Constant-Rate Coding for Multiparty Interactive Communication Is Impossible

Mark Braverman*
Princeton University
New-Jersey, USA
mbraverm@cs.princeton.edu

Klim Efremenko†
Tel-Aviv University
Israel
klimefrem@gmail.com

Ran Gelles
Princeton University
New-Jersey, USA
rgelles@cs.princeton.edu

Bernhard Haeupler
Carnegie Mellon University
Pennsylvania, USA
haeupler@cs.cmu.edu

ABSTRACT

We study coding schemes for multiparty interactive communication over synchronous networks that suffer from stochastic noise, where each bit is independently flipped with probability ε . We analyze the minimal overhead that must be added by the coding scheme in order to succeed in performing the computation despite the noise.

Our main result is a lower bound on the communication of any noise-resilient protocol over a synchronous star network with n -parties (where all parties communicate in every round). Specifically, we show a task that can be solved by communicating T bits over the noise-free network, but for which any protocol with success probability of $1 - o(1)$ must communicate at least $\Omega(T \frac{\log n}{\log \log n})$ bits when the channels are noisy. By a 1994 result of Rajagopalan and Schulman, the slowdown we prove is the highest one can obtain on any topology, up to a $\log \log n$ factor.

We complete our lower bound with a matching coding scheme that achieves the same overhead; thus, the capacity of (synchronous) star networks is $\Theta(\log \log n / \log n)$. Our bounds prove that, despite several previous coding schemes with rate $\Omega(1)$ for certain topologies, no coding scheme with constant rate $\Omega(1)$ exists for arbitrary n -party noisy networks.

*Research supported in part by an NSF CAREER award (CCF-1149888), NSF CCF-1525342, a Packard Fellowship in Science and Engineering, and the Simons Collaboration on Algorithms and Geometry.

†The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257575.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

STOC'16, June 19–21, 2016, Cambridge, MA, USA
ACM. 978-1-4503-4132-5/16/06...\$15.00
<http://dx.doi.org/10.1145/2897518.2897563>

Categories and Subject Descriptors

E.4 [Coding and Information Theory]; F.2.0 [Theory of Computation]: Analysis of Algorithms and Problem Complexity—*Communication Complexity*

General Terms

Theory

Keywords

Interactive communication, Coding theory, Stochastic noise, Communication complexity, Rate, Network topology

1. INTRODUCTION

Assume a network of n remote parties who perform a distributed computation of some function of their private inputs, while their communication may suffer from stochastic noise. The task of *coding for interactive communication* seeks for coding schemes that allow the parties to correctly compute the needed function while limiting the overhead occurred by the coding. For the two party case, $n = 2$, Schulman, in a pioneering line of results [31, 32, 33], showed how to convert any protocol that takes T rounds when the communication is noiseless, into a resilient protocol that succeeds with high probability and takes $O(T)$ rounds when the communication channel is a *binary symmetric channel (BSC)*¹, that is, when the communication may suffer from random noise.

For the general case of n parties, Rajagopalan and Schulman [30] showed a coding scheme that succeeds with high probability and takes $O(T \log n)$ rounds in the worst case. Here, a “round” means simultaneous communication of a single bit over each one of the channels. More precisely, the communication of the coding scheme in [30] depends on the specific way the parties are connected to each other. Specifically, the scheme takes $O(T \log(d + 1))$ rounds, where d is the maximal number of neighbors a party may have. Thus, for certain topologies like a line or a cycle, the slowdown is constant $O(1)$, however in the worst case, i.e., when the

¹The BSC channel, parametrized by a probability ε , flips each bit independently with probability ε , and leaves the bit unflipped with probability $1 - \varepsilon$.

topology is a complete graph, the scheme has a slowdown of $O(\log n)$.

The work of Alon et al. [2] shows how to improve the $O(\log n)$ slowdown when the network's topology is a complete graph. Specifically, they provide a coding scheme with high probability of success and slowdown of $O(1)$ for a rich family of "highly connected" topologies, including the complete graph. Therefore, a constant-slowdown coding scheme is achievable either when the degree is constant [30], or when the connectivity is large [2], i.e., when many disjoint paths connect every two parties.

The main outstanding open question left by these works is whether a constant-slowdown coding scheme can be obtained for *all* topologies. We answer this question in the negative and show a lower bound on the overhead of any coding scheme with high probability of success, over a star network:

THEOREM 1.1 (MAIN, LOWER BOUND). *Assume n parties connected as a star, and let $\varepsilon < 1/2$ be given. There exists an n -party protocol χ that takes T rounds assuming noiseless channels, such that any coding scheme that simulates χ with probability above $1/5$ when each channel is a BSC_ε , takes $\Omega(T \frac{\log n}{\log \log n})$ rounds.*

By making "running χ " the interactive task to be performed, Theorem 1.1 implies the $\Omega(\frac{\log n}{\log \log n})$ slowdown in interactive coding. By [30], our result is tight up to an $O(\log \log n)$ factor, since all topologies admit a scheme with a $O(\log n)$ slowdown. On the other hand, we show that coding with a slowdown of $O(\frac{\log n}{\log \log n})$ is achievable and therefore tight for interactive coding over a star topology.

THEOREM 1.2 (UPPER BOUND). *Assume n parties connected as a star, and let $\varepsilon < 1/2$. For any n -party protocol χ that takes T rounds assuming noiseless channels, there exists a coding scheme that simulates χ assuming each channel is a BSC_ε , takes $N = O(T \frac{\log n}{\log \log n})$ rounds, and succeeds with probability $1 - 2^{-\Omega(N)}$.*

The upper bound follows quite straightforwardly from an observation by Alon et al. [2], showing that as long as one round of the noiseless χ can be simulated with high probability, then the entire protocol χ can be simulated with a high probability by employing the techniques of [30]. Over a star, it is quite simple to simulate $\log \log n$ rounds of an arbitrary noiseless χ using only $O(\log n)$ noisy rounds, with high probability. Thus, we can apply the technique of [2, 30] on segments of $\log \log n$ rounds of χ , and achieve the stated coding scheme. We prove Theorem 1.2 in Section 4.

We devote Section 5 to sketch the proof of the more involved lower bound of Theorem 1.1. Full details can be found in the full version of this manuscript [6]. Below we give a rather intuitive overview of our lower bound result and the techniques we use.

1.1 Lower Bound: Overview and Techniques

In order to achieve our lower bound of $\Omega(\frac{\log n}{\log \log n})$ on the overhead, we consider protocols for the *pointer jumping task* of depth T , between n parties (also called *clients*) and the center of the star (also called the *server*). In the pointer jumping task, each client gets as an input a binary tree of depth T , where each edge is labeled with a single bit. The server's input is a 2^n -ary tree of depth T where each edge is labeled with an n -bit string. Solving the pointer jumping

task is equivalent to performing the following protocol: all parties begin from the root of their trees. At each round, simultaneously for $1 \leq i \leq n$, the i -th client receives a bit b_i from the center and descends in his tree to the b_i -th child of its current node. The client then sends back to the server the label of the edge through which it traversed. The server receives, at each round, the string $B = b_1 \cdots b_n$ from the clients and descends to the B -th child of its current node. If the edge going to that node is labeled with the n -bit string $b'_1 \cdots b'_n$, then the server sends b'_i to the i -th client. The process then repeats, until the parties reach the T -th level in their respective tree. At the end, each party outputs the leaf it has reached (or equivalently, it outputs the "path" it traversed). Note that the T -level pointer jumping task can be solved using $2T$ rounds of alternating noiseless communication. By *alternating* we mean here that the server speaks on, say, odd rounds, while the clients speak on even rounds. Also note that the pointer jumping task is complete for interactive communication, i.e., any interactive protocol for $n + 1$ parties connected as a star can be represented as a specific input-instance of the above pointer jumping task. See Section 3 for further details about the multiparty pointer jumping task.

Next we assume the channels are noisy. In fact, we can weaken the noise model and assume that the noise erases bits rather than flipping them, that is, we consider the binary erasure channel, BEC_ε ; see Definition 2.1. Note that since the considered noise model is *weaker*, our lower bound becomes *stronger*.

Consider any protocol that solves the pointer jumping task of depth T assuming the channels are $\text{BEC}_{1/3}$. We divide the protocol into segments of length $0.1 \log n$ rounds each and show that at each such segment the protocol "advances" by at most $O(\log \log n)$ levels in the underlying pointer jumping task, in expectation. Very roughly, the reason for this slow progress follows from the observation that during each segment of $0.1 \log n$ rounds, with high probability there exists a set of $\sqrt{n}$ clients whose communication was completely erased. It follows that the server is missing knowledge on $\sqrt{n}$ parties and thus cannot infer its next node with high probability. On average, the server sends a very small amount of information on the labels descending from its current node that belong to the "correct" path. As a result, the clients practically receive no meaningful information on the next level(s) of the server. This in turn limits the amount of information they can send on *their* "correct" paths to $O(\log \log n)$ bits in expectation, thus limiting the maximal advancement in the underlying pointer jumping task. For instance, if some client who does not know the correct path in his input pointer-jumping tree, communicates to the server all the labels descending from its current node, say in a breadth-first manner, the information sent during $0.1 \log n$ rounds can contain at most $O(\log \log n)$ levels of this client's *correct* path.

Not surprisingly, the technical execution of the above strategy requires tools for careful and accurate bookkeeping of the information the parties have learned at any given time of the (noisy) execution. The basic definition of information a party has about a random variable X sampled from a space Ω_X we employ is $I(X) \stackrel{\text{def}}{=} \log |\Omega_X| - H(X)$, where $H(X)$ is Shannon's entropy of X given the party's current knowledge. Note that if *a-priori* X is uniformly distributed, $I(X)$ is exactly the mutual information between what the

party knows and X . However, our information notion behaves more nicely under conditioning (i.e. when changing what the party knows about X as the protocol progresses), and seems generally easier to work with.

A central notion in our analysis is the *cutoff* round of the protocol, which relates to the deepest level of the underlying pointer jumping task that the parties can infer from the communication they have received so far. Very roughly, if the cutoff is k , then parties have small information on labels below level k in the underlying tree of the party (or parties) connected to them. More precisely, for any (partial) transcript π the parties observe, we define $\text{cutoff}(\pi)$ to be the minimal round $1 \leq k \leq T$ for which the parties have small amount of information about labels in the underlying pointer jumping task that lie in the subtree rooted at the end of the correct path of depth k , conditioned on the transcript π and on the correct path up to level k (see Definition 5.2 for the exact formulation).

The core of our analysis shows that, given a certain cutoff, $\text{cutoff}(\pi) = \ell$, and assuming the parties communicate the next $0.1 \log n$ rounds of the protocol (denote the observed transcript in this new part as Π^{new}), then in expectation over the possible inputs, noise, and randomness of the protocol, the cutoff does not increase by more than $O(\log \log n)$; that is,

$$\mathbb{E}[\text{cutoff}(\pi, \Pi^{\text{new}}) \mid \text{cutoff}(\pi) = \ell] \leq \ell + O(\log \log n).$$

This implies that, unless the protocol runs for $\Omega(T \frac{\log n}{\log \log n})$ rounds, then the cutoff at the end of the protocol is substantially smaller than T , with high probability. Using Fano's inequality, this in turn implies that the protocol cannot output the correct path (beyond the cutoff round) with high probability.

Bounding the information revealed by the parties at each step is the deepest technical contribution of this paper, and is done in methods which are close at spirit to a technique by Kol and Raz [26] for obtaining lower bounds in the two-party case.² We bound separately the information that the server reveals and the information the clients reveal in each segment of $0.1 \log n$ rounds (conditioned on a given cutoff level, i.e., on the transcript of the protocol so far and on the correct path up to the cutoff level).

Very informally, we show that the information revealed during a single chunk on labels below a continuation of the correct path (i.e., the information captured by the “new” cutoff), can be bounded by the product of (i) the probability to guess the continuation of the correct path (between the current and the new cutoff levels), and (ii) the information that the transcript so far contains on all the labels (either on the correct path or not) that lie below the new cutoff level. Indeed, if a party wants to give information about the labels of its correct path, but that party doesn't know the correct path, it can't do much more than guessing the path and sending information about that guess; alternatively, it can give information on labels in all possible paths, where the amount of information of each label corresponds to the probability of this label to be part of the correct path.

We bound each one of the above terms separately. For the first part (i), we bound the guessing probability of a continuation of the correct path as a function of the informa-

tion the observed transcript contains on the labels below the current cutoff *in the tree of the other parties*. For instance, guessing the correct path in the server's tree is bounded by the amount of information the transcript gives on labels along the correct path in the clients' trees, at the same levels (because these labels exactly determine the path the server should take in his tree). The definition of the cutoff and the fact that these levels lie below the cutoff level, give a bound the amount of information we have on these labels, which can be translated to a bound on the probability of guessing the corresponding path. Fano's inequality is not strong enough to our needs (i.e., sub-exponential guessing probability from sub-exponentially small information), and we devise a tighter bound via a careful analysis of the positive and negative parts of the Kullback–Leibler divergence; see Lemma 2.13. This (entropy vs. min-entropy) relation may be of independent interest.

To bound the second part (ii), we observe that the information on labels below the current cutoff is bounded *in expectation* using the definition of the cutoff, up to possibly additional $0.1n \log n$ bits that were communicated during the new segment of $0.1 \log n$ rounds.

The fact that the bound of part (ii) works only in expectation is a major hurdle, because it prevents us from bounding the above product directly (these two multiplicands are dependent!). We detour around this issue by narrowing down the probability space by conditioning on additional information that makes the two multiplicands independent. As conditioning potentially increases the information we wish to bound, it is essential to carefully limit the amount of additional information we condition on, so that the bound remains meaningful. Giving more details (yet still very intuitively speaking), we condition on all the labels that lie between the old and new cutoff levels, of either the server's input *or* the clients' input, according to the specific information we are currently bounding. We show that this conditioning only increases the information, thus bounding the conditioned version in expectation, also bounds the unconditioned information that we care about. This conditioning, however, takes out the dependency caused by the interaction (since the labels of one side are fixed up to some given level) and makes the labels below the new cutoff independent of labels above it; specifically, the correct path between the current and the new cutoff (which is involved in the first multiplicand) is conditionally independent of the labels below the new cutoff (which are involved in the second one). This independence allows us to bound the expectation of the above product by bounding each term separately as described above.

1.2 Related Work

As mentioned above, coding for interactive communication in the presence of random noise was initiated by Schulman for the two-party case [31, 32, 33]. The coding scheme of Schulman [32, 33] achieves slowdown of $O(1)$ and exponentially high success probability; however, it is not computationally efficient and can take exponential time in the worst case. Gelles, Moitra, and Sahai [16, 17] showed how to obtain an efficient coding scheme while maintaining a constant slowdown and exponentially high success probability. Braverman [4] gave another efficient coding scheme, yet with a slightly reduced success probability. Other related work in the two party setting considers the case of adversarial noise rather than random noise, in various settings [7, 3, 11, 9, 1, 18, 19,

²In fact, it is an interesting question whether our techniques can be used to simplify the analysis in [26].

5, 21, 20, 10, 15]; see [13] for a survey.

In the two-party setting, the minimal possible slowdown over a BSC_ε as a function of the noise parameter ε , was initially considered by Kol and Raz [26], who showed a lower bound of $1 + \Omega(\sqrt{\varepsilon \log 1/\varepsilon})$ on the slowdown. Later, Haeupler [23] showed that the order in which the parties are speaking affects the slowdown, and if the parties are assumed to be alternating, a slowdown of $1 + O(\sqrt{\varepsilon})$ is achievable. When the noise is adversarial rather than random, the slowdown increases to $1 + O(\sqrt{\varepsilon \log \log 1/\varepsilon})$ [23]. The slowdown in other types of channels, such as the binary erasure channel BEC_ε or channels with noiseless feedback, was considered by Gelles and Haeupler [14], who showed efficient coding schemes with an optimal slowdown of $1 + \Theta(\varepsilon \log 1/\varepsilon)$ over these channels.

As for the multiparty case, the work of Rajagopalan and Schulman [30] was the first to give a coding scheme for the case of random noise over arbitrary topology, with a slowdown of $O(\log(d+1))$ for d the maximal degree of the connectivity graph. As in the two-party case, that scheme is not efficient, but can be made efficient using [16, 17]. Alon, Braverman, Efremenko, Gelles, and Haeupler [2] considered coding schemes over d -regular graphs with mixing time³ m , and obtained a slowdown of $O(m^3 \log m)$. This implies a coding scheme with a constant slowdown $O(1)$ whenever the mixing time is constant, $m = O(1)$, e.g., over complete graphs.

For the case of adversarial noise in the multiparty setting, Jain, Kalai, and Lewko [25] showed an asynchronous coding scheme for star topologies with slowdown $O(1)$ for up to $O(1/n)$ -fraction of noise. A communication-balanced version of that scheme was given by Lewko and Vitercik [28]. Hoza and Schulman [24] showed a coding scheme in the synchronous model that works for any topology, tolerates $O(1/n)$ -fraction of noise, and demonstrates a slowdown of $O(\frac{m}{n} \log n)$ where m here is the number of edges in the given connectivity graph.

Finally, we mention the work of Gallager [12]. This work assumes a different setting than the above works, namely, the case where parties are all connected via a noisy broadcast channel (the noisy blackboard model). Gallager showed that a slowdown of $O(\log \log n)$ is achievable for the task where each party begins with a bit and needs to output the input bits of all other parties. Goyal, Kindler, and Saks [22] showed that this slowdown is tight by providing a matching slowdown of $\Omega(\log \log n)$ for the same task in the noisy broadcast model. It is not clear whether there is a direct connection between results in these two models—there does not seem to be a way to translate results in either direction.

1.3 Open Questions

It is already well established that topology matters in communication [8] and in network coding [29]. Our work (along with previous results [30, 2]) suggests that the same holds also for the field of interactive communication when the noise is random. While for certain topologies (e.g., a line, a cycle, a complete graph) one can achieve a coding scheme with slowdown $O(1)$, other topologies necessitate an overhead of $\Theta(\log n / \log \log n)$, e.g. the star topology. The

main open question is to better characterize the way topology affects slowdown.

OPEN QUESTION 1. *For any function $f(n) \in o(\log n)$, define the exact set of topologies for which n -party interactive coding schemes with $f(n)$ slowdown exist. In particular, characterize the set of topologies for which n -party interactive coding schemes with $O(1)$ slowdown exist.*

While [30] shows that, given any topology, interactive coding with $O(\log n)$ slowdown exists, our lower bound demonstrates a necessary slowdown of only $\Omega(\log n / \log \log n)$. This gap leads to the following question:

OPEN QUESTION 2. *Show a topology (if such exists) for which $\Omega(\log n)$ slowdown is necessary for n -party interactive coding.*

Currently, we do not have a candidate topology for showing an overhead of $\omega(\log n / \log \log n)$.

2. PRELIMINARIES

For $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, 2, \dots, n\}$. The $\log()$ function is taken to base 2. We denote the natural logarithm by $\ln()$.

2.1 Coding Over Noisy Networks

Given an undirected graph $G = (V, E)$ we assume a network with $n = |V|$ parties, where $u, v \in V$ share a communication channel if $(u, v) \in E$. In the case of a noisy network, each such link is assumed to be a BSC_ε or a BEC_ε .

DEFINITION 2.1. *For $\varepsilon \in [0, 1]$ we define the binary symmetric channel $\text{BSC}_\varepsilon : \{0, 1\} \rightarrow \{0, 1\}$ in which the input bit is flipped with probability ε , and remains the same with probability $1 - \varepsilon$. The binary erasure channel $\text{BEC}_\varepsilon : \{0, 1\} \rightarrow \{0, 1, \perp\}$, turns each input bit into an erasure mark $\perp$ with probability ε , or otherwise keeps the bit intact. When a channel is accessed multiple times, each instance is independent.*

A *round* of communication in the network means the simultaneous transmission of $2|E|$ messages: for any $(u, v) \in E$, u sends a bit to v and receives a bit from v . A protocol for an n -party function $f(x_1, \dots, x_n) = (y_1, \dots, y_n)$, is a distributed algorithm between n parties $\{p_1, \dots, p_n\}$, where each p_i begins the protocol with an input x_i , and after N rounds of communication outputs y_i . The communication complexity of a protocol, $\text{CC}()$, is the number of bits sent throughout the protocol. Note that given any network G , the round complexity of a protocol and its communication complexity differ by a factor of $2|E|$.

Assume χ is a protocol over a *noiseless* network G . We say that a protocol χ' simulates χ over a channel C with rate R if, when χ' is run with inputs $(x_1, \dots, x_n)$ over the network G where each communication channel is C , the parties output $\chi(x_1, \dots, x_n)$ with high probability and it holds that $\text{CC}(\chi)/\text{CC}(\chi') = R$. We also use the terms *slowdown* and *overhead* to denote the inverse of the rate, R^{-1} , that is, the (multiplicative) increase in the communication due to the coding.

2.2 Information, Entropy, and Min-Entropy

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

³Intuitively speaking, the mixing time of a graph is the minimal number of steps a random walk needs in order to end up at every node with approximately equal probability.

DEFINITION 2.2 (INFORMATION). Let X be a random variable over a finite discrete domain Ω . The information of X is given by

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

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

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

$$I(X | Y) \stackrel{\text{def}}{=} \log |\Omega| - H(X | Y) \\ = \mathbb{E}_Y I(X | Y = y).$$

LEMMA 2.3 (SUPERADDITIVITY OF INFORMATION). Let $X_1, \dots, X_n$ be n random variables. Then,

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

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

PROOF. Immediate from the subadditivity of the entropy function. $\square$

LEMMA 2.4. Let X, Y be random variables over a finite discrete domains Ω_X and Ω_Y , respectively. Then,

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

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. $I(X | Y) = \log |\Omega_X| - H(X | Y) \\ = \log |\Omega_X| - H(X) + H(Y) - H(Y | X) \\ = I(X) + I(X; Y).$
2. Follows from (1) and the fact that $I(X; Y) \leq \log |\Omega_Y|$.
3. $I(X, Y) = \log |\Omega_X| + \log |\Omega_Y| - H(X, Y) \\ \geq \log |\Omega_X| + H(Y) - (H(Y) + H(X | Y)) \\ = I(X | Y).$

$\square$

DEFINITION 2.5 (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., $p_{\max}(X) \stackrel{\text{def}}{=} \max_{x \in \Omega} \Pr(X = x)$. At times, $p_{\max}$ is called the guessing probability of X .

We relate information (or, entropy) with the guessing probability (or, min-entropy) via the next Lemma, which is a special case of Fano's inequality.

LEMMA 2.6. 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.

PROOF. The lemma is an immediate corollary of the following version of Fano's inequality,

$$H(X) \leq \log |\Omega| (1 - 2^{-H_\infty(X)}) + h(2^{-H_\infty(X)}). \quad (1)$$

Let us prove Eq. (1). Assume without loss of generality that $\Omega = \{1, \dots, n\}$. Let $p_i = \Pr(X = i)$, and again assume without loss of generality that for any $i < j$, it holds that $p_i \geq p_j$. Thus, $p_{\max}(X) = p_1$. If $p_1 = 1$, the claim is trivial. Otherwise,

$$H(X) = p_1 \log \frac{1}{p_1} + \sum_{i=2}^n p_i \log \frac{1}{p_i}. \quad (2)$$

Define Y to be distributed over $\{2, \dots, n\}$ with probabilities $\Pr(Y = i) = p_i/(1 - p_1)$. Note that this is a valid distribution as all the probabilities are non-negative, and $\sum_i \Pr(Y = i) = 1$. Note that

$$H(Y) = \sum_{i=2}^n \frac{p_i}{1 - p_1} \log \frac{1 - p_1}{p_i} \\ = \left(\sum_{i=2}^n \frac{p_i}{1 - p_1} \log \frac{1}{p_i} \right) - \log \frac{1}{1 - p_1}.$$

Going back to Eq. (2), we have

$$H(X) = p_1 \log \frac{1}{p_1} + (1 - p_1) H(Y) + (1 - p_1) \log \frac{1}{1 - p_1} \\ \leq h(p_1) + (1 - p_1) \log |\Omega|,$$

which holds because $H(Y) \leq \log(|\Omega| - 1) < \log |\Omega|$. Then Eq. (1) and the lemma follow by substituting $p_1 = p_{\max}(X) = 2^{-H_\infty(X)}$. $\square$

We note here that similar claims to the above lemmas hold when we additionally condition on some event $\mathcal{E}$; indeed, one can apply these lemmas on the random variable $(X | \mathcal{E})$.

Another key tool we use is the Kullback-Leibler divergence.

DEFINITION 2.7 (KL-DIVERGENCE [27]). Let X, Y be random variables over a discrete domain Ω . The KL-divergence of X and Y is

$$D(X \| Y) \stackrel{\text{def}}{=} \sum_{\omega \in \Omega} \Pr(X = \omega) \log \left(\frac{\Pr(X = \omega)}{\Pr(Y = \omega)} \right).$$

Define $\Omega^+ = \{\omega \in \Omega : \Pr(X = \omega) > \Pr(Y = \omega)\}$ and $\Omega^- = \Omega \setminus \Omega^+$. We can split the KL-Divergence into its positive and negative parts,

$$D(X \| Y) = D^+(X \| Y) - D^-(X \| Y),$$

where $D^+(X \| Y) = \sum_{\omega \in \Omega^+} \Pr(X = \omega) \log \left(\frac{\Pr(X = \omega)}{\Pr(Y = \omega)} \right)$ and $D^-(X \| Y) = - \sum_{\omega \in \Omega^-} \Pr(X = \omega) \log \left(\frac{\Pr(X = \omega)}{\Pr(Y = \omega)} \right)$.

LEMMA 2.8. Let X, Y be random variables over a discrete domain Ω . Then, for every $\Omega' \subseteq \Omega$ it holds that

$$\sum_{\omega \in \Omega'} \Pr(X = \omega) \log \left(\frac{\Pr(X = \omega)}{\Pr(Y = \omega)} \right) \leq D^+(X \| Y).$$

PROOF. Immediate from the definition of $D^+(\cdot\|\cdot)$. $\square$

LEMMA 2.9 (PINSKER INEQUALITY). *Let X, Y be random variables over a discrete domain Ω , then*

$$\|X - Y\|^2 \leq 2 \ln(2) \cdot D(X\|Y),$$

where $\|X - Y\| = \sum_{\omega \in \Omega} |\Pr(X = \omega) - \Pr(Y = \omega)|$.

We now upper bound the negative part of the KL-Divergence. Note that one can easily show that $D^-(X\|Y) \leq 1$, but we will need a better upper bound that applies in the case where $D^-(X\|Y) \ll 1$.

LEMMA 2.10.

$$D^-(X\|Y) \leq \sqrt{\frac{2}{\ln(2)} D(X\|Y)}.$$

PROOF. For every $\omega \in \Omega$, let $p_\omega \stackrel{\text{def}}{=} \Pr(X = \omega)$ and $q_\omega \stackrel{\text{def}}{=} \Pr(Y = \omega)$. We can relate any negative term of the divergence with a difference of probabilities via the following claim:

CLAIM 2.11. *For $p_\omega \leq q_\omega$ it holds that $\ln(2)p_\omega \log \frac{q_\omega}{p_\omega} \leq q_\omega - p_\omega$.*

PROOF. Note that the equality holds for $p_\omega = q_\omega$. If we take the derivative with respect to q_ω , the LHS is $\frac{p_\omega}{q_\omega}$ and the RHS is 1. Since $\frac{p_\omega}{q_\omega} \leq 1$ when $p_\omega \leq q_\omega$, the claim holds. $\square$

Note that by definition, $D^-(X\|Y) = \sum_{\omega: p_\omega \leq q_\omega} p_\omega \log \frac{q_\omega}{p_\omega}$. From the claim above it holds that

$$D^-(X\|Y) \leq \frac{1}{\ln(2)} \|X - Y\|.$$

The lemma then follows from Pinsker's inequality (Lemma 2.9).

2.3 Technical Lemma: Bounding Min-Entropy Via Entropy

The following lemma is central in the proof of the lower bound, and may be of independent interest. It allows bounding the probability of guessing a sequence of random variables as a function of the information known about the sequence.

LEMMA 2.12. *Let T 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 T , 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 $t \geq 0$ 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 2.13, that proves a similar claim when considering any subset S of n binary random variables. In particular, for the special case of Lemma 2.12, the subset S contains variables along a single path in T (note that the parameter n in the above Lemma corresponds to $|S|$ of Lemma 2.13).

LEMMA 2.13. *Let $B = (B_1, \dots, B_n)$ 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 indexed 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 $t \geq 0$,*

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

PROOF. For any n -bit string s we define $p_s \stackrel{\text{def}}{=} \Pr(B = s)$ the probability that B attains the value s . For any $1 \leq i \leq n$, we fix S_i and $\beta_i = b_1 b_2 \dots b_i$ to be a specific subset of size i of variables and its assignment that attains the maximal probability, i.e., for which $\Pr(B(S_i) = \beta_i) = p_{\max}(i)$. Define $V_i \stackrel{\text{def}}{=} \{s \in \{0, 1\}^n \mid s(S_i) = \beta_i\}$ as the set of all the binary strings s of length n whose restriction to S_i equals to β_i . Define

$$W_i \stackrel{\text{def}}{=} V_i \setminus \left(\bigcup_{j>i} V_j \right)$$

to be the set of all the strings s such $s(S_i) = \beta_i$, but for any $j > i$, $s(S_j) \neq \beta_j$. Let $w_i \stackrel{\text{def}}{=} \sum_{s \in W_i} p_s$. Note that $W_1, W_2, \dots, W_n$ are disjoint and that $V_i = \bigcup_{j=i}^n W_j$, thus it is clear that $p_{\max}(i) \leq \sum_{j=i}^n w_j$. Therefore it holds that

$$\sum_{i=t}^n p_{\max}(i) = \sum_{i=t}^n \sum_{j=i}^n w_j = \sum_{j=t}^n (j-t+1) \cdot w_j \leq \sum_{j=t}^n j \cdot w_j. \quad (3)$$

Next we want to give an upper bound on $\sum_{j=t}^n j w_j$. Recall that any convex function f satisfies $\sum_i p_i f(x_i) \geq w f(\sum_i p_i x_i / w)$ where for any i , $p_i > 0$ and $\sum_i p_i = w$. For a given j , consider the sum $\sum_{s \in W_j} p_s \log(2^n p_s)$. From the convexity of the log function we thus get,

$$\begin{aligned} \sum_{s \in W_j} p_s \log(2^n p_s) &\geq \left(\sum_{s \in W_j} p_s \right) \log \left(2^n \frac{\sum_{s \in W_j} p_s}{|W_j|} \right) \\ &= w_j \log \left(\frac{2^n}{|W_j|} w_j \right) \geq w_j \log(2^j w_j), \end{aligned}$$

where the last inequality holds since $|W_j| \leq 2^{n-j}$. Therefore,

$$\begin{aligned} \sum_{j=t}^n \sum_{s \in W_j} p_s \log(2^n p_s) &\geq \sum_{j=t}^n w_j \log(2^j w_j) \\ &= \sum_{j=t}^n j w_j + \sum_{j=t}^n w_j \log w_j. \end{aligned} \quad (4)$$

CLAIM 2.14. $\sum_{j=t}^n w_j \log(1/w_j) \leq 10 \cdot 2^{-t/4} + \frac{1}{2} \sum_{j=t}^n j w_j$.

PROOF. Split the sum to indices where $w_j < 2^{-j/2}$ and indices where $w_j \geq 2^{-j/2}$,

$$\begin{aligned} &\sum_{j=t, w_j \geq 2^{-j/2}} w_j \log(1/w_j) + \sum_{j=t, w_j < 2^{-j/2}} w_j \log(1/w_j) \\ &\leq \frac{1}{2} \sum_{j=t}^n j w_j + \frac{1}{2} \sum_{j=t}^n \frac{j}{2^{j/2}} \leq \frac{1}{2} \sum_{j=t}^n j w_j + 10 \cdot 2^{-t/4}, \end{aligned}$$

where the last inequality is just a rough bound, and the inequality just before it follows since $w_j \log 1/w_j$ is increasing in interval $(0, e^{-1})$. $\square$

Thus from Claim 2.14 it follows that

$$\begin{aligned} & \sum_{j=t}^n jw_j + \sum_{j=t}^n w_j \log w_j \\ & \geq \sum_{j=t}^n jw_j - (10 \cdot 2^{-t/4} + \frac{1}{2} \sum_{j=t}^n jw_j) \\ & \geq \frac{1}{2} \sum_{j=t}^n jw_j - 10 \cdot 2^{-t/4}. \end{aligned}$$

Equation (4) then implies that,

$$\sum_{j=t}^n jw_j \leq 2 \sum_{j=t}^n \sum_{s \in W_j} p_s \log(2^n p_s) + 20 \cdot 2^{-t/4}.$$

Since the W_j are disjoint, we have (via Lemma 2.8) that

$$\begin{aligned} \sum_{j=t}^n \sum_{s \in W_j} p_s \log(2^n p_s) & \leq D^+(B \| U_n) \\ & = D(B \| U_n) + D^-(B \| U_n). \end{aligned}$$

Recall that $D(B \| U_n) = I(B)$ by definition. From Lemma 2.10 it follows that $D^-(B \| U_n) \leq \sqrt{\frac{2}{\ln 2} I(B)}$. Thus we get that

$$\sum_{j=t}^n jw_j \leq 2I(B) + \sqrt{\frac{8}{\ln 2} I(B)} + 20 \cdot 2^{-t/4}.$$

The above and Eq. (3) complete the proof.

3. MULTIPARTY INTERACTIVE COMMUNICATION OVER NOISY NETWORKS

In the following we assume a network of $n + 1$ parties that consists of a server p_S and n clients $p_1, \dots, p_n$. The network consists of a communication channel (p_i, p_S) for every $i \in [n]$, that is, the topology is a star.

3.1 The Pointer Jumping Task

We assume the parties want to compute a generalized *pointer jumping task*. Formally, the pointer jumping task of depth T over star-networks is the following. Each client p_i holds a binary tree x_i of depth T where each edge is labelled by a bit b . The server holds a 2^n -ary tree x_S of depth T where each edge of the tree is labeled with an n -bit string from $\{0, 1\}^n$.

The server starts from the root of x_S . At each round, the server receives from the clients n bits which it interprets as an index $i \in [2^n]$. The server then transmits back to the clients the label on the i -th edge descending from his current node (one bit per client). The node at the end of this edge becomes the server's new node. Similarly, each client receives at each round a bit b from the server, and sends back the label of the edge indexed by b descending from its current node. For the first round, we can assume that the clients take the left child of the root of x_i and transmit to the server the label of that edge. The above is repeated until both the server and the clients have reached depth T in their trees. The parties then output the path from the root to their current node (i.e., to a leaf at depth T).

We denote this “correct” output of party p_i by path_i . The entire output is denoted $\text{path} = (\text{path}_S, \text{path}_1, \dots, \text{path}_n)$. For a certain party $i \in [n] \cup \{S\}$, and a level $1 \leq k \leq T$ we let $\text{path}_i(k)$ be the first k edges of path_i .

We use the following notations throughout. 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 $z = (e_1, e_2, \dots)$, 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 . [For instance, many times z will be the correct path so far (e.g., until some round ℓ) in the input tree x_i ; then we will care about the subtrees $x_i[\text{path}_i(\ell)]$, effectively obtaining a new instance of the pointer jumping task, with a smaller depth.] We let $x = (x_S, x_1, \dots, x_n)$ be the entire input and also use the short notation $x = (x_S, x_{[n]})$ for the server's and clients' part, respectively. The above notation composes in a straightforward way, e.g., $x^{\leq k}$ and $x_{[n]}^{\leq k}$ denote the appropriate set of partial trees in x and $x_{[n]}$, respectively, and $x[\text{path}(\ell)]$ denotes the set of subtrees $x_i[\text{path}_i(\ell)]$. We will sometimes be negligent and write $x_i[\text{path}(\ell)]$ for $x_i[\text{path}_i(\ell)]$. See Figure 1 for an illustration of some of the notations.

(a) A possible input x_i of some client p_i ; $\text{path}_i(3)$ is marked with bold edges.

(b) A possible input x_S of the server

Figure 1: An illustration of the inputs, the “correct” path (marked with bold lines) and the sub-input conditioned on a partial correct path.

The above pointer jumping task is complete for the case of a star network. That is, any noiseless protocol over a star network can be described as a pointer jumping task by setting the inputs $(x_S, x_1, \dots, x_n)$ appropriately. For our purpose we will have the inputs distributed randomly. That is, for every client, the label on each edge is distributed uniformly in $\{0, 1\}$ independently of all other edges; for the server, each label is uniform and independent over $\{0, 1\}^n$. We denote the random variable describing the input of p_i by X_i . The correct path also becomes a random variable

which we denote PATH_i and which is a function of the inputs. The same holds for the subtree of a certain input, given the certain path of some depth ℓ , etc.

Lastly, we denote by π an observed transcript (possibly noisy) of the protocol. That is, π is the string *received* by the parties (in some natural order); note that no single party observes the entire transcript π , but each party observes some part of it. The corresponding random variable is denoted Π . At times π will denote a partial transcript, that is, the communication observed by the parties up to some round k of the protocol.

3.2 Inputs Independence Conditioned on the Transcript

An important property that will be needed for our lower bound, is the fact that the inputs of the users are independent, *even when conditioned on the transcript so far*. This implies that only party p_i is capable of sending useful information about its input x_i , regardless of the transcript so far (and therefore, if the communication of p_i is noisy, the information is lost; it is impossible that a different party p_j compensates for this loss).

LEMMA 3.1. *Conditioned on the observed transcript Π , the random variables $X_S, X_1, \dots, X_n$ 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 $X_S, X_1, \dots, X_n$.

Assume the claim holds for some transcript $\Pi = \pi$ of length $\ell - 1$, and consider the next bit Π_ℓ , sent without loss of generality by p_i , where $i \in \{S\} \cup [n]$. This bit (in case it was not changed 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 $X_{\neq i} = (X_S, X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n)$ all the variables except X_i . We have,

$$\begin{aligned} & \Pr(X_1 = x_1, \dots, X_S = x_S \mid \Pi = \pi, \Pi_\ell = b) \\ &= \frac{\Pr(X_1 = x_1, \dots, X_S = x_S, \Pi_\ell = b \mid \Pi = \pi)}{\Pr(\Pi_\ell = b \mid \Pi = \pi)} \\ &= \frac{\Pr(X_{\neq i} = x_{\neq i} \mid \Pi = \pi) \Pr(X_i = x_i, \Pi_\ell = b \mid \Pi = \pi)}{\Pr(\Pi_\ell = b \mid \Pi = \pi)} \\ &= \left(\prod_{j \neq i} \Pr(X_j = x_j \mid \Pi = \pi) \right) \frac{\Pr(X_i = x_i, \Pi_\ell = b \mid \Pi = \pi)}{\Pr(\Pi_\ell = b \mid \Pi = \pi)} \\ &= \prod_{j \neq i} \Pr(X_j = x_j \mid \Pi = \pi, \Pi_\ell = b) \\ & \quad \times \Pr(X_i = x_i \mid \Pi = \pi, \Pi_\ell = b), \end{aligned}$$

where the second equality holds by induction, since the random variables $X_i, f(X_i, \Pi)$ are independent of $X_{\neq i}$, conditioned on Π , and 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 b was changed by the channel, $b' = b \oplus E$ the claim still holds since the noise E is independent of all the other variables (i.e., we can condition on E and reduce to the case above). If the bit b was erased (in the case of a BEC) then the claim trivially holds. $\square$

As a corollary to the above, note that conditioned on any piece of information that the parties can communicate as part of their transcripts, the variables $X_S, X_1, \dots, X_n$ remain

independent. Specifically, the above holds if we condition on the correct path (up to some level), or on some levels of the inputs—we can assume a protocol in which the parties simply communicate that information (so it is a part of Π), and apply the above lemma.

COROLLARY 3.2. *The random variables $X_S, X_1, \dots, X_n$ are independent, conditioned on the observed transcript $\Pi = \pi$, (parts of) the correct path $\text{PATH} = \text{path}$, and parts of the inputs.*

4. UPPER BOUND

Showing an upper bound of $O(\log n / \log \log n)$ on the slowdown for multiparty interactive communication on star networks is rather straightforward. Essentially, all that we need to show is that every $\log n$ rounds of communication, the parties can advance $\Theta(\log \log n)$ levels in the underlying pointer jumping task.

THEOREM 4.1. *For any $\varepsilon < 1/2$, There exists a coding scheme for the pointer jumping task of depth T over a star network with $n + 1$ parties, that takes $O_\varepsilon(T \frac{\log n}{\log \log n})$ rounds and succeeds with high probability if each communication channel is a BSC_ε .*

PROOF. First, let us recall the existence of good error correction codes.

LEMMA 4.2 (SHANNON CODING THEOREM [34]). *For any discrete memoryless channel CH with capacity C and any k , there exists a code $\text{ECC} : \{0, 1\}^k \rightarrow \{0, 1\}^n$ and $\text{ECC}^{-1} : \{0, 1\}^n \rightarrow \{0, 1\}^k$ with $n = O(\frac{1}{C}k)$ such that for any $m \in \{0, 1\}^k$ it holds that,*

$$\Pr[\text{ECC}^{-1}(\text{CH}(\text{ECC}(m))) \neq m] < 2^{-\Omega(n)}.$$

The coding scheme is as follows. Assume that the parties have already correctly solved the pointer jumping task until a certain depth $\gamma \geq 0$. Each client encodes the next $\log \log n$ levels of his input (this is a subtree of size $\log n$, rooted at the current position) using a good Shannon error correcting code given by Lemma 4.2. The encoded message is of length $O(\log n)$, and we are guaranteed that the server can correctly decode the entire subtree with probability $1 - n^{-c}$, for some constant $c > 1$ to our choice. Using a union bound, the server gets all the subtrees of the clients with high probability $1 - n^{-c+1}$. Next, the server computes the correct path (of length $\log \log n$) that corresponds to each party, and sends an encoding of this path to the corresponding party. The process then repeats from the new depth $\gamma + \log \log n$. The entire scheme therefore takes $\frac{T}{\log \log n} \cdot O(\log n)$ rounds and succeeds with probability $1 - \frac{T}{\log \log n} \cdot n^{-\Omega(1)}$.

However, T may be very large with respect to n . To further improve the probability of success and prove Theorem 1.2, we use a theorem by Rajagopalan and Schulman (see [2, Section 3]).

THEOREM 4.3 ([30, 2]). *For any T round protocol over any n -party network G with maximal degree d , there exists a coding scheme Π , that takes $O(T)$ rounds and succeeds with probability $1 - n(2(d+1)p)^{\Omega(T)}$ given that any symbol transmitted in the network is correctly received with probability $1 - p$.*

In the scheme we describe above any $\log \log n$ symbols are correctly decoded with probability $1 - p$ where we can choose p

to be small enough, e.g., by taking $p = O(n^{-2})$. In this case Theorem 4.3 guarantees a coding scheme for the pointer jumping task with the same slowdown of $O(\log n / \log \log n)$ as above, which succeeds with probability $1 - n^{-\Omega(T / \log \log n)}$, that is, $1 - 2^{-\Omega(T \log n / \log \log n)}$. $\square$

5. LOWER BOUND

In this section we prove our main theorem of a lower bound of $\Omega(\frac{\log n}{\log \log n})$ on the slowdown of coding for interactive communication over star networks. Toward the lower bound, we can assume the noisy channel is actually a BEC_ε rather than a BSC_ε . This only makes the noise model weaker, and renders the lower bound stronger. In the following we assume the channel erasure probability is $\varepsilon = 1/3$. The specific value of $\varepsilon < 1$ only affects the constants involved and does not affect the validity of our result. Fixing its value will allow an easier exposition of the result.

Our main theorem is the following,

THEOREM 5.1. *There exists a constant c such that for large enough n , any protocol that solves the pointer jumping task of depth T over star networks with $n + 1$ parties in less than $c \cdot T \frac{\log n}{\log \log n}$ rounds assuming each communication channel is a $\text{BEC}_{1/3}$, has a success probability at most $1/5$.*

We begin by defining the *cutoff* of the protocol, which is an information-based measure of progress, related to the advancement in the underlying pointer jumping task.

DEFINITION 5.2 (CUTOFF). *For any transcript π , and any input $x = (x_s, x_1, \dots, x_n)$, the cutoff of the protocol $\text{cutoff}(\pi, x)$ is the minimal number k , such that both the equations below are satisfied,*

$$I(X_S[\text{path}_S(k)] \mid \Pi = \pi, \text{PATH}(k) = \text{path}(k)) \leq 2^{-0.1\sqrt{n}}, \quad (5)$$

$$\sum_{i=1}^n I(X_i[\text{path}_i(k)] \mid \Pi = \pi, \text{PATH}(k) = \text{path}(k)) \leq 0.01n. \quad (6)$$

If no such k exists we set $\text{cutoff} = T$.

The operational meaning of the cutoff is that if k is the cutoff level, then the parties know very little information on the correct paths beyond the first k edges in that path. Note that if $\text{cutoff}(\pi, x) = k$ then for any x' such that $x'^{\leq k} = x^{\leq k}$, it holds that $\text{cutoff}(\pi, x') = k$. Furthermore, the cutoff is only a function of the path up to level k , that is, if $\text{cutoff}(\pi, x) = k$ then for any input x' where $\text{path}_x(k) = \text{path}_{x'}(k)$ it holds that $\text{cutoff}(\pi, x') = k$; When the path is fixed (but we do not care about the specific input), we will usually abuse notations and write $\text{cutoff}(\pi, \text{path}(k)) = k$.

Our analysis will actually bound the cutoff in two steps. Very roughly, at the first step we will bound the information of the server given the “new” chunk of communication π^{new} , yet bound the clients’ information without it. At the second step we condition on π^{new} both for the server and the clients. For the first step described above, we define the following “server cutoff”:

DEFINITION 5.3 (SERVER CUTOFF). *Given any transcript π , and any input $x = (x_s, x_1, \dots, x_n)$, and given any continuation of the transcript π^{new} , we define the server’s cutoff level*

$\text{cutoff}_S(\pi, \pi^{\text{new}}, x)$ as the minimal number k for which

$$I(X_S[\text{path}_S(k)] \mid \Pi = \pi \circ \pi^{\text{new}}, \text{PATH}(k) = \text{path}(k)) \leq 2^{-0.2\sqrt{n}}, \quad (7)$$

and

$$\sum_{i=1}^n I(X_i[\text{path}_i(k)] \mid \Pi = \pi, \text{PATH}(k) = \text{path}(k)) \leq 0.01n. \quad (8)$$

If no such k exists we set $\text{cutoff}_S = T$.

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 $\approx T$. Hence, protocols that succeed with high probability must produce transcripts whose cutoff is large in expectation.

PROPOSITION 5.4. *Fix a protocol that solves the pointer jumping task of depth T over a star network with $n+1$ parties, that succeeds with probability at least $1/5$ on average, i.e., a protocol for which $\Pr_{X, \Pi}(\text{correct output}) \geq 1/5$. Then,*

$$\mathbb{E}_{X, \Pi}[\text{cutoff}(\Pi, X)] \geq \left(\frac{1}{5} - \frac{2}{n}\right) T.$$

PROOF. Recall that the event $\text{cutoff}(\pi, x) = k$ depends only on π and $\text{path}(k)$ and is independent of $x^{>k}$. We show that if $\text{cutoff}(\pi, \text{path}(k)) = k$ for some $k < T$, then the protocol gives the correct output with only small probability of $2/n$. This will bound the probability of the event $\text{cutoff}(\Pi, X) < T$ by $1/5 - 2/n$, and will prove that in expectation (over all inputs and possible transcripts), the cutoff is at least $T/5 - 2T/n$.

CLAIM 5.5. *Given π and $k < T$ and $\text{path}(k)$ such that $\text{cutoff}(\pi, \text{path}(k)) = k$,*

$$\Pr[\text{correct output} \mid \Pi = \pi, \text{PATH}(k) = \text{path}(k)] < \frac{2}{n}.$$

PROOF. Let L be the n -bit label of $\text{PATH}_S(k+1)$. Note that this label is included in the subtree $X_S[\text{path}(k)]$. If $\text{cutoff}(\pi, \text{path}(k)) = k$, then by the cutoff’s definition

$$I(X_S[\text{path}_S(k)] \mid \Pi = \pi, \text{PATH}(k) = \text{path}(k)) \leq 2^{-0.1\sqrt{n}},$$

and by Lemma 2.6 it holds that

$$2^{-H_\infty(L \mid \Pi = \pi, \text{PATH}(k) = \text{path}(k))} \leq \frac{1 + 2^{-0.1\sqrt{n}}}{|L|} \leq \frac{2}{n}.$$

Then, the probability that the protocol is correct is at least the probability that the clients (here treated as a single party) output the correct label L

$$\begin{aligned} & \Pr[\text{correct output} \mid \Pi = \pi, \text{PATH}(k) = \text{path}(k)] \\ & \leq 2^{-H_\infty(L \mid \Pi = \pi, \text{PATH}(k) = \text{path}(k), X_{[n]})} \\ & = 2^{-H_\infty(L \mid \Pi = \pi, \text{PATH}(k) = \text{path}(k))} \\ & \leq \frac{2}{n}. \end{aligned}$$

where the equality holds since the input of the server is independent of the input of the users conditioned on π and $\text{path}(k)$. This is implied by Lemma 3.1 (as also stated by Corollary 3.2): consider a protocol that, after completing the

pointer jumping task, communicates the correct path during its last T rounds. That is, $\text{path}(k)$ is simply part of the transcript of this protocol. Now Lemma 3.1 suggests that, because the inputs are independent when conditioned on that transcript, and because the path is simply the suffix of the transcript, then the inputs are independent conditioned on both the correct path and the prefix of the transcript (that doesn't contain the path) $\square$

The above holds for any $k < T$ and any $\pi, \text{path}(k)$ for which $\text{cutoff}(\pi, \text{path}(k)) = k$. Therefore, conditioned on the event that $\text{cutoff}(\Pi, X) < T$ the protocol outputs the correct value with probability at most $2/n$, that is,

$$\Pr_{X, \Pi}[\text{correct output} \mid \text{cutoff}(\Pi, X) < T] \leq 2/n.$$

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,

$$\begin{aligned} \frac{1}{5} &\leq \Pr_{X, \Pi}[\text{correct output}] \\ &= \Pr[\text{cutoff}(\Pi, X) < T] \Pr[\text{correct output} \mid \text{cutoff}(\Pi, X) < T] \\ &\quad + \Pr[\text{cutoff}(\Pi, X) = T] \Pr[\text{correct output} \mid \text{cutoff}(\Pi, X) = T] \\ &\leq \Pr[\text{cutoff}(\Pi, X) < T] \cdot 2/n + \Pr[\text{cutoff}(\Pi, X) = T] \cdot 1, \end{aligned}$$

ergo,

$$\Pr[\text{cutoff}(\Pi, X) = T] \geq \frac{1}{5} - \frac{2}{n}$$

and

$$\mathbb{E}_{X, \Pi}[\text{cutoff}(\Pi, X)] \geq T \left(\frac{1}{5} - \frac{2}{n} \right),$$

as claimed.

In order to prove the main theorem we show that during every $0.1 \log n$ rounds of communication, the cutoff level increases by at most $O(\log \log n)$, in expectation. Formally,

THEOREM 5.6. *Given a protocol for the pointer jumping task, let π be the transcript of the protocol observed up to some round, and let Π^{new} be a random variable describing the observed transcript over the next $0.1 \log n$ rounds. Then, for any $\ell \leq T$, and for any $x^{\leq \ell}$ it holds that*

$$\begin{aligned} \mathbb{E} \left[\text{cutoff}(\pi \circ \Pi^{\text{new}}, X) \mid \begin{array}{l} \Pi = \pi, \text{PATH}(\ell) = \text{path}(\ell), \\ \text{cutoff}(\pi, X) = \ell \end{array} \right] \\ \leq \ell + O(\log \log n). \end{aligned}$$

Note that the expectation is over the inputs, the noise, and the protocol's randomness.

With the above propositions, the proof of Theorem 5.1 is immediate: if the protocol can output the correct answer with probability at least $1/5$, it must be that the expected cutoff level at the end of the protocol is $> T/5 - o(T)$, but this would take $O(T \frac{\log n}{\log \log n})$ rounds of communication, in expectation. Formally,

PROOF. (Theorem 5.1) Using Theorem 5.6, for any protocol for the pointer jumping task there exists a (small enough) constant $c > 0$ such that after running $cT \frac{\log n}{\log \log n}$ rounds of the protocol, the expected cutoff for the observed transcript is small, $\mathbb{E}_{X, \Pi}[\text{cutoff}(\Pi, X)] < T/10$. Therefore, it cannot

be that the protocol correctly solves the T -depth pointer jumping with probability above $1/5$ as this will contradict Proposition 5.4. $\square$

We now turn to prove the key technical Theorem 5.6. Intuitively speaking, the main idea is the following. We cut the protocol into chunks of length $0.1 \log n$ rounds and treat each one separately, showing that the cutoff level cannot increase during that chunk by more than $O(\log \log n)$. We can assume that at the beginning of each chunks all the parties are given the information about the correct path up to the depth matching the current cutoff level, and reduce this case (in some sense⁴) to a new instance of the pointer jumping task starting at that depth.

During the $0.1 \log n$ rounds of the next chunk, with probability at least $1 - 2^{-\sqrt{n}}$, there exists a subset Q of $\sqrt{n}$ parties about which the server does not have much information (beyond the cutoff point) whose communication was *completely erased* by the channel throughout this chunk. We can assume that other than this set of parties Q , the communication is noiseless. In this case, it is quite intuitive that the cutoff levels cannot increase by too much: the server is missing any relevant information about the inputs of parties in Q beyond the cutoff level, thus the information that it sends during that chunk is practically meaningless, and the server's cutoff level remains more or less the same. Additionally, since the server did not communicate a lot of meaningful information about his input, the clients do not know how to proceed and cannot send too much relevant information; thus, their cutoff level does not increase too much as well. On the other hand, in the rare case where no subset Q exists (i.e., the communication in this chunk is practically noiseless), the cutoff may tremendously increase, however since this event is so rare it will add only $O(1)$ to the accumulated cutoff level throughout the entire protocol, in expectation.

PROOF. (Theorem 5.6) We begin by showing that with high probability, there exists a subset of size $\sqrt{n}$ of the clients, for which the server knows very little information beyond the cutoff level, and yet in the next $0.1n \log n$ rounds their communication was completely erased by the channel.

DEFINITION 5.7. *Given a transcript π and an input x so that $\text{cutoff}(\pi, x) = k$. For $i \in [n]$, we say that a client p_i is critical if*

$$I(X_i[\text{PATH}_i(k)] \mid \Pi = \pi, \text{PATH}(k) = \text{path}(k)) \leq 0.02.$$

LEMMA 5.8. *Let π be the transcript so far and consider the next $0.1 \log n$ rounds of communication. Denote by E_{silence} the event that there exists a subset Q of parties of size at least $\sqrt{n}$, such that all the parties in Q are critical and all the bits sent by parties in Q were erased by the channel. Then,*

$$\Pr[E_{\text{silence}}] > 1 - 2^{-\sqrt{n}}.$$

PROOF. There are at least $n/2$ critical parties, or otherwise,

$$\begin{aligned} \sum_i I(X_i[\text{PATH}_i(k)] \mid \Pi = \pi, \text{PATH}(k) = \text{path}(k)) \\ \geq \frac{n}{2} \cdot 0.02 \geq 0.01n, \end{aligned}$$

⁴The main difference is that previous communication may have leaked some information on this new instance, and we need to account for this information as well.

and k cannot be the cutoff round, by Definition 5.2. Moreover, note that the probability that all the $0.1 \log n$ transmissions of a specific party p_i are erased (or even the $0.2 \log n$ bits sent and received by this party), is $\frac{1}{3}^{0.1 \log n} \geq n^{-0.4}$. Let Q be the set of all critical parties whose entire communication was deleted, using Chernoff bound and assuming large enough n ,

$$\Pr[|Q| < \sqrt{n}] < \exp\left(-\frac{n^{0.6}}{4}\right).$$

Here we use the fact that $\varepsilon = 1/3$, however it is clear that for any other constant ε we can reduce the length of a chunk to be $c \log n$ such that, say, $\varepsilon^{c \log n} \geq n^{-0.4}$ and all the other proofs below remain valid, maybe up to adjusting the constants as needed. $\square$

For any $\ell \leq T$, any fixing $\text{path}(\ell)$ and any transcript π denote by $E_{(\pi, \text{path}(\ell), \ell)}$ the event

$$(\Pi = \pi, \text{PATH}(\ell) = \text{path}(\ell), \text{cutoff}(\pi, X) = \ell).$$

Recall that whether the cutoff is ℓ depends only on π and the first ℓ levels the correct path, therefore $E_{(\pi, \text{path}(\ell), \ell)}$ is either empty or equal to $(\Pi = \pi, \text{PATH}(\ell) = \text{path}(\ell))$. For any continuation π_S^{new} of bits sent by the server in the new chunk define $E_{(\pi, \pi_S^{\text{new}}, \text{path}(\ell), \ell)}$ the event $(\Pi = \pi, \Pi_S^{\text{new}} = \pi_S^{\text{new}}, \text{PATH}(\ell) = \text{path}(\ell), \text{cutoff}_S(\pi, \Pi_S^{\text{new}}, X) = \ell)$.

The proof of the theorem will follow from the next three propositions:

PROPOSITION 5.9. *For any $\ell \leq T$, any $\text{path}(\ell)$ and any transcript π ,*

$$\mathbb{E}[\text{cutoff}_S(\pi, \Pi^{\text{new}}, X) \mid E_{(\pi, \text{path}(\ell), \ell), E_{\text{silence}}}] \leq \ell + 61.$$

PROPOSITION 5.10. *Split the observed transcript of the new chunk $\Pi^{\text{new}} = (\Pi_S^{\text{new}}, \Pi_{[n]}^{\text{new}})$ to the parts corresponding to information sent by the server and by the clients, respectively. For any $\ell' \leq \ell \leq T$, any fixing $\text{path}(\ell)$, any transcript π , and any (server's) new transcript π_S^{new} ,*

$$\mathbb{E} \left[\text{cutoff}(\pi \circ \Pi^{\text{new}}, X) \mid \begin{array}{c} E_{(\pi, \text{path}(\ell'), \ell')} \\ E_{(\pi, \pi_S^{\text{new}}, \text{path}(\ell), \ell)} \\ E_{\text{silence}} \end{array} \right] \leq \ell + 5 \log \log n.$$

PROPOSITION 5.11. *For any $\ell \leq T$, any fixing $\text{path}(\ell)$ and any transcript π ,*

$$\mathbb{E} \left[\text{cutoff}(\pi \circ \Pi^{\text{new}}, X) \mid \frac{E_{(\pi, \text{path}(\ell), \ell)}}{E_{\text{silence}}} \right] \leq \ell + O(\log n \log \log n).$$

The above three propositions prove the theorem: When the good event E_{silence} doesn't happen, the cutoff increases by at most $O(\log n \log \log n)$ (Proposition 5.11), but this happens with probability at most $\Pr[E_{\text{silence}}] < 2^{-\sqrt{n}}$ (by Lemma 5.8), thus the expected contribution to the increase of the cutoff by such chunks is bounded by a negligible amount of $O(\log n \log \log n) \cdot 2^{-\sqrt{n}}$. Otherwise, assuming the previous cutoff was ℓ , then with the information of Π^{new} the server's cutoff level according to Proposition 5.9, is in expectation at most $\ell^S \leq \ell + 61$. Finally, given that the server's cutoff is ℓ^S , Proposition 5.10 guarantees that the new cutoff (i.e., when considering Π^{new} for both the server and the clients), is in expectation at most $\ell^S + 5 \log \log n = \ell + O(\log \log n)$.

In the full version of this manuscript [6] we prove the above three propositions.

6. REFERENCES

- [1] S. Agrawal, R. Gelles, and A. Sahai. Adaptive protocols for interactive communication. *Computing Research Repository (CoRR)*, arXiv:1312.4182 (cs.DS), 2013.
- [2] N. Alon, M. Braverman, K. Efremenko, R. Gelles, and B. Haeupler. Reliable communication over highly connected noisy networks. *Electronic Colloquium on Computational Complexity (ECCC)*, TR15-014, 2015.
- [3] Z. Brakerski, Y. T. Kalai, and M. Naor. Fast interactive coding against adversarial noise. *J. ACM*, 61(6):35:1–35:30, Dec 2014.
- [4] M. Braverman. Towards deterministic tree code constructions. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 161–167, 2012.
- [5] M. Braverman and K. Efremenko. List and unique coding for interactive communication in the presence of adversarial noise. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, FOCS '14, pages 236–245, 2014.
- [6] M. Braverman, K. Efremenko, R. Gelles, and B. Haeupler. Constant-rate coding for multiparty interactive communication is impossible. *Electronic Colloquium on Computational Complexity (ECCC)*, TR15-197, 2015.
- [7] M. Braverman and A. Rao. Toward coding for maximum errors in interactive communication. *Information Theory, IEEE Transactions on*, 60(11):7248–7255, Nov 2014.
- [8] A. Chattopadhyay, J. Radhakrishnan, and A. Rudra. Topology matters in communication. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 631–640, Oct 2014.
- [9] K.-M. Chung, R. Pass, and S. Telang. Knowledge-preserving interactive coding. In *Proceedings of the 54th annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 449–458, 2013.
- [10] K. Efremenko, R. Gelles, and B. Haeupler. Maximal noise in interactive communication over erasure channels and channels with feedback. In *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science*, ITCS '15, pages 11–20, 2015.
- [11] 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, Jan 2015.
- [12] R. Gallager. Finding parity in a simple broadcast network. *Information Theory, IEEE Transactions on*, 34(2):176–180, Mar 1988.
- [13] R. Gelles. Coding for interactive communication: A survey, 2015.
- [14] R. Gelles and B. Haeupler. Capacity of interactive communication over erasure channels and channels with feedback. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 1296–1311, 2015.
- [15] R. Gelles, B. Haeupler, G. Kol, N. Ron-Zewi, and A. Wigderson. Towards optimal deterministic coding for interactive communication. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1922–1936, 2016.

- [16] R. Gelles, A. Moitra, and A. Sahai. Efficient and explicit coding for interactive communication. In *Proceeding of the IEEE Symposium on Foundations of Computer Science*, FOCS '11, pages 768–777, 2011.
- [17] R. Gelles, A. Moitra, and A. Sahai. Efficient coding for interactive communication. *Information Theory, IEEE Transactions on*, 60(3):1899–1913, March 2014.
- [18] R. Gelles, A. Sahai, and A. Wadia. Private interactive communication across an adversarial channel. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS '14, pages 135–144, 2014.
- [19] R. Gelles, A. Sahai, and A. Wadia. Private interactive communication across an adversarial channel. *Information Theory, IEEE Transactions on*, 61(12):6860–6875, Dec 2015.
- [20] M. Ghaffari and B. Haeupler. Optimal Error Rates for Interactive Coding II: Efficiency and List Decoding. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, FOCS '14, pages 394–403, 2014.
- [21] M. Ghaffari, B. Haeupler, and M. Sudan. Optimal error rates for interactive coding I: Adaptivity and other settings. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 794–803, 2014.
- [22] N. Goyal, G. Kindler, and M. Saks. Lower bounds for the noisy broadcast problem. *SIAM Journal on Computing*, 37(6):1806–1841, 2008.
- [23] B. Haeupler. Interactive Channel Capacity Revisited. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, FOCS '14, pages 226–235, 2014.
- [24] W. M. Hoza and L. J. Schulman. The adversarial noise threshold for distributed protocols. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 240–258, 2016.
- [25] A. Jain, Y. T. Kalai, and A. Lewko. Interactive coding for multiparty protocols. In *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science*, ITCS '15, pages 1–10, 2015.
- [26] G. Kol and R. Raz. Interactive channel capacity. In *STOC '13: Proceedings of the 45th annual ACM Symposium on theory of computing*, pages 715–724, 2013.
- [27] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):pp. 79–86, Mar 1951.
- [28] A. Lewko and E. Vitercik. Balancing communication for multi-party interactive coding, 2015. arXiv preprint arXiv:1503.06381.
- [29] L. Lima, D. Ferreira, and J. Barros. Topology matters in network coding. *Telecommunication Systems*, 51(4):247–257, 2012.
- [30] S. Rajagopalan and L. Schulman. A coding theorem for distributed computation. In *STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 790–799, 1994.
- [31] L. J. Schulman. Communication on noisy channels: a coding theorem for computation. *Foundations of Computer Science, Annual IEEE Symposium on*, pages 724–733, 1992.
- [32] L. J. Schulman. Deterministic coding for interactive communication. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 747–756, 1993.
- [33] L. J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996.
- [34] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001. Originally appeared in *Bell System Tech. J.* 27:379–423, 623–656, 1948.