} \times (+1) > d.$$

We now bound the failure probability, i.e., the probability to have at least $8d$ cycles with incorrect decodings. For the i-th party, we can denote by $\ell_i(j)$ the magnitude of error at the j-th cycle $j \in [d']$. This is the length of the incorrect decoded suffix (so $\ell_i(j) = 0$ when the entire j-symbol word is correctly decoded). The tree code definition tells us that in order to cause a decoding error of magnitude ℓ at cycle j, at least $\alpha\ell/2$ k-errors must have happened in the symbols received at cycles $[j - \ell_i(j), j]$ for that same party; this follows directly from the Hamming distance constrains (Definition 3.3). However, note that intervals may overlap, thus the noise that causes different intervals may be dependent.

Let I_i be the set of intervals $I_i = \{[j - \ell_i(j), j] \mid j \in [d']\}$ corresponding to the error-intervals of the i-th party. Lemma 7 in [Sch96] shows it is always possible to find a subset $I'_i \subseteq I_i$ with non-overlapping intervals whose union is at least half the size of the union of I_i, that is $|\bigcup I'_i| \geq \frac{1}{2}|\bigcup I_i|$.[4] Since the intervals in I'_i are non-overlapping, the noise causing any two such intervals is independent.

Having at least $8d$ cycles with decoding error implies that $|\bigcup \left(\bigcup_{i\in[n]} I'_i\right)| > 4d$. Also note that any two intervals in $\bigcup_{i\in[n]} I'_i$ are independent in the sense that the noise causing one interval is independent of noise causing any other interval. It follows that at least $\frac{\alpha}{2} \cdot 4d$ corruptions (i.e., k-errors) must have happened during the protocol. Recall that the probability to have one or more k-errors in a given cycle is bounded by n^{-9}. Union bounding on all the possible noise patterns (along the $d' = 100d$ cycles), the probability for such a large amount of noise is bounded by

$$\binom{100d}{2\alpha d} \cdot (n^{-9})^{2\alpha d} \leq 2^{100d} \cdot 2^{-18\alpha d \log n} \leq 2^{-\Omega(d \log n)}.$$

□

[4] For a set of intervals I, the notation $|\bigcup I|$ denotes the length of the union of all the intervals in I.

4 The Cutoff of the Protocol and its Relation to the Success Probability

In this section we outline the proof of our lower bound. The reader may wish to refer to Appendix A and Appendix B for several preliminaries regarding information theory and several technical lemmas, that we will use in the following sections.

Following [BEGH16], we define the notion of *cutoff* which measures the progress a protocol has performed in simulating the cycle task, as a function of the (noisy) transcript communicated by the protocol. We show that the cutoff of a protocol is correlated with the length of the correct simulated output, in the sense that if the cutoff is k, it is improbable that the protocol gives an output whose correct prefix is of length more than k. Hence, if a protocol for the cycle task of depth d is correct with high probability, the implied cutoff must be high (i.e., around d).

Recall that x_i is the input of the i-th party, and X_i is the random variable describing it; similarly, π is used to describe a specific (observed) transcript while Π is the corresponding random variable. Also recall that the output of the i-th party is path_i describing the root-to-leaf path that the party traversed along x_i. Finally, recall that we denote by $\mathsf{path}_i(k)$ the first k edges in path_i and by $x_i[\mathsf{path}_i(k)]$ the subtree of x_i rooted at the end of $\mathsf{path}_i(k)$.

Definition 4.1 (Cutoff)**.** *For any transcript π, and any input $x = (x_1, \ldots, x_n)$, the* cutoff of the protocol*, denoted by $\mathsf{cutoff}(\pi, x)$, is the minimal number k, such that*

$$\sum_{i=1}^{n} I(X_i[\mathsf{path}_i(k)] \mid \Pi = \pi, \mathsf{PATH}(k) = \mathsf{path}(k)) \leq 0.01n. \tag{1}$$

We note that if $\mathsf{cutoff}(\pi, x) = k$ then for any x' such that $x'^{\leq k} = x^{\leq k}$, it holds that $\mathsf{cutoff}(\pi, x') = k$. Furthermore, the cutoff is only a function of the *path* up to level k, that is, if $\mathsf{cutoff}(\pi, x) = k$ then for any input x' that has the same $\mathsf{path}(k)$ it holds that $\mathsf{cutoff}(\pi, x') = k$; This property allows us to abuse notation and write $\mathsf{cutoff}(\pi, \mathsf{path}(k)) = k$, when the path is fixed but we do not care about the specific input.

The following proposition shows that in order for a protocol to output the correct value with high probability, the cutoff (given the complete transcript) must be $\approx d$. Hence, protocols that succeed with high probability must produce transcripts whose cutoff is large in expectation.

Proposition 4.2. *Fix a protocol that solves the cycle task of depth d over a network with n parties (with large enough n), that succeeds with probability at least $1/5$ on average, i.e., a protocol for which $\Pr_{X,\Pi}[\textit{correct output}] \geq 1/5$. Then,*

$$\mathbb{E}_{X,\Pi}[\mathsf{cutoff}(\Pi, X)] \geq \frac{d}{10}.$$

Proof. Recall that the event $\mathsf{cutoff}(\pi, x) = k$ depends only on π and $\mathsf{path}(k)$ and is independent of $x^{>k}$. In Claim 4.3 below we prove that if $\mathsf{cutoff}(\pi, \mathsf{path}(k)) = k$ for some $k < d$, then the protocol gives the correct output with only a small probability of 0.02. Therefore, conditioned on the event that $\mathsf{cutoff}(\Pi, X) < d$ the protocol outputs the correct value with probability at most 0.02, that is, $\Pr_{X,\Pi}[\text{correct output} \mid \mathsf{cutoff}(\Pi, X) < d] \leq 0.02$. Since the protocol is correct with probability $1/5$ on average over the inputs and randomness of the protocol (and the noise), the claim follows.

Indeed,

$$\begin{aligned}\frac{1}{5} &\leq \Pr_{X,\Pi}[\text{correct output}] \\ &= \Pr[\mathsf{cutoff}(\Pi, X) < d] \Pr[\text{correct output} \mid \mathsf{cutoff}(\Pi, X) < d] \\ &\quad + \Pr[\mathsf{cutoff}(\Pi, X) = d] \Pr[\text{correct output} \mid \mathsf{cutoff}(\Pi, X) = d] \\ &\leq \Pr[\mathsf{cutoff}(\Pi, X) < d] \cdot 0.02 + \Pr[\mathsf{cutoff}(\Pi, X) = d] \cdot 1,\end{aligned}$$

hence, $\Pr[\mathsf{cutoff}(\Pi, X) = d] \geq 1/5 - 0.02 = 0.18$ and $\mathbb{E}_{X,\Pi}[\mathsf{cutoff}(\Pi, X)] \geq 0.18 \cdot d$. □

Claim 4.3. *Given π, $k < d$, and $\mathsf{path}(k)$ such that $\mathsf{cutoff}(\pi, \mathsf{path}(k)) = k$,*

$$\Pr[\text{correct output} \mid \Pi = \pi, \mathsf{PATH}(k) = \mathsf{path}(k)] < 0.02.$$

Proof. Let $L = L_1, \ldots, L_n$ be the last (array of) edge(s) in the array of paths $\mathsf{PATH}(k+1) = \mathsf{PATH}_1(k+1), \ldots, \mathsf{PATH}_n(k+1)$; note that L is part of the output, specifically, $L_i \in \{0,1\}^n$ is part of the output of the i-th party.

Note that once we condition on $\mathsf{path}(k)$, the edge L_i is determined by $X_{[n]\setminus\{i\}}$ alone, thus the information about L_i conditioned on any event is bounded by the information on $X_{[n]\setminus\{i\}}$ below the cutoff level k, conditioned on the same event. We can therefore bound the probability that p_i correctly outputs L_i—it is at most the probability to guess this label given all its knowledge. We can assume that p_i learns the entire (corrupted) transcript, and we give it the path $\mathsf{path}(k)$ "for free". Note that the i-th party knows (in addition to its observed transcript, etc.) also its own input x_i.

$$\begin{aligned}2^{-H_\infty(L_i \mid \Pi=\pi, \mathsf{PATH}(k)=\mathsf{path}(k), X_i)} &\leq \frac{1 + I(L_i \mid \Pi = \pi, \mathsf{PATH}(k) = \mathsf{path}(k), X_i)}{|L_i|} && (2) \\ &\leq \frac{1}{n}\left(1 + I(X_{[n]\setminus\{i\}}[\mathsf{path}(k)] \mid \Pi = \pi, \mathsf{PATH}(k) = \mathsf{path}(k), X_i)\right) && (3) \\ &= \frac{1}{n}\left(1 + I(X_{[n]\setminus\{i\}}[\mathsf{path}(k)] \mid \Pi = \pi, \mathsf{PATH}(k) = \mathsf{path}(k))\right) && (4) \\ &= \frac{1}{n}\left(1 + \sum_{j \neq i} I(X_j[\mathsf{path}(k)] \mid \Pi = \pi, \mathsf{PATH}(k) = \mathsf{path}(k))\right) && (5) \\ &\leq \frac{1}{n}(1 + 0.01n) < 0.02. && (6)\end{aligned}$$

Transition (2) follows from applying Lemma A.4. Transition (3) follows from the fact that L_i is a function of $X_{[n]\setminus\{i\}}$ alone, so the information about the inputs $X_{[n]\setminus\{i\}}$ bounds the information about L_i. Transition (4) follows from the fact that the different inputs $\{X_j\}_{j\in[n]}$ are independent conditioned on π and $\mathsf{path}(k)$, as implied by Corollary B.8. Transition (5) follows from the fact that the inputs are independent, together with the fact that the superadditivity of information satisfies an equality in this case (Lemma A.2). Finally, Transition (6) follows from Definition 4.1 since k is the cutoff given $\pi, \mathsf{path}(k)$.

Then, the probability that the protocol is correct is at most the probability that the i-th party outputs the correct L_i. Using the above derivation we get,

$$\Pr[\text{correct output} \mid \Pi = \pi, \mathsf{PATH}(k) = \mathsf{path}(k)] \leq 2^{-H_\infty(L_i \mid \Pi=\pi, \mathsf{PATH}(k)=\mathsf{path}(k), X_i)} < 0.02.$$

□

5 Communication Lower Bound of Coding for the Cycle Task

In this section we prove our main theorem and show a lower bound of $\Omega(\log n)$ on the blowup of the communication for any protocol that solves the cycle task over noisy channels and succeeds with high probability. Formally, our main theorem is the following.

Theorem 5.1. *For any $\varepsilon \in (0,1)$ there exists a constant $c = c(\varepsilon)$ such that for large enough n, any protocol that solves the cycle task of depth d over a network with n parties communicating less than $cd \cdot n \log n$ symbols assuming each communication channel is an EC_ε, has a success probability at most $1/5$.*

Let χ' be a resilient protocol that solves the cycle task assuming noisy channels. The main idea is to show that $O(n \log n)$ symbols sent by the protocol χ' can increase the cutoff by at most $O(1)$, in expectation. That is, $O(\log n)$ cycles of the resilient protocol are required in order to advance $O(1)$ cycles in the cycle task, yielding a rate of $O(1/\log n)$.

Assume that given the (partial) observed transcript π and some path $\mathsf{path}(\ell)$, the cutoff of χ' is ℓ, that is, $\mathsf{cutoff}(\pi, \mathsf{path}(\ell)) = \ell$. Then, assume we let χ' communicate another $\delta \cdot n \log n$ symbols for some parameter $\delta = \delta(\varepsilon)$ we set later. We denote these new observed (potentially erased) symbols by Π^{new}; This is a random variable that depends on the noise and the randomness of the protocol. We claim that the new cutoff (i.e., with respect to $\pi \circ \Pi^{new}$), is bounded by $\ell + O(1)$ in expectation.

Proposition 5.2. *For any $\ell \leq d$, any $\mathsf{path}(\ell)$ and any transcript π,*

$$\mathbb{E}[\mathsf{cutoff}(\pi \circ \Pi^{new}, X) \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell), \mathsf{cutoff}(\pi, \mathsf{path}(\ell)) = \ell] \leq \ell + 500.$$

The proof of Proposition 5.2 spans the next several sections and concludes in Section 8. With the above proposition, the proof of the main theorem is immediate.

Proof. (**Theorem 5.1**) Assume χ' is a resilient protocol for the cycle task that succeeds with probability at least $1/5$. Proposition 4.2 claims that the expected cutoff at the end of the protocol χ' is at least $d/10$.

On the other hand, assume toward contradiction that χ' communicates less than $c \cdot d \cdot n \log n$ symbols. Split the transcript of χ' transcript into segments of $\delta \cdot n \log n$ transmissions each. Using Proposition 5.2, the cutoff at the end of χ' is bounded in expectation by

$$cd \cdot n \log n \cdot \frac{1}{\delta n \log n} \cdot 500 \leq \frac{500c}{\delta} d$$

By choosing, say, $c < \delta/5000$, we get that the expected cutoff at the end of χ' is strictly less than $d/10$, contradicting Proposition 4.2. □

5.1 Critical parties, and the event $\mathcal{E}_s$

We prove Proposition 5.2 in two steps. Most of the times, the noise in Π^{new} is large enough to show that many parties were completely erased. In this case, one can bound the expected increase in the cutoff by a constant (Section 6). However, it may happen that in a given segment of $\delta n \log n$ symbols, there was very little noise, so the resilient protocol χ' could practically advance without any restrictions. In this case, we show that the increase in the cutoff is at most $O(n \log n)$ in expectation (Section 7). However, this happens with very small probability of $2^{-\sqrt{n}}$, and has essentially no effect on the expected increase in the total cutoff (Section 8).

We now formally define the event $\mathcal{E}_s$ that indicates that the noise, in a given segment of $\delta n \log n$ symbols, is sufficient for our needs of bounding the cutoff. We begin by defining *critical parties*. These are parties that we know very small amount of information about their inputs below the cutoff level, and on top of that, all their communication (in the new segment) was erased.

Definition 5.3 (Critical party)**.** *Assume a (partial) transcript π and a path $\mathsf{path}(k)$ for which $\mathsf{cutoff}(\pi, \mathsf{path}(k)) = k$, and let Π^{new} be the observed transcript of the next $\delta n \log n$ symbols. The i-th party is called* critical *at the cutoff, if*

(i) $I(X_i[\mathsf{path}_i(k)] \mid \Pi = \pi, \mathsf{PATH}(k) = \mathsf{path}(k)) < 0.03$, *and*

(ii) all its outgoing communication was completely erased in Π^{new} (or alternatively, this party didn't speak at all).

Denote the set of all the critical parties as $\mathcal{C}$ and the non-critical as $\overline{\mathcal{C}}$.

The next claim proves that with high probability, at any segment of $\delta n \log n$ transmissions, there exists a large set of critical parties.

Claim 5.4. *Assume a (partial) transcript π and a path $\mathsf{path}(k)$ for which $\mathsf{cutoff}(\pi, \mathsf{path}(k)) = k$. For any $\varepsilon \in (0,1)$, $\delta < \min\{0.1, 1/\log(\varepsilon^{-10})\}$ let Π^{new} be the observed transcript of the next $\delta n \log n$ symbols. Denote by $\mathcal{E}_s$ the event that $|\mathcal{C}| > \sqrt{n}$, then*

$$\Pr[\mathcal{E}_s] \geq 1 - 2^{-\sqrt{n}}.$$

Proof. Note that there are at least $2n/3$ parties for which

$$I(X_i[\mathsf{path}_i(k)] \mid \Pi = \pi, \mathsf{PATH}(k) = \mathsf{path}(k)) < 0.03.$$

Indeed, if there are more than $n/3$ parties whose information is above 0.03, then

$$\sum_{i=1}^{n} I(X_i[\mathsf{path}_i(k)] \mid \Pi = \pi, \mathsf{PATH}(k) = \mathsf{path}(k)) \geq n/3 \times 0.03 \geq 0.01n$$

in contradiction to the cutoff definition. Denote the above set of parties by Q_1. At the same time, we consider only parties that speak at most $3\delta \log n$ symbols in the given continuation. There are at least $2n/3$ such parties, or otherwise, the total communication exceeds $n/3 \times 3\delta \log n = \delta n \log n$. Denote these parties by Q_2.

Next we focus only on parties in $Q_1 \cap Q_2$ (there are at least $n/3$ such parties), and show that many of them are fully erased in the given continuation segment. Each such party is completely erased with probability at least $\varepsilon^{3\delta \log n}$. Since we assume $\delta < \min\{0.1, 1/\log(\varepsilon^{-10})\}$ we get that $\varepsilon^{3\delta \log n} \geq 2^{3\delta \log \varepsilon \log n} \geq n^{-0.4}$. Since there are at least $n/3$ parties in $Q_1 \cap Q_2$, in expectation we will have $n^{0.6}/3$ parties completely silenced. Using Chernoff's inequality (Lemma B.6) it is easy to show that, except with probability at most $2^{-\sqrt{n}}$, at least $\sqrt{n}$ parties from $Q_1 \cap Q_2$ are completely erased. Any such party is critical by definition, which completes the proof. □

6 Bounding the cutoff when $\mathcal{E}_s$ occurs

In this section we bound the progress of the cutoff, assuming $\mathcal{E}_s$ occurs. That is, we prove the following proposition.

Proposition 6.1. *For any $\ell \leq T$, any $\mathsf{path}(\ell)$ and any transcript π*

$$\mathbb{E}[\mathsf{cutoff}(\pi \circ \Pi^{new}, X) \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell), \mathsf{cutoff}(\pi, \mathsf{path}(\ell)) = \ell, \mathcal{E}_s] \leq \ell + 100.$$

Generally speaking, the proof follows the bounding technique of [BEGH16]. We begin by setting some notations used throughout this section and proving some auxiliary properties (Section 6.1). We then bound the expected amount of information the new segment leaks about the subtrees of the inputs rooted on the correct path below the current cutoff (Section 6.2). Finally, we use the fact that the information on such subtrees is small to bound the increase in the new cutoff (Section 6.3).

6.1 Preliminaries

Let $\mathcal{E}_s$ be the event that $|\mathcal{C}| \geq \sqrt{n}$ as defined in Claim 5.4. Furthermore, we define the following shorthands,

$$\begin{aligned}
\mathcal{E} &\stackrel{\mathsf{def}}{=} (\Pi = \pi, \Pi^{new} = \pi^{new}, \mathsf{PATH}(\ell) = \mathsf{path}(\ell)), \\
\mathcal{E}^{+i} &\stackrel{\mathsf{def}}{=} (\mathcal{E}, X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}), \\
\mathcal{E}^{+} &\stackrel{\mathsf{def}}{=} \bigcup_{i \in [n]} \mathcal{E}^{+i}, \\
Z_i(k) &\stackrel{\mathsf{def}}{=} \mathsf{PATH}_i(k + \ell).
\end{aligned}$$

Recall that whether the cutoff is ℓ depends only on π and the first ℓ levels of the correct path, therefore, the event $(\Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell), \mathsf{cutoff}(\pi, \mathsf{path}(\ell)) = \ell)$ is either empty or equal to $(\Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell))$. In the following we will implicitly assume that the event is not empty and explicitly condition only on $(\Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell))$.

Given any $\pi, \mathsf{path}(\ell)$ we define the functions

$$C_i^*(k \mid \pi^{new}, \mathsf{path}(k + \ell)) \stackrel{\mathsf{def}}{=} I\left(X_i[\mathsf{path}_i(k + \ell)] \mid \Pi = \pi, \Pi^{new} = \pi^{new}, \mathsf{PATH}(k + \ell) = \mathsf{path}(k + \ell)\right)$$

and

$$\begin{aligned}
C_i(k \mid \pi^{new}, x_i[\mathsf{path}_i(\ell)]^{\leq k}) \stackrel{\mathsf{def}}{=} \mathbb{E}_{\rho \sim \mathsf{PATH}(k+\ell) \mid x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathcal{E}} \\
I(X_i[\rho_i] \mid X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathsf{PATH}(k + \ell) = \rho, \mathcal{E}).
\end{aligned}$$

In words, $C_i^*()$ measures the information on the inputs when starting k levels below $\mathsf{path}(\ell)$, assuming we are given the path continuation from level ℓ to level k, that is, we know $\mathsf{path}(\ell + k)$. In the measure $C_i()$ we do not know how to continue $\mathsf{path}(\ell)$ from level ℓ to k, so we take the expectation (of the information below level $k + \ell$) on all possible continuations $\mathsf{path}(\ell + k)$. Moreover, in $C_i()$ we condition on additional information, namely, the labels in the first k levels of the subtree of x_i rooted at the end of $\mathsf{path}_i(\ell)$.

The quantity $\sum_i C_i^*()$ is exactly the measure of information we wish to bound in order to bound the new cutoff (towards satisfying Eq. (1) given the new segment of communication). However, due to technical reasons, namely, in order to obtain independence (see Claim 6.3 below), we will actually bound $\sum_i C_i()$, which in turn bounds $\sum_i C_i^*()$ via the following claim.

Claim 6.2 ([BEGH16])**.** *Given any $\pi, \pi^{new}, \mathsf{path}(\ell)$, and for any k, and any $i \in [n]$,*

$$\mathbb{E}_{\mathsf{path}(k+\ell) \mid \mathcal{E}, \mathcal{E}_s} C_i^*(k \mid \pi^{new}, \mathsf{path}(k + \ell)) \leq \mathbb{E}_{x_i[\mathsf{path}_i(\ell)]^{\leq k} \mid \mathcal{E}, \mathcal{E}_s} C_i(k \mid \pi^{new}, x_i[\mathsf{path}_i(\ell)]^{\leq k}).$$

Proof. Note that $\mathcal{E}$ fully determines whether $\mathcal{E}_s$ occurs or not. Indeed: π^{new} determines which bits are erased, and $\pi, \mathsf{path}(\ell)$ determine the set of critical parties. Therefore, conditioning on $\mathcal{E}$ for some fixed $(\pi, \mathsf{path}(\ell), \pi^{new}) \in \mathcal{E}_s$ is equivalent to conditioning on both $\mathcal{E}$ and $\mathcal{E}_s$ (or otherwise, the claim vacuously holds).

$$\mathbb{E}_{x_i[\mathsf{path}(\ell)]^{\leq k}|\mathcal{E}} C_i(k \mid \pi^{new}, x_i[\mathsf{path}_i(\ell)]^{\leq k})$$
$$= \mathbb{E}_{x_i[\mathsf{path}(\ell)]^{\leq k}|\mathcal{E}} \mathbb{E}_{\rho\sim\mathsf{PATH}(k+\ell)|x_i[\mathsf{path}_i(\ell)]^{\leq k},\mathcal{E}} I(X_i[\rho_i] \mid X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathsf{PATH}(k+\ell) = \rho, \mathcal{E})$$

exchanging the order of expectations

$$= \mathbb{E}_{\rho\sim\mathsf{PATH}(k+\ell)|\mathcal{E}} \mathbb{E}_{x_i[\mathsf{path}(\ell)]^{\leq k}|\rho,\mathcal{E}} I(X_i[\rho_i] \mid X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathsf{PATH}(k+\ell) = \rho, \mathcal{E})$$

by the definition of conditional information (Definition A.1),

$$= \mathbb{E}_{\rho\sim\mathsf{PATH}(k+\ell)|\mathcal{E}} I(X_i[\rho_i] \mid X_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathsf{PATH}(k+\ell) = \rho, \mathcal{E})$$

using Lemma B.1(1) we get

$$\geq \mathbb{E}_{\rho\sim\mathsf{PATH}(k+\ell)|\mathcal{E}} I(X_i[\rho_i] \mid \mathsf{PATH}(k+\ell) = \rho, \mathcal{E})$$
$$= \mathbb{E}_{\rho\sim\mathsf{PATH}(k+\ell)|\mathcal{E}} C_i^*(k \mid \pi^{new}, \rho).$$

□

Claim 6.3. *Conditioned on $\mathcal{E}^{+i}$, the path $Z_i(k)$ is independent of the labels in the subtrees of X_i rooted at $\mathsf{path}_i(\ell)$.*

Proof. Once we condition on $\mathsf{path}_i(\ell)$ and on $X_i[\mathsf{path}_i(\ell)]^{\leq k}$, then the continuation $Z_i(k)$ depends only on $X_{\neq i}$ between layers ℓ and $k+\ell$, and these are independent of X_i below ℓ, even conditioned on the transcript, etc. (Corollary B.8). □

6.2 Bounding the information in subtrees below the cutoff

In the rest of this subsection we prove that the expected information on subtrees of the input starting 30 steps below the cutoff level is sub-exponentially small.

Lemma 6.4. *Given any $(\pi, \mathsf{path}(\ell))$ for which the cutoff is ℓ, it holds that*

$$\sum_{k=30}^{d-\ell} \mathbb{E}_{\pi^{new}, x[\mathsf{path}(\ell)]^{\leq k}|\Pi=\pi, \mathsf{PATH}(\ell)=\mathsf{path}(\ell), \mathcal{E}_s} \left[\sum_{i=1}^{n} C_i(k \mid \pi^{new}, x_i[\mathsf{path}_i(\ell)]^{\leq k}) \right] < 2^{-0.1\sqrt{n}} \cdot n^2 \log n.$$

Proof. Lemma B.2 implies that, for any k, we can bound $C_i(k)$ as the product of the probability to guess a path Z_i of length k, times the information the i-th party communicated about its x_i below level $k+\ell$. Formally,

$$C_i(k \mid \pi^{new}, x_i[\mathsf{path}_i(\ell)]^{\leq k}) \leq p_{\max}\left(Z_i(k) \;\middle|\; X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathcal{E}\right) \times I\left(X_i[\mathsf{path}_i(\ell)]^{>k} \;\middle|\; X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathcal{E}\right). \quad (7)$$

Note that in order to use Lemma B.2 we need to assert that $Z_i(k)$ is independent of X_i below level $\ell + k$ (conditioned on $\mathcal{E}^{+i}$), which is provided by Lemma 6.3.

The lemma then follows from the following claims, which bound the two multiplicands separately.

Claim 6.5. *For any $i \in [n]$, let $a_i(k)$ be such that $\Pr[Z_i(k) = \vec{a}_i(k) \mid \mathcal{E}^{+i}] = p_{\max}\big(Z_i(k) \mid \mathcal{E}^{+i}\big)$. $\vec{a}_i(k)$ can be seen as the path of length k taken by p_i, which corresponds to the labels of the $n-1$ paths traversed by the other $n-1$ parties; we let $(\vec{a}_i(k))_j$ be the part corresponding to the j-th party. Denote by $P_j(k)$ a possible path of length k in x_j starting from the ℓ-th level. Then,*

$$p_{\max}\big(Z_i(k) \mid \mathcal{E}^{+i}\big) \leq \prod_{j \in \mathcal{C}} \max_{P_j(k)} \Pr[label(P_j(k)) = (\vec{a}_i(k))_j \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell)].$$

In particular, the right-hand side is not conditioned on π^{new} nor on $x_i[\mathsf{path}_i(\ell)]^{\leq k}$.

Claim 6.6. *For any $i \in [n]$,*

$$\begin{aligned}\sum_{k=30}^{d-\ell} p_{\max}\big(Z_i(k) \mid \mathcal{E}^{+i}\big) &\leq \sum_{k=30}^{d-\ell} \prod_{j \in \mathcal{C}} \max_{P_j(k)} \Pr[label(P_j(k)) = (\vec{a}_i(k))_j \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell)] \\ &\leq 2^{-0.1\sqrt{n}}\end{aligned}$$

Claim 6.7.

$$\sum_{i=1}^{n} \mathbb{E}_{\pi^{new} \mid \Pi=\pi, \mathsf{PATH}(\ell)=\mathsf{path}(\ell), \mathcal{E}_s} \left[I(X_i[\mathsf{path}_i(\ell)] \mid \mathcal{E})\right] \leq n^2 \log n.$$

Armed with the above claims, we can prove the lemma.

$$\begin{aligned}&\sum_{k=30}^{d-\ell} \mathbb{E}_{\pi^{new}, x[\mathsf{path}(\ell)]^{\leq k} \mid \Pi=\pi, \mathsf{PATH}(\ell)=\mathsf{path}(\ell), \mathcal{E}_s} \left[\sum_{i=1}^{n} C_i(k \mid \pi^{new}, x_i[\mathsf{path}_i(\ell)]^{\leq k})\right] \\ &\leq \sum_{k=30}^{d-\ell} \mathbb{E}_{\pi^{new}, x[\mathsf{path}(\ell)]^{\leq k} \mid \Pi=\pi, \mathsf{PATH}(\ell)=\mathsf{path}(\ell), \mathcal{E}_s} \Bigg[\sum_{i=1}^{n} p_{\max}\left(Z_i(k) \;\middle|\; X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathcal{E}\right) \\ &\qquad \times I\left(X_i[\mathsf{path}_i(\ell)]^{>k} \;\middle|\; X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathcal{E}\right)\Bigg]\end{aligned}$$

Using Claim 6.5, and noting that the first multiplicand is constant with respect to the expectation

$$\begin{aligned}&\leq \sum_{i=1}^{n} \sum_{k=30}^{d-\ell} \prod_{j \in \mathcal{C}} \max_{P_j(k)} \Pr[label(P_j(k)) = (\vec{a}_i(k))_j \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell)] \\ &\qquad \times \mathbb{E}_{\pi^{new}, x[\mathsf{path}(\ell)]^{\leq k} \mid \Pi=\pi, \mathsf{PATH}(\ell)=\mathsf{path}(\ell), \mathcal{E}_s} I\left(X_i[\mathsf{path}_i(\ell)]^{>k} \;\middle|\; X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathcal{E}\right)\end{aligned}$$

Performing the expectation over $x_i[\mathsf{path}_i(\ell)]^{\leq k}$ (note that given $\Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell)$, the variables $x_i[\mathsf{path}_i(\ell)]^{\leq k}$ are independent of the event $\mathcal{E}_s$), and then using Lemma B.1(3),

$$\begin{aligned}&\leq \sum_{i=1}^{n} \left(\sum_{k=30}^{d-\ell} \prod_{j \in \mathcal{C}} \max_{P_j(k)} \Pr[label(P_j(k)) = (\vec{a}_i(k))_j \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell)]\right) \\ &\qquad \times \mathbb{E}_{\pi^{new} \mid \Pi=\pi, \mathsf{PATH}(\ell)=\mathsf{path}(\ell), \mathcal{E}_s} I\left(X_i[\mathsf{path}_i(\ell)] \mid \mathcal{E}\right)\end{aligned}$$

Now that the second term does not depend on k, we use Claim 6.6 on the first term (note that Claim 6.6 applies to any $i \in [n]$)

$$\leq 2^{-0.1\sqrt{n}} \times \sum_{i=1}^{n} \mathbb{E}_{\pi^{new} \mid \Pi=\pi, \mathsf{PATH}(\ell)=\mathsf{path}(\ell), \mathcal{E}_s} I\left(X_i[\mathsf{path}_i(\ell)] \mid \mathcal{E}\right)$$

We conclude by employing Claim 6.7,

$$\leq 2^{-0.1\sqrt{n}} \times n^2 \log n.$$

□

We are left to prove the above three claims.

The following derivation proves both Claim 6.5 and Claim 6.6. The claim shows that since the path Z_i is affected by the labels of all other $n-1$ parties, guessing that path is at least as difficult as guessing the labels of all the critical parties (that were erased so we have no information about their labels). Since there are $\sqrt{n}$ such critical parties, we can guess their labels with only sub-exponentially small probability.

Claim 6.8. *For any $i \in [n]$,*

$$\sum_{k=30}^{d-\ell} p_{\max}\big(Z_i(k) \mid \mathcal{E}^{+i}\big) \leq 2^{-0.1\sqrt{n}}$$

Proof. For any specific k, assume a path $\vec{a}_i(k)$ of length k that maximizes this probability,

$$\Pr[Z_i(k) = \vec{a}_i(k) \mid X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathcal{E}].$$

Once fixing $\vec{a}_i(k)$, it is implied that there exist $n-1$ paths $P_{i-1}(k), \ldots, P_{i-(n-1)}(k)$ of length k in $X_{i-1}, \ldots, X_{i-(n-1)}$ respectively (modulus the number of parties, and adjusting the level in the input tree if we "loop around"), where each path starts from level ℓ, as a continuation of $\mathsf{path}(\ell)$. **The labels along these paths are exactly** $\vec{a}_i(k)$ (up to trivial reordering). The probability that Z_i goes through a path $a_i(k)$ is bounded by the probability to see the corresponding labels somewhere in the respective trees,

$$\begin{aligned}
&\Pr[Z_i(k) = \vec{a}_i(k) \mid X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathcal{E}] \\
&\leq \max_{P(k)} \Pr[\{label(P_{i-1}(k)), \ldots, label(P_{i-(n-1)}(k))\} = \vec{a}_i(k) \mid X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathcal{E}] \\
&= \max_{P(k)} \Pr[\{label(P_{i-1}(k)), \ldots, label(P_{i-(n-1)}(k))\} = \vec{a}_i(k) \mid \mathcal{E}],
\end{aligned}$$

where the last step follows from Corollary B.8 that guarantees us the independence of the labels of $X_j{}^{>\ell}$ from those of $X_i{}^{>\ell}$ for any $j \neq i$, even when conditioning on the transcript so far π, and on $\mathcal{E}$.

We can then bound the sum of the guessing probability of the path $Z_i(k)$, for all large enough k's:

$$\begin{aligned}
&\sum_{k=30}^{d-\ell} p_{\max}(Z_i(k) \mid X_i[\mathsf{path}_i(\ell)]^{\leq k} = x_i[\mathsf{path}_i(\ell)]^{\leq k}, \mathcal{E}) \\
&\leq \sum_{k=30}^{d-\ell} \max_{P(k)} \Pr[\{label(P_{i-1}(k)), \ldots, label(P_{i-n}(k))\} = \vec{a}_i(k) \mid \mathcal{E}]
\end{aligned}$$

Using the independence of labels for different parties, Corollary B.8 we have

$$\leq \sum_{k=30}^{d-\ell} \prod_{j \in [n] \setminus \{i\}} \max_{P_j(k)} \Pr[label(P_j(k)) = (\vec{a}_i(k))_j \mid \mathcal{E}]$$

In the above, $(\vec{a}_i(k))_j$ denotes the parts of $\vec{a}_i(k)$ that correspond to the j-th party. We can consider only the critical parties (assume we can fully guess the others)[5]. Recall that each critical party is completely erased in π^{new}; therefore (along with Corollary B.8) conditioning on π^{new} does not affect the probability to see a specific labeling on a given path of x_j

$$\leq \sum_{k=30}^{d-\ell} \prod_{j\in\mathcal{C}} \max_{P_j(k)} \Pr[label(P_j(k)) = (\vec{a}_i(k))_j \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell)]$$
$$\leq \prod_{j\in\mathcal{C}} \sum_{k=30}^{d-\ell} \max_{P_j(k)} \Pr[label(P_j(k)) = (\vec{a}_i(k))_j \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell)]$$

Using Lemma B.4

$$\leq \prod_{j\in\mathcal{C}} \left(2I_j + 4\sqrt{I_j} + 20\cdot 2^{-30/4}\right)$$
$$\leq \prod_{j\in\mathcal{C}} 0.9 \leq 2^{-0.1\sqrt{n}}.$$

where the penultimate transition is via Lemma B.4 by letting T of the lemma be all the labels of $X_j[\mathsf{path}_j(\ell)]^{>\ell}$, and setting $I_j = I(X_j[\mathsf{path}_j(\ell)]^{>\ell} \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell))$. If j is critical (given the "old" transcript, $\Pi = \pi$), then $I_j < 0.03$ by definition, and $2I_j + 4\sqrt{I_j} + 20\cdot 2^{-30/4} < 0.9$. Recalling that $|\mathcal{C}| > \sqrt{n}$ completes the proof. □

We now continue to proving Claim 6.7. Let us recall the claim:

Claim 6.7.

$$\sum_{i=1}^{n} \mathbb{E}_{\pi^{new}|\Pi=\pi,\mathsf{PATH}(\ell)=\mathsf{path}(\ell),\mathcal{E}_s} [I(X_i[\mathsf{path}_i(\ell)] \mid \mathcal{E})] \leq n^2 \log n.$$

Proof. Let $\mathcal{E}^- \stackrel{\mathsf{def}}{=} (\Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell))$. Writing $\mathcal{E}$ explicitly in the claim's statement, we have

$$\sum_{i=1}^{n} \mathbb{E}_{\pi^{new}|\mathcal{E}^-,\mathcal{E}_s} I(X_i[\mathsf{path}_i(\ell)] \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell), \Pi^{new} = \pi^{new}).$$

by linearity of expectation and the superadditivity of information (Lemma A.2),

$$\leq \mathbb{E}_{\pi^{new}|\mathcal{E}^-,\mathcal{E}_s} I(X[\mathsf{path}(\ell)] \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell), \Pi^{new} = \pi^{new})$$
$$= I(X[\mathsf{path}(\ell)] \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell), \widetilde{\Pi}^{new})$$

where $\widetilde{\Pi}^{new}$ is distributed according to Π^{new} conditioned on $\mathcal{E}^-, \mathcal{E}_s$. Recall that $\widetilde{\Pi}^{new}$ contains up to $\delta n \log n$ symbols, each of size at most n bits (where some symbols may be corrupted). Using Lemma B.1(2),

$$\leq I(X[\mathsf{path}(\ell)] \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell)) + \delta \cdot n^2 \log n.$$

Now note that, conditioned on $(\Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell))$, the variables $X_1, \ldots, X_n$ are mutually independent by Corollary B.8, thus the superadditivity (Lemma A.2) in this case satisfies an equality,

$$= \delta n^2 \log n + \sum_{i=1}^{n} I(X_i[\mathsf{path}_i(\ell)] \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell))$$

[5]We neglect the fact that party i shouldn't be included in the product, in case $i \in \mathcal{C}$.

finally, since ℓ is the cutoff given the transcript π, and recalling that $\delta < 0.1$ we get

$$\leq \delta n^2 \log n + 0.01n$$
$$\leq n^2 \log n.$$ □

6.3 Bounding the increase in the cutoff

We now show that, given that the old cutoff was ℓ and that $\mathcal{E}_s$ occurred in the new $\delta n \log n$ transmissions, then the expected new cutoff is at most $\ell + O(1)$.

Proof. (**Proposition 6.1**) Given $(\pi, \mathsf{path}(\ell))$ for which the cutoff is ℓ, consider the following series of non-negative random variables

$$\left\{ \tilde{C}(k) \stackrel{\mathsf{def}}{=} \mathbb{E}_{\pi^{new}, \mathsf{path}(k+30+\ell) | \pi, \mathsf{path}(\ell), \mathcal{E}_s} \Big[\sum_{i=1}^{n} C_i^*(k+30 \mid \pi^{new}, \mathsf{path}(k+30+\ell)) \Big] \right\}_{k \geq 0}$$

Lemma 6.4 and Claim 6.2 certify that $\sum_k \tilde{C}(k) \leq 2^{-0.1\sqrt{n}} \cdot n^2 \log n$. Therefore from Lemma B.3 it follows that the expectation of the minimal k^* for which $\sum_{i=1}^{n} C_i^*(k^*+30 \mid \pi^{new}, \mathsf{path}(k+30+\ell)) < 0.01n$ is bounded, for large enough n, by

$$\mathbb{E}[k^*] \leq 1 + \frac{2^{-0.1\sqrt{n}} \cdot n^2 \log n}{0.01n} \leq 2.$$

This implies that the cutoff has advanced by at most $30 + k^* \leq 32$ in expectation, since the new cutoff is exactly the point where $\sum_{i=1}^{n} C_i^*(k^*+30 \mid \pi^{new}, \mathsf{path}(k+30+\ell))$ goes below the threshold $0.01n$. (Note: we give the first 30 rounds "for free" since our bound in Lemma 6.4 applies only for levels below depth $\ell + 30$.) □

7 Bounding the cutoff when $\mathcal{E}_s$ does not occur

The following proposition bounds the progress of the cutoff in the rare case where $\mathcal{E}_s$ doesn't occur.

Proposition 7.1. *For any $\ell \leq d$, any $\mathsf{path}(\ell)$ and any transcript π*

$$\mathbb{E}[\mathsf{cutoff}(\pi \circ \Pi^{new}, X) \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell), \mathsf{cutoff}(\pi, \mathsf{path}(\ell)) = \ell, \overline{\mathcal{E}_s}] \leq \ell + 500n \log n.$$

Proof. We show that each single noiseless transmission can be simulated, by $k \cdot \delta n \log n$ transmissions for which $\mathcal{E}_s$ occurs, in expectation, where k is a small constant. We know (Proposition 6.1) that a segment of $\delta n \log n$ transmissions in which $\mathcal{E}_s$ occurs can increase the cutoff by at most $O(1)$ in expectation. Furthermore, k only depends on the random noise pattern and is independent of the increase in the cutoff. Therefore, a segment of $\delta n \log n$ noiseless transmissions (i.e., a segment in which $\mathcal{E}_s$ doesn't occur) increases the cutoff by at most $(\delta n \log n) \cdot k \cdot O(1) = O(n \log n)$ levels in expectation.

Let $c \geq 1$ be some constant. We can assume that all the noisy transmissions belong to the same channel as the (noiseless) transmission we wish to simulate[6]. The probability that c segments of

[6]This assumption, in fact, causes $\mathcal{E}_s$ to occur with probability 1, but we will nevertheless bound the event probability by $\Pr[\mathcal{E}_s] \geq 1 - 2^{-\sqrt{n}}$ via Claim 5.4.

$\delta n \log n$ noisy transmissions each, in each of which $\mathcal{E}_s$ occurs, are not enough to simulate one single noiseless transmission is at most

$$\begin{aligned}\Pr[k \geq c] &\leq \Pr\left[\begin{matrix}\text{all the transmissions in the first } c \\ \text{segments are erased}\end{matrix} \;\middle|\; \mathcal{E}_s \text{ occurs in all } c \text{ segments}\right] \\ &< \frac{\varepsilon^{c\delta n \log n}}{\left(1-2^{-\sqrt{n}}\right)^c} \\ &= \left(\frac{2^{\delta \log \varepsilon \cdot n \log n}}{1-2^{-\sqrt{n}}}\right)^c\end{aligned}$$

The base of the above exponent goes to zero with $n \to \infty$, thus for large enough n,

$$\begin{aligned}\mathbb{E}[k] &= \sum_{c=0}^{\infty} \Pr[k \geq c] \\ &\leq 5\end{aligned}$$

It follows that a segment of $\delta n \log n$ transmissions in which $\mathcal{E}_s$ did not occur, can increase the cutoff by at most $500\delta n \log n$ levels in expectation. □

8 Completing the proof of Proposition 5.2

Given Proposition 6.1 and Proposition 7.1 we are able to complete the proof of Proposition 5.2.

Proof. For large enough n,

$$\begin{aligned}&\mathbb{E}[\mathsf{cutoff}(\pi \circ \Pi^{new}, X) \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell), \mathsf{cutoff}(\pi, \mathsf{path}(\ell)) = \ell] \\ &= \mathbb{E}[\mathsf{cutoff}(\pi \circ \Pi^{new}, X) \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell), \mathsf{cutoff}(\pi, \mathsf{path}(\ell)) = \ell, \mathcal{E}_s] \times \Pr[\mathcal{E}_s] \\ &\quad + \mathbb{E}[\mathsf{cutoff}(\pi \circ \Pi^{new}, X) \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell), \mathsf{cutoff}(\pi, \mathsf{path}(\ell)) = \ell, \overline{\mathcal{E}_s}] \times \Pr[\overline{\mathcal{E}_s}] \\ &\leq (\ell + 100) \times (1 - 2^{-\sqrt{n}}) + (\ell + 500n \log n) 2^{-\sqrt{n}} \\ &\leq \ell + 500.\end{aligned}$$

□

9 On the Rate vs. Channel's Alphabet

In this section we discuss the effect of the channel's alphabet size on the obtainable rate. We can consider four independent settings: binary/large alphabet at the original (noiseless) scheme vs. binary/large alphabet at the coding scheme. For any $n \in \mathbb{N}$ and for $orig, code \in \{b, l\}$ let $c_{orig,code}(n)$ be the infimum over all possible n-party functions f of the maximal rate obtainable when the original protocol χ for f is binary ($orig = b$) or with a large alphabet ($orig = l$) and the coding schemes χ' for f is binary or with a large alphabet ($code = b$ or $code = l$, respectively),

$$c_{orig,code}(n) = \inf_f \frac{\min_\chi \mathsf{CC}(\chi)}{\min_{\chi'} \mathsf{CC}(\chi')}.$$

The capacity of each setting—the maximal achievable rate in each setting—is defined to be the limit inferior of the above quantities when n tends to infinity,

$$c_{orig,code} = \liminf_{n \to \infty} c_{orig,code}(n).$$

We now explore relations between the four capacities. See Table 1 for a summary of the relations between the capacities of the different settings.

Noiseless Scheme χ \ Coding Scheme χ'	binary alphabet	large alphabet
binary alphabet	c_{bb}	$c_{bl} \geq \frac{c_{bb}}{\log \lvert\Sigma\rvert}$
large alphabet	$\Omega(c_{ll}) \leq c_{lb}$ $c_{bb} \leq c_{lb}$	$c_{ll} \geq c_{bl}$ $c_{ll} \geq \frac{c_{lb}}{\log \lvert\Sigma\rvert}$

Table 1: The relations between maximal rates of coding schemes with {binary, large}-alphabet, given the noiseless protocol uses {binary, large}-alphabet.

Any binary coding can be simulated by a large-alphabet coding by incurring a blowup of $\log |\Sigma|$, thus trivial relations are $c_{bl} \geq c_{bb}/\log |\Sigma|$ and $c_{ll} \geq c_{lb}/\log |\Sigma|$.

When the original protocol uses large alphabet, a large-alphabet coding can be reduced to a binary one by translating each symbol to a sequence of bits encoded with a standard error-correction code (so that the probability for the entire sequence to be decoded incorrectly is below ε; this can be done with a constant overhead). Thus $\Omega(c_{ll}) \leq c_{lb}$.

To see that $c_{lb} \geq c_{bb}$, note that we can convert the original large-alphabet protocol (that determines c_{lb}) into a binary one with the same communication complexity; this converted protocol may not be the hardest one for coding with a binary simulation, thus the rate we can achieve when coding it may be larger than the rate for the "worst" binary protocol, which determines c_{bb}. A similar reasoning yields $c_{ll} \geq c_{bl}$.

The above relations still allow c_{bb} to be either larger or smaller than c_{ll}, and their specific relation (as well as their feasibility with respect to a given underlying topology) remains an interesting open question.

References

[AGS16] S. Agrawal, R. Gelles, and A. Sahai. Adaptive protocols for interactive communication. *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 595–599, 2016.

[ABE+16] N. Alon, M. Braverman, K. Efremenko, R. Gelles, and B. Haeupler. Reliable communication over highly connected noisy networks. *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, pp. 165–173, 2016.

[BEGH16] M. Braverman, K. Efremenko, R. Gelles, and B. Haeupler. Constant-rate coding for multiparty interactive communication is impossible. *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2016, pp. 999–1010, 2016.

[BGMO16] M. Braverman, R. Gelles, J. Mao, and R. Ostrovsky. Coding for interactive communication correcting insertions and deletions. *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, *Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 55, pp. 61:1–61:14, 2016.

[FGOS15] M. Franklin, R. Gelles, R. Ostrovsky, and L. J. Schulman. Optimal coding for streaming authentication and interactive communication. *Information Theory, IEEE Transactions on*, 61(1):133–145, 2015.

[Gal88] R. Gallager. Finding parity in a simple broadcast network. *Information Theory, IEEE Transactions on*, 34(2):176–180, 1988.

[Gel15] R. Gelles. Coding for interactive communication: A survey, 2015.

[GMS14] R. Gelles, A. Moitra, and A. Sahai. Efficient coding for interactive communication. *Information Theory, IEEE Transactions on*, 60(3):1899–1913, 2014.

[GHS14] M. Ghaffari, B. Haeupler, and M. Sudan. Optimal error rates for interactive coding I: Adaptivity and other settings. *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pp. 794–803, 2014.

[GKS08] N. Goyal, G. Kindler, and M. Saks. Lower bounds for the noisy broadcast problem. *SIAM Journal on Computing*, 37(6):1806–1841, 2008.

[Hae14] B. Haeupler. Interactive Channel Capacity Revisited. *Proceedings of the IEEE Symposium on Foundations of Computer Science*, FOCS '14, pp. 226–235, 2014.

[HS16] W. M. Hoza and L. J. Schulman. The adversarial noise threshold for distributed protocols. *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 240–258, 2016.

[JKL15] A. Jain, Y. T. Kalai, and A. Lewko. Interactive coding for multiparty protocols. *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science, ITCS '15*, pp. 1–10, 2015.

[KR13] G. Kol and R. Raz. Interactive channel capacity. *STOC '13: Proceedings of the 45th annual ACM Symposium on theory of computing*, pp. 715–724, 2013.

[RS94] S. Rajagopalan and L. Schulman. A coding theorem for distributed computation. *STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pp. 790–799, 1994.

[Sch92] L. J. Schulman. Communication on noisy channels: a coding theorem for computation. *Foundations of Computer Science, Annual IEEE Symposium on*, pp. 724–733, 1992.

[Sch96] L. J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996.

Appendix

A Preliminaries: Information, entropy, and min-entropy

Throughout, we will use U_n to denote a random variable uniformly distributed over $\{0,1\}^n$.

Definition A.1 (information)**.** *Let X be a random variable over a finite discrete domain Ω. The* information *of X is given by*

$$I(X) \stackrel{\mathsf{def}}{=} \log|\Omega| - H(X),$$

where $H(X)$ is the Shannon entropy of X, $H(X) = \sum_{x\in\Omega} \Pr(X=x)\log(1/\Pr(X=x))$.

Given a random variable Y, the conditional information *of X given Y is*

$$\begin{aligned} I(X \mid Y) &\stackrel{\mathsf{def}}{=} \log|\Omega| - H(X \mid Y) \\ &= \mathbb{E}_y I(X \mid Y = y). \end{aligned}$$

Lemma A.2 (superadditivity of information)**.** *Let $X_1, \ldots, X_n$ be n random variables. Then,*

$$\sum_{i=1}^{n} I(X_i) \leq I(X_1, \ldots, X_n).$$

The equality is satisfied when $X_1, \ldots, X_n$ are mutually independent.

Proof. Using the subadditivity of the entropy function, we get

$$\sum_{i=1}^{n} I(X_i) = \sum_{i} \left(\log |\Omega_i| - H(X_i)\right) \leq \log \left(\prod_{i} |\Omega_i|\right) - H(X_1, \ldots, X_n) = I(X_1, \ldots, X_n).$$

□

Definition A.3 (min-entropy)**.** *Let X be a random variable over a discrete domain Ω. The min-entropy of X is given by*

$$H_\infty(X) = \log(1/p_{\max}(X)).$$

$p_{\max}(X)$ is the probability of the most probable value of X, i.e., $p_{\max}(X) \stackrel{\mathsf{def}}{=} \max_{x \in \Omega} \Pr(X = x)$. At times, $p_{\max}$ is called the guessing probability *of X.*

Information (i.e., entropy) can be related to guessing probability (i.e., min-entropy) via the next Lemma, which is a special case of Fano's inequality.

Lemma A.4 ([BEGH16])**.** *Let X be a random variable over a discrete finite domain Ω. It holds that*

$$I(X) \geq p_{\max}(X) \log(|\Omega|) - h(p_{\max}(X)),$$

where $h(x) = -x \log x - (1-x)\log(1-x)$ is the binary entropy. Note that $0 \leq h(x) \leq 1$, then an immediate corollary is

$$p_{\max}(X) \leq \frac{I(X) + 1}{\log |\Omega|}.$$

We note here that similar claims to the above lemmas hold when we additionally condition on some event $\mathcal{E}$, by applying the above lemmas on the random variable $(X \mid \mathcal{E})$.

B Technical Lemmas

In this section we provide several technical lemmas which we will use throughout the paper. Most of these lemmas appear in [BEGH16]. In order to be self-contained, we repeat the proofs in some cases.

Lemma B.1. *Let X, Y be random variables over a finite discrete domains Ω_X and Ω_Y, respectively. Then,*

1. $I(X \mid Y) = I(X) + I(X;Y)$
2. $I(X \mid Y) \leq I(X) + \log |\Omega_Y|$
3. $I(X \mid Y) \leq I(X,Y)$

where $I(X;Y) = H(X) + H(Y) - H(X,Y)$ is the mutual information between X and Y (not to be confused with $I(X,Y) = \log |\Omega_X| + \log |\Omega_Y| - H(X,Y)$).

Proof. We prove the three claims by order,

1. $$\begin{aligned} I(X \mid Y) &= \log|\Omega_X| - H(X \mid Y) \\ &= \log|\Omega_X| - H(X) + H(Y) - H(Y \mid X) \\ &= I(X) + I(X;Y). \end{aligned}$$

2. Follows from (1) and the fact that $I(X;Y) \leq \log|\Omega_Y|$.

3. $$\begin{aligned} I(X,Y) &= \log|\Omega_X| + \log|\Omega_Y| - H(X,Y) \\ &\geq \log|\Omega_X| + H(Y) - (H(Y) + H(X \mid Y)) \\ &= I(X \mid Y). \end{aligned}$$

□

Lemma B.2. *Let $Z, D, X_1, \ldots, X_n$ be random variables. Let $f : Z \to [n]$ be some function. Suppose that, conditioned on $D = d$, Z and $(X_1, \ldots, X_n)$ are independent. Denote the guessing probability $p_{\max}(f(Z) \mid D = d) = 2^{-H_\infty(f(Z)|D=d)}$, then*

$$\mathbb{E}_{z \sim Z|D=d} I(X_{f(Z)} \mid D = d, Z = z) \leq p_{\max}(f(Z) \mid D = d) \cdot I(X_1, \ldots, X_n \mid D = d).$$

Proof.

$$\begin{aligned} \mathbb{E}_{z \sim Z|D=d} I(X_{f(Z)} \mid D = d, Z = z) &= \sum_z \Pr(Z = z \mid D = d) I(X_{f(z)} \mid D = d, Z = z) \\ &= \sum_{i=1}^{n} \left(\sum_{z:f(z)=i} \Pr(Z = z \mid D = d) \right) I(X_i \mid D = d) \\ &= \sum_{i=1}^{n} \Pr(f(Z) = i \mid D = d) I(X_i \mid D = d) \\ &\leq \sum_{i=1}^{n} p_{\max}(f(Z) \mid D = d) \cdot I(X_i \mid D = d) \\ &\leq p_{\max}(f(Z) \mid D = d) \cdot I(X_1, \ldots, X_n | D = d). \end{aligned}$$

The second line follows due the fact that Z and $(X_1, \ldots, X_n)$ are independent conditioned on $D = d$, grouping together terms with the same $f(Z)$ value. The last inequality follows from the super-additivity of information (Lemma A.2). □

Lemma B.3. *Let $X_1, \ldots, X_n \geq 0$ be random variables, with expectations $\mu_i = \mathbb{E}[X_i]$, and assume that $\sum_{i=1}^{n} \mu_i \leq C$, for some constant C. Let $M(t) = \operatorname{argmin}_i\{(X_i < t)\}$ be the minimal index i for which X_i is below the threshold t. Then,*

$$\mathbb{E}[M(t)] \leq 1 + \frac{C}{t}.$$

Proof.

$$\begin{aligned}
\mathbb{E}[M(t)] &= \sum_{i=1}^{n} \Pr[M(t) \geq i] \\
&\leq 1 + \sum_{i=1}^{n} \Pr[M(t) > i] \\
&= 1 + \sum_{i=1}^{n} \Pr[(X_1 \geq t) \wedge \cdots \wedge (X_i \geq t)] \\
&\leq 1 + \sum_{i=1}^{n} \Pr[X_i \geq t] \\
&\leq 1 + \sum_{i=1}^{n} \frac{\mu_i}{t} \\
&\leq 1 + \frac{C}{t}.
\end{aligned}$$

where the penultimate inequality is due Markov's inequality. □

Lemma B.4. *Let d be a set of binary random variables, ordered as a tree of depth n. For any fixed path P of depth $i \leq n$ starting from the root of d, let $T[P]$ be the set of variables along that path, and let $p_{\max}(T[P]) = 2^{-H_\infty(T[P])}$ be the maximal probability that some assignment to $T[P]$ can obtain. For any $i \leq n$ define*

$$p_{\max}(i) = \max_{P \text{ s.t. } |P|=i} \{p_{\max}(T[P])\}.$$

Then for any $t \geq 6$ it holds that

$$\sum_{i=t}^{n} p_{\max}(i) < 2I(T) + 4\sqrt{I(T)} + 20 \cdot 2^{-t/4}.$$

This lemma is an immediate corollary of the following stronger Lemma B.5, that proves a similar claim when considering any subset S of n binary random variables. In particular, for the special case of Lemma B.4, the subset S contains variables along a single path in d (note that the parameter n in the above Lemma corresponds to $|S|$ of Lemma B.5).

Lemma B.5 ([BEGH16])**.** *Let $B = (B_1, \ldots, B_n)$ be a sequence of n random variables, where $B_i \in \{0,1\}$. For any $S \subseteq [n]$ we let $B(S) \stackrel{\text{def}}{=} \{B_i \mid i \in S\}$ be the variables indexes by S. Let $p_{\max}(S) = 2^{-H_\infty(B(S))}$ i.e., the maximal probability that $B(S)$ can attain. For $1 \leq i \leq n$, let $p_{\max}(i) = \max_{|S|=i} p_{\max}(S)$. Then it holds that for any $t \geq 6$,*

$$\sum_{i=t}^{n} p_{\max}(i) < 2I(B) + 4\sqrt{I(B)} + 20 \cdot 2^{-t/4}.$$

Lemma B.6 (Chernoff)**.** *Let $X_1, \ldots, X_n$ be independent identically distributed random variables on $\{0,1\}$ with expectation $\mathbb{E}[X_i] = \mu$. Then, for every $\delta > 0$,*

$$\Pr\left[\sum_i X_i < (1-\delta)n\mu\right] < e^{-\delta^2 n\mu/2}.$$

B.1 Independence of inputs given (a corrupted) transcript in interactive communication

A well known property of interactive communication is that conditioning on the transcript doesn't create dependencies in the inputs. That is, if the inputs were independent, they remain independent conditioned on any given (noisy) transcript (or a prefix of a transcript).

Lemma B.7 ([BEGH16])**.** *Conditioned on the observed transcript* Π*, the random variables* $X_1, \ldots, X_n$ *are mutually independent.*

Proof. The proof goes by induction on the length of Π. The base case where $|\Pi| = 0$ is trivial from the definition of the inputs $X_1, \ldots, X_n$.

Assume the claim holds for some transcript $\Pi = \pi$ of length $\ell - 1$, and consider the next observed symbol Π_ℓ, sent without loss of generality by the i-th party. This symbol (in case it was not corrupted by the channel) depends only on X_i and the previous communication Π, that is $\Pi_\ell = f(\Pi, X_i)$. To simplify notations, denote by $X_{\neq i} = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$ all the variables except X_i. We have,

$$
\begin{aligned}
&\Pr(X_1 = x_1, \ldots, X_n = x_n \mid \Pi = \pi, \Pi_\ell = \sigma) \\
&= \frac{\Pr(X_1 = x_1, \ldots, X_n = x_n, \Pi_\ell = \sigma \mid \Pi = \pi)}{\Pr(\Pi_\ell = \sigma \mid \Pi = \pi)} && \text{by definition} \\
&= \frac{\Pr(X_{\neq i} = x_{\neq i} \mid \Pi = \pi)\Pr(X_i = x_i, \Pi_\ell = \sigma \mid \Pi = \pi)}{\Pr(\Pi_\ell = \sigma \mid \Pi = \pi)} && \text{by induction, since } X_i, f(X_i, \Pi) \perp X_{\neq i} \mid \Pi \\
&= \left(\prod_{j \neq i} \Pr(X_j = x_j \mid \Pi = \pi)\right) \frac{\Pr(X_i = x_i, \Pi_\ell = \sigma \mid \Pi = \pi)}{\Pr(\Pi_\ell = \sigma \mid \Pi = \pi)} \\
&= \prod_{j \neq i} \Pr(X_j = x_j \mid \Pi = \pi, \Pi_\ell = \sigma) \times \Pr(X_i = x_i \mid \Pi = \pi, \Pi_\ell = \sigma),
\end{aligned}
$$

where the last transition follows since X_i and $X_{\neq i}$ are independent given Π, thus conditioning on a function of either X_i or Π does not change the probability. Finally, note that if the symbol σ was erased, the claim trivially holds. □

As a corollary to the above, note that the inputs remain independent when conditioned on any information that can be communicated by the parties by some protocol. For instance, parts of the inputs or path up to some level, etc.

Corollary B.8. *The random variables* $X_1, \ldots, X_n$ *are independent, conditioned on the observed transcript* $\Pi = \pi$*, the correct path* $\mathsf{PATH} = \mathsf{path}$ *(up to some level* $\ell \leq d$*), and parts of the inputs (i.e., any fixed set of edges from* $X_{[n]}$*).*

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il