Reliable Communication over Highly Connected Noisy Networks

Noga Alon[*], Mark Braverman[†], Klim Efremenko[‡], Ran Gelles[§], and Bernhard Haeupler[¶]

Abstract

We consider the task of multiparty computation performed over networks in the presence of random noise. Given an n-party protocol that takes R rounds assuming noiseless communication, the goal is to find a coding scheme that takes R' rounds and computes the same function with high probability even when the communication is noisy, while maintaining a constant asymptotic *rate*, i.e., while keeping $\liminf_{n,R\to\infty} R/R'$ positive.

Rajagopalan and Schulman (STOC '94) were the first to consider this question, and provided a coding scheme with rate $O(1/\log(d+1))$, where d is the maximal degree in the network. While that scheme provides a constant rate coding for many practical situations, in the worst case, e.g., when the network is a complete graph, the rate is $O(1/\log n)$, which tends to 0 as n tends to infinity.

We revisit this question and provide an efficient coding scheme with a constant rate for the interesting case of fully connected networks. We furthermore extend the result and show that if a (d-regular) network has mixing time m, then there exists an efficient coding scheme with rate $O(1/m^3 \log m)$. This implies a constant rate coding scheme for any n-party protocol over a d-regular network with a constant mixing time, and in particular for random graphs with n vertices and degrees $n^{\Omega(1)}$.

[*]Sackler School of Mathematics and Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel and School of Mathematics, Institute for Advanced Study, Princeton, NJ 08540. Email: nogaa@tau.ac.il. Research supported in part by a USA-Israeli BSF grant, by an ISF grant, by the Israeli I-Core program and by the Oswald Veblen Fund.

[†]Princeton University, mbraverm@cs.princeton.edu. Research supported in part by an NSF CAREER award (CCF-1149888), a Turing Centenary Fellowship, a Packard Fellowship in Science and Engineering, and the Simons Collaboration on Algorithms and Geometry.

[‡]UC Berkeley, klimefrem@gmail.com

[§]Princeton University, rgelles@cs.princeton.edu

[¶]Carnegie Mellon University, haeupler@cs.cmu.edu

1 Introduction

The field of coding for interactive communication, initiated by Schulman in the early 90's [Sch92, Sch93, Sch96], aims at performing arbitrary distributed computations when the communication channels between the different nodes suffer from noise. For the case where two parties communicate over a discrete memoryless channel (say, the binary symmetric channel that flips every bit with an independent probability of ε, denoted BSC_ε), the scheme of [Sch96] provides an efficient coding with good parameters: the coding of an R-round (noiseless) protocol takes only $O(R)$ rounds, and gives the correct outputs with probability $1 - 2^{-\Omega(R)}$ over the noisy network.

In 1994, Rajagopalan and Schulman [RS94] extended the result to the multiparty case. Here we are given a network with n nodes of some arbitrary topology, where each communication link is an independent BSC_ε, where at every step of the protocol each party sends one bit through each one of the channels it is connected to (possibly sending different bits to different parties). In this case, any protocol of R rounds over the noiseless network can be coded into a resilient protocol that takes $O(R \log(d+1) + \log n)$ rounds and succeeds with probability $1 - 2^{-\Omega(R)}$. The parameter d is the maximum degree of any node in the network, i.e., the maximal number of links connected to a single party. Although the scheme of [RS94] is not efficient, Gelles, Moitra and Sahai [GMS11, GMS14] showed that it can be extended to a fully efficient (randomized) scheme. The coding scheme of [RS94] has good parameters for any constant number of parties, however it may not work as well when the number of parties is large. Indeed, in highly connected networks, and in particular when the topology is a complete graph on n vertices, the redundancy added by the coding becomes $\Theta(\log n)$. In other words, the *rate*, the length of the noiseless protocol divided by the length of the encoded protocol, is vanishing, being $O(1/\log n)$.

We revisit the question of coding for multiparty interactive communication, and ask whether it is possible to find efficient coding schemes with rate $O(1)$ for the case where the network is highly connected, for example, when the topology is a complete graph.

We answer the above in the affirmative.

Theorem 1.1 (coding over complete graphs)**.** *For any n-party protocol π that takes R-rounds over the fully-connected (noiseless) network, and for any constant $\varepsilon < 1/2$, there exists a resilient protocol Π that simulates π over a fully connected network in which every link is a BSC_ε. The simulation is computationally efficient, takes $O_\varepsilon(R)$ rounds and succeeds with probability $1 - 2^{-\Omega(\sqrt{n}R)}$.*

This result sheds some light on the differences between the case where each pair of parties share a separate (noisy) channel, and the case where all the parties share a joint (noisy) broadcast channel. It was previously shown that if the users share a noisy *broadcast channel* then certain tasks, such as computing the parity of all the inputs, or learning the input bit of all the parties can be done in $O(n \log\log n)$ noisy-broadcast rounds [Gal88], while assuming a noiseless broadcast channel, these tasks trivially take n rounds. Furthermore, these tasks cannot be done with fewer rounds when the channel is noisy, i.e., the $O(\log\log n)$ blowup is tight [GKS08]. On the other hand, as implied by Theorem 1.1, such a blowup no longer holds in a setting where any two connected parties use a separate noisy channel.

In order to prove Theorem 1.1, we show a coding that simulates a single round of the noiseless protocol. Consider the *neighborhood connectivity* task in which each party holds one bit designated to each one of its neighbors. It is easy to verify that sending each bit directly requires $\Omega(\log n)$ rounds in order to be decoded correctly with high probability. However, we show a coding protocol that solves the neighborhood-connectivity task over a noisy network with high probability in $O(1)$ rounds. Instead of transmitting each bit directly, we relay each bit through large portions of the network using an appropriate (Shannon) error correcting code [Sha48]. To illustrate this simple idea, consider a very simplified case in which some source s wishes to send a single bit to a target node t, over the noisy

network. In order to complete this task in $O(1)$ rounds, s can relay its bit to all its $n-1$ neighbors, and then they will send their (noisy) copy to t. Thus, in two rounds the target node t receives $n-1$ independent estimations of the bit, where each estimation is correct with probability $(1-\varepsilon)^2+\varepsilon^2$; this allows t to correctly decode the bit with high probability by taking the majority of the estimations.

With the above relaying technique in mind, we can design a coding scheme for the neighborhood connectivity in $O(1)$ rounds. Here is an outline. We describe the neighborhood connectivity task as an $n \times n$ matrix where each row and column describes a specific party and the (i,j) entry is the amount of bits the i-th party wants to communicate to the j-th party (in particular, the matrix is all-ones in the case of a complete graph). We first divide the n parties into subsets of size $\sqrt{n}$. This division can be seen as splitting the above matrix into n blocks of size $\sqrt{n}$ by $\sqrt{n}$. Next, we associate each such block with a specific party, or more accurately, we associate this party with the n-bits of information defined by that block in the matrix. The coding will work in two symmetric parts. First, each party (each row in the matrix) will encode each of the $\sqrt{n}$ bits that belong to a specific block and send it to the party associated with it. After this step, each party knows all the n bits of the block associated to it. Next, each associated party of a specific block will encode the $\sqrt{n}$ bits designated to a specific party (that is, the bits that lie in a column of the matrix), and send them to that party. Since both parts encode the bits prior to sending them, they will be decoded correctly with high probability.

Note that using the above relaying technique, it is possible to send each of the encodings (of length $O(\sqrt{n})$) in a constant number of round. In fact, we can parallelize their transmissions and communicate all the necessary information in a constant number of rounds.

Coding over highly-connected topologies. In addition to the complete graph topology, we also consider highly-connected networks whose topology is a d-regular graph with a small mixing time m (see Definition 5.1). We show a coding scheme with rate $O(1/(m^3 \log m))$ that succeeds with high probability.

Theorem 1.2 (coding over d-regular graphs)**.** *Assume a network topology G of a d-regular graph with mixing time m, and assume $d > \log^{1+\Omega(1)} n$. For any n-party protocol π that takes R rounds over the noiseless network and for any $\varepsilon < 1/2$, there exists a resilient protocol Π that simulates π over the network G where every link is a BSC_ε. The simulation is computationally efficient, takes $O_\varepsilon(R \cdot m^3 \log m)$ rounds and succeeds with probability $1 - 2^{-\Omega(d^{\Omega(1)} \cdot R)} > 1 - n^{-\omega(1)R}$.*

For the case of $m = O(1)$, e.g., for random graphs with $d = n^\alpha$ for some constant $\alpha > 0$, Theorem 1.2 implies a coding scheme with a constant rate $\Theta(1)$, and a success probability of $1 - 2^{-n^{\Omega(1)}R}$. Note that the rate obtained by the coding scheme of [RS94] for such networks is $O(1/\log n^\alpha) = o(1)$.

Similar to the case of complete graphs, it suffices to solve the neighborhood-connectivity task in $O(1)$ rounds in order to obtain a constant rate scheme over a d-regular graph, assuming a constant mixing time $m = O(1)$. However, the challenge here is bigger than in the complete graph case, as every node is connected to a relatively small number of nodes, and it is not clear how to relay bits using arbitrary portions of the network. Nevertheless, we show a way for large subsets of nodes to talk with each other simultaneously without disturbing each other, in a reliable way. Specifically, assume we have a list of (distinct) source-target pairs where each source node aims to send a total amount of $O(d)$ bits, and so that each node appears at most $O(d/\Lambda)$ times in the list, for some parameter Λ. First we encode each chunk of information using a standard error correcting code that has failure probability $2^{-\Omega(\Lambda)}$ over a BSC_ε, and define a new list where each source-target pair appears with multiplicity which equals the number of bits to be transferred from that source to the target after the encoding. Note that in the transformed list, each node may appear up to $O(d)$ times. Next we show that it is possible to choose a set of short paths, such that for every element in the list there is a unique path connecting the source with the target (i.e., with parallel paths between multiple occurrences of the same source and target), and yet these paths are *jointly edge-disjoint*. This implies that we can

deliver all the codewords to their destinations in $O(1)$ rounds, and successfully decode each codeword with high probability. Choosing a set of edge-disjoint paths applies a variant of the methods used in the papers about finding edge-disjoint paths in expander graphs and in particular [BFU94], see also [AC07] and the references therein. However, in our case we need all the paths to be of constant length $O(m)$, and we do not restrict the list of source-target pairs to be disjoint. Our approach applies the Lovász Local Lemma, where the basic combinatorial statement applied is the fact, first proved in [Alo88], that given any family of pairwise disjoint sets of vertices in a graph, each of size somewhat larger than the maximum degree in the graph, there is always an independent set containing a vertex from each of these sets. Moreover, such an independent set can be found efficiently.

The above reliable coding is not enough to complete the proof: since we are restricted to source-target lists where each node appears at most $O(d/\Lambda)$ times, we cannot apply this method directly to the neighborhood-connectivity task. Indeed, each party begins with one bit to send to each of its d neighbors, and the $O(d/\Lambda)$ restriction cannot hold.

Still, using the above coding we show how to perform a sequence of relays where each causes the communication to be more "local". In other words, each relay splits the network into disjoint subsets where the communication is guaranteed to occur only between parties of the same subset. Each such a relay reduces the subset size by a factor of $\frac{\Lambda}{d}$ and increases the communication by only a constant factor. Since a mixing time m implies $d > n^{1/m}$, after $O(m) = O(1)$ relays (with the right choice of Λ), each subset is of size at most $O(d/\Lambda)$ while each party needs to communicate at most $O(d)$ bits. Now the communication demand satisfies the conditions of the coding scheme (i.e., every node appears at most $O(d/\Lambda)$ times in the induced source-target list) and we can employ the coding one last time to complete the task.

Related Work. As mentioned above, the task of coding for interactive communication in the two party case, assuming random noise, was first considered by Schulman [Sch92, Sch93, Sch96]. These constructions either have non-constant rate, or they utilize a data structure named *tree-code* for which no efficient construction is known. Later, Gelles, Moitra and Sahai [GMS11, GMS14] provided a randomized relaxation for the tree code which can be constructed efficiently, thus solving the task efficiently when the noise model is random.

Another interesting model for the two-party interactive communication task is when the noise is not random but rather adversarial, where the only limit is on the total fraction of bit flips allowed. The maximal noise that can be tolerated, and efficient schemes that tolerate a constant fraction of noise (up to the possible limit) were considered in [BR11, BR14, BK12, BN13, GH14, BE14]. A long sequence of works consider two-party interactive communication for various models and assumptions, e.g. when the parties are allowed to share a (private) random string [FGOS13, FGOS15], the case of adaptive protocols [AGS13, GHS14], the case of erasure channels and channels with feedback [FGOS15, GH15, EGH15], and the case of private computations [CPT13, GSW14]. The capacity (the maximal rate) of interactive protocols in the two-party case was considered by [KR13, Pan13, Hae14, GH15].

Despite a large body of work on the two-party case, not too much is known for the multiparty case, beyond the aforementioned result of Rajagopalan and Schulman [RS94], and its efficient extension [GMS11, GMS14]. Subsequent to the appearance of a preliminary version of this work, it was shown that a rate of $O(1)$ *cannot* be achieved for certain topologies, e.g., a star [BEGH15].

For the adversarial noise model, Jain, Kalai and Lewko [JKL15] consider the case of multiparty interaction and show a tight $\Theta(1/n)$ bound on the fraction of noise, for star topology networks in the asynchronous model. Hoza and Schulman [HS14] show a coding scheme that applies to any network topology in the synchronous model, and resists a maximal noise level of $O(1/n)$ with rate $O(n/m \log n)$, where m is the number of edges in the network. In addition, they provide a coding scheme, along with tight bounds on the permissible level of noise, for the interesting case where the noise level *per edge* is bounded.

2 Model Definition and Preliminaries

Notations. Let us fix some notations used throughout. Let $\mathbb{N} = \{0, 1, 2, \ldots\}$ denote the natural numbers (including zero), and for any integer $n \geq 1$, put $[n] = \{1, 2, 3, \ldots, n\}$. We say that a function $f(n)$ is *negligible* in n, if $f(n) \leq n^{-\omega(1)}$; We will usually want our constructions to fail with at most negligible probability in the number of parties. All logarithms throughout the paper are in base 2.

Noisy Networks and Protocols. 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.

Definition 2.1. *A binary symmetric channel with error probability ε, is a binary channel $\mathsf{BSC}_\varepsilon : \{0,1\} \to \{0,1\}$ such that for any $b \in \{0,1\}$, it holds that $\Pr[\mathsf{BSC}_\varepsilon(b) \neq b] = \varepsilon$ independently for each instantiation of the channel.*

A single *round* of communication in the network means the simultaneous transmission of $2|E|$ bits: for any $(u, v) \in E$, u sends a bit to v and receives a bit from v. A protocol of length m that computes $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ is a distributed algorithm where each p_i begins the protocol with an input x_i, and after m rounds of communication each p_i outputs y_i.

In order to ease notations we usually assume G contains self loops and that a party can "send" a bit to itself. Concretely for a complete graph, each node has n neighbors rather than $n - 1$.

Finally, we will be using a standard error correction code, implied by the work of Shannon. Formally,

Lemma 2.2 (Shannon Coding Theorem [Sha48])**.** *For any discrete memoryless channel T with capacity C and any k, there exists a code $\mathsf{ECC} : \{0,1\}^k \to \{0,1\}^n$ and $\mathsf{ECC}^{-1} : \{0,1\}^n \to \{0,1\}^k$ with $n = O(\frac{1}{C}k)$ such that for any $m \in \{0,1\}^k$ it holds that,*

$$\Pr\left[\mathsf{ECC}^{-1}(T(\mathsf{ECC}(m))) \neq m\right] < 2^{-\Omega(n)}.$$

For a BSC_ε channel, the capacity C is given by $1 - H(\varepsilon) = 1 + \varepsilon \log \varepsilon + (1 - \varepsilon) \log(1 - \varepsilon)$. We note that one can efficiently construct codes with the above parameters (and efficiently encode and decode them), see, e.g., [Spi95, GI05].

3 The Neighborhood Connectivity Task and Interactive Protocols

A trivial observation is that any R-round multiparty interactive protocol can be split into R basic steps, in each of which every party has a single bit to send to any of its neighbors. We define this task as the *neighborhood connectivity* task.

Formally, for any network with n nodes, the neighborhood connectivity task is defined by

Definition 3.1. *Let $\vec{a}_i = (a_{i,1}, \ldots, a_{i,d_i})$ where for any $i \in [n], j \in [d_i]$ the bit $a_{i,j} \in \{0,1\}$ is to be interpreted as the input of party i which is designated to its j-th neighbor. The $\mathsf{Neighbor}$ function is defined as:*

$$\mathsf{Neighbor}(\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n) = (\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n).$$

where $\vec{b}_i$ consists of the d_i bits designated to party i by its d_i neighbors.

We say that some protocol computes the $\mathsf{Neighbor}$ task over a BSC_ε, if it fails with a negligible probability in n. Note that a successful computation of the $\mathsf{Neighbor}$ task implies that *all* parties have received the correct output; this is different from the standard successful transmission which usually accounts for only a single party, or a single bit transmission.

Composing R rounds of neighborhood connectivity immediately gives a coding scheme for any R-round protocol, yet with a success probability that decreases with R.

Claim 3.2. *For any network G and any R-round multiparty protocol π, if the neighborhood connectivity task over G can be performed in k rounds with probability $1 - n^{-\omega(1)}$, then there exists a coding Π that simulates π with $O(kR)$ rounds and succeeds with probability $1 - R \cdot n^{-\omega(1)}$.*

However in general the number of rounds R can be very large. We claim that even in this case, obtaining a coding protocol that succeeds with high probability is possible. To this end we use a result by Rajagopalan and Schulman [RS94] who showed an efficient coding scheme that succeeds to simulate any protocol over a memoryless noisy channel as long as the probability of correctly decoding a single transmission is at least $1 - (d+1)^{-\Omega(1)}$, where d is the maximal degree of a node in the network.

Theorem 3.3 ([RS94])**.** *For any R round protocol π over any network G, there exists a coding scheme Π, that takes $O(R)$ rounds and succeeds with probability $1 - n(2(d+1)p)^{\Omega(R)}$ given that any symbol transmitted in the network is correctly received with probability $1 - p$ where d is the maximal degree of nodes in G.*

Sketch of proof. The theorem is an immediate consequence of the analysis of [RS94]. Although not written explicitly there, it easily follows from the analysis of Lemmas 5.1.1 and 5.1.2 in [RS94]; see also the detailed analysis in [Raj94, Section 3]. We omit the details here. □

To bring the decoding probability of a single transmission to the required level of $p < (2(d+1))^{-\Omega(1)}$, Rajagopalan and Schulman simply use a Shannon code of length $O(\log(d+1))$, thus obtaining a similar overhead (see Lemma 5.1.2 in [RS94]). However, if we replace the Shannon code used in [RS94] with $O(1)$ rounds[1] of the neighborhood connectivity task, we effectively reduce the probability of a failed transmission to $p < n^{-\omega(1)} \ll (d+1)^{-\Omega(1)}$. This immediately implies the following.

Corollary 3.4. *For any network G and any R-round multiparty protocol π, if the neighborhood connectivity task over G can be performed in k rounds with probability $1 - p$ where $p = n^{-\omega(1)}$, then there exist a coding Π that simulates π with $O(kR)$ rounds and succeeds with probability $1 - p^{\Omega(R)}$.*

4 Resilient Communication over Complete Graphs

In this section we show a coding scheme for the neighborhood connectivity task that takes $O(1)$ rounds assuming the network has an underlying topology of a complete graph. Consider n parties where each two are connected via a BSC_ε for some constant $\varepsilon < 1/2$. The neighborhood connectivity task in the case of complete-graph networks can be described by giving each p_i the n input bits $\vec{a}_i \equiv (a_{i,1}, a_{i,2}, \ldots, a_{i,n})$ where the i-th bit should be sent to the i-th party. Then,

$$\mathsf{Neighbor}(\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_n) = (\vec{a}_1^{\intercal}, \vec{a}_2^{\intercal}, \ldots, \vec{a}_n^{\intercal}),$$

where $\vec{a}_j^{\intercal} \equiv (a_{1,j}, a_{2,j}, \ldots, a_{n,j})$.

Our main theorem in this section is a coding scheme that solves the Neighbor task in $O(1)$ rounds over complete graph noisy networks.

Theorem 4.1. *For any constant $\varepsilon < 1/2$, the Neighbor task for n-parties can be efficiently computed with probability $1 - 2^{-\Omega(\sqrt{n})}$ in $O(1)$ communication rounds over a fully-connected network, where each channel is a BSC_ε.*

[1] Each transmission in [RS94] is of a symbol taken from a small finite alphabet. Hence, each such symbol can be communicated with $O(1)$ rounds of (bit) neighborhood connectivity.

The above theorem along with Corollary 3.4 give a constant rate coding scheme for any multiparty protocol over complete graphs, establishing Theorem 1.1.

Before we prove the theorem, let us begin by showing that any specific party can reliably transfer a large amount of information to a single party, using $O(1)$ rounds of communication. Assume p_1 wishes to send n bits to p_2. Obviously, p_1 can send the information to p_2 bit by bit, yet this would take them n rounds. A different approach would be to relay the bits using the entire network. That is, on the first round, p_1 sends one bit of its input to all the other parties (e.g., the i-th bit is sent to the i-th party). On the second round, all the parties relay the bit they have received at the first round back to p_2. This way, p_2 gets all the n bits, where each bit is flipped with probability $2\varepsilon(1-\varepsilon)$. In order to send the information reliably, p_1 can use a code ECC using Lemma 2.2. Such an encoding increases the amount of information to be communicated to $c \cdot n$ bits for some constant $c > 1$, and thus requires repeating the above process $c = O(1)$ times. We name the above approach *two-steps transfer*.

An interesting observation is that during each round, we utilize only n links of the entire networks (that has n^2 links). This means that we can perform the above two-steps transfer n times *in parallel* as long as no two instantiations have the same sender or the same receiver (this is a special case of the routing scheme by [DLP12, Len13]). Using this observation, we can complete the proof of Theorem 4.1. See Appendix A for full details.

5 Resilient Communication over Highly Connected Graphs

In this section we generalize the above result, to the case where the underlying network is a d-regular graph, rather than a complete graph. The overhead of the coding scheme in this case depends on the mixing time of the network graph.

Definition 5.1. *Let $\hat{A}$ be the normalized adjacency matrix of a graph $G = (V, E)$ with n nodes. Let $\vec{u} = (1/n, 1/n, \ldots, 1/n)$. The graph G has* mixing time m *if, for any probability vector $\vec{p}$,*

$$\left\| \hat{A}^m \vec{p} - \vec{u} \right\|_\infty \leq \frac{1}{2n}.$$

We note that if a d-regular graph has a mixing time m, then $d > n^{\frac{1}{m}}$. Also we note that a random graph with $d \geq \Omega((n \log n)^{\frac{1}{m}})$ has mixing time at most m.

As before, we aim to solve the neighborhood connectivity task. That is, each party now begins with d bits designated to his neighbors, and the goal is to reliably transfer all these $n \cdot d$ bits to their destinations, in a number of rounds that only depends on m.

Theorem 5.2. *For any constant $\varepsilon < 1/2$, the $\mathsf{Neighbor}$ task can be efficiently computed with probability $1 - mn^2 2^{-d^{\Omega(1)}}$ in $O(m^3 \log m)$ communication rounds over any d-regular graph network with a mixing time m, assuming each link is a BSC_ε.*

For any graph with a constant mixing time $m = O(1)$ (and thus, with $d \geq n^\alpha$, for some constant $\alpha > 0$), Theorem 5.2 determines that we can solve the neighborhood connectivity task in $O(1)$ rounds and success probability $1 - 2^{-n^{\Omega(1)}}$. By Corollary 3.4 the above implies a constant rate coding scheme that succeeds with subexponentially high probability for any multiparty interactive protocol over G with the same parameters. In the general case, as long as $d > \log^{1+\Omega(1)} n$, Theorem 5.2 and Corollary 3.4 prove Theorem 1.2.

The proof of Theorem 5.2 is composed of two main parts. First, in Section 5.1 we show that for any not-too-large list of pairs of nodes $\{(s_i, t_i)\}$, it is possible to find *disjoint* paths between any s_i and t_i, such that the length of each such path is $2m$. Then, in Section 5.2 we use these disjoint paths (along with some standard coding) to reliably relay large chunks of information between any two parties.

5.1 Finding disjoint paths

In this subsection we show how to find short edge disjoint paths for any list $\{(s_i, t_i)\}$ of source nodes and target nodes, such that no node appears more than $O(d/m)$ times in the list. It is important that the paths are edge disjoint so that the coding scheme could send messages from each source s_i to its target t_i without colliding with a message sent from some other s_j to t_j.

More precisely, note that we do not need the paths to be fully disjoint. Assume all the paths are of the same length, say ℓ. Sending a message from s_i to t_i would take ℓ rounds, where at the k-th round $1 \leq k \leq \ell$ the k-th link in the path is utilized, and the others are not. Therefore, it suffices that every two paths are disjoint in their k-th edge, for all $k \in [1, \ell]$. We denote such paths, whose k-th edges are disjoint (for all $k \in [1, \ell]$), as *time-multiplexed* edge disjoint.

Theorem 5.3. *Let $G = (V, E)$ be a d-regular graph with mixing time m. Let $\mathcal{L} = \{(s_i, t_i)\}$ be a list of pairs of nodes, $s_i, t_i \in V$, such that every $u \in V$ appears at most $\frac{d}{1600m}$ times in the list. Then, one can efficiently construct a set of* time-multiplexed edge disjoint *paths of length $2m$, connecting each $(s_i, t_i) \in \mathcal{L}$.*

Proof. Fix a pair (s_i, t_i). We count the number of paths of length $2m$ connecting these two nodes. For any two nodes u, v define $\mathsf{P}_{u,v}(m)$ to be the set of all paths of length m connecting u and v. Note that the paths are not necessarily simple, and are allowed to intersect themselves or even repeat edges.

Claim 5.4. *For any $u, v \in V$,*

$$\frac{1}{2n} d^m \leq |\mathsf{P}_{u,v}(m)| \leq \frac{2}{n} d^m.$$

Proof. Recall that G has a mixing time of m. Therefore, beginning at any node u, the probability to reach v after m (uniformly random) steps is $\frac{1}{n} \pm \frac{1}{2n}$. Since there are d^m different paths of length m starting at u, the claim follows. □

Now, set $\tilde{\mathsf{P}}_{s_i,t_i}(2m) = \bigcup_{v \in V} \mathsf{P}_{s_i,v}(m) \times \mathsf{P}_{v,t_i}(m)$ to be the set of all the paths of size $2m$ composed as an m-long path from s_i to some middle point v concatenated to an m-long path from v to t_i. Clearly,

$$\frac{1}{4n} d^{2m} \leq |\tilde{\mathsf{P}}_{s_i,t_i}(2m)| \leq \frac{4}{n} d^{2m}.$$

Next, we would like to choose, for each i, one path out of $\tilde{\mathsf{P}}_{s_i,t_i}(2m)$ so that the collection of joint paths are time-multiplexed edge disjoint. This can be done using the combinatorial result from [Alo88], see also [AS08], Propostion 5.5.3, applied to the coincidence graph H described below. It is also similar to the approach of [BFU94]. For completeness, we describe the argument, which proceeds by bounding the dependency between the events of two paths sharing an edge, and then by using the Lovász Local Lemma to prove there exists a set of paths that are jointly time-multiplexed edge disjoint. The details follow.

Define the following coincidence graph $H = (V', E')$. For every i, every path in $\tilde{\mathsf{P}}_{s_i,t_i}(2m)$ becomes a node in H, that is,

$$V' = \{p_{i,j} \mid p_{i,j} \text{ is the } j\text{-th path in } \tilde{\mathsf{P}}_{s_i,t_i}(2m)\}.$$

The edges E' are defined as follows. For any i and $i' \neq i$, we connect the node $p_{i,j}$ with $p_{i',j'}$ if, for some $1 \leq k \leq 2m$, these two paths share the k-th edge. We say that such paths are *k-time-colliding*.

Claim 5.5. *The degree of each node in H is at most $\frac{d^{2m}}{400n}$.*

Proof. In the following we will fix a path p^* (i.e., a node in H) and bound the number of paths that k-time-collide with p^*, for some $k \in [1, 2m]$. Denote $p^* = (e_1, e_2, \ldots, e_{2m})$, and consider the k-th edge, e_k. Let us assume that $k \leq m$ (the other case is symmetric). First, we note that there are exactly d^{m-1} paths of length m, in which e_k is the k-th edge. Denote them as,

$$P(e_k, k) \triangleq \left\{(e'_1, \ldots, e'_{k-1}, e_k, e'_{k+1}, \ldots, e'_m) \mid e'_j \in E\right\}.$$

If we fix a specific $p \in P(e_k, k)$, and assume its end nodes are (u, v), we can ask how many nodes in H have p as their first half. Since fixing p fixes a specific starting node u and this node can appear at most $d/1600m$ times in $\mathcal{L}$, it follows that p is the first half of at most

$$\frac{d}{1600m} \times \frac{2}{n} d^m$$

paths in H. Summing over all possible p's, the number of paths (nodes in H) whose first half is some path in $P(e_k, k)$ is bounded by $\frac{d^{2m}}{800nm}$. This is also the number of paths that k-time-collide with p^*. Summing over all $k \in [1, 2m]$ completes the claim. □

Given the bound on the degree of each node in H, we can use the Lovász Local Lemma ([EL75], cf. also [AS08], Chapter 5) to show that we can pick, for every i, one node $p_{i,j} \in V'$ so that we obtain an independent set. Such an independent set implies non-colliding paths in G.

Lemma 5.6 (Lovász Local Lemma [EL75])**.** *Let $\{A_i\}$ be a finite set of events. If,*

(1) $\forall i,\ \Pr[A_i] \leq p$,
(2) $\forall i,\ A_i$ *is mutually independent of all other events but at most d, and*
(3) $e(d+1)p < 1$,

then, $\Pr[\overline{A_1} \wedge \overline{A_2} \wedge \cdots] > 0$.

Assume that for each (s_i, t_i) we choose one of the paths in $P_{s_i,t_i}(2m)$ at random. For any two time colliding paths $p_{i,j}$, $p_{i',j'}$ denote by $A_{ij,i'j'}$ the bad event that we choose $p_{i,j}$ for (s_i, t_i) and $p_{i',j'}$ for $(s_{i'}, t_{i'})$, thus

$$\Pr[A_{ij,i'j'}] \leq \frac{1}{|\tilde{\mathsf{P}}_{s_i,t_i}(2m)|} \cdot \frac{1}{|\tilde{\mathsf{P}}_{s_{i'},t_{i'}}(2m)|} \leq \frac{16n^2}{d^{4m}}.$$

Each such a bad event is independent of all other events $A_{ab,cd}$ besides those with either $a = i$ or $c = i'$ (note that $A_{ab,cd}$ is the same as $A_{cd,ab}$). Since each path collides with at most $\frac{d^{2m}}{400n}$ other paths (Claim 5.5), each such bad event is independent of all but at most

$$\deg(A_{ij,i'j'}) \leq \left(|\tilde{\mathsf{P}}_{s_i,t_i}(2m)| + |\tilde{\mathsf{P}}_{s_{i'},t_{i'}}(2m)|\right) \cdot \frac{d^{2m}}{400n} \leq \frac{d^{4m}}{50n^2}$$

others. It is easy to verify that the conditions of Lemma 5.6 are satisfied,

$$e \cdot \Pr[A_{ij,i'j'}] \cdot (\deg(A_{ij,i'j'}) + 1) < 1,$$

which implies we can pick paths that connect all pairs in $\mathcal{L}$ such that no two paths are time-colliding. We note that finding such a set in our case can be done with high probability in quasilinear time $\tilde{O}(n)$, via the algorithm of Moser and Tardos [MT10]. A deterministic construction with polynomial time is possible as well, as mentioned in [Alo91], see also [CGH13]. □

5.2 The coding scheme

The ability to find many time-multiplexed edge disjoint paths allows us to communicate large chunks of information by relaying them through intermediate points in a way that resembles the approach of Section 4. Specifically, if each party sends and receives $O(d)$ bits from at most $O(d/\Lambda)$ different parties the communication can be done in a reliable way, except with probability $n^2 2^{-\Omega(\Lambda)}$.

Definition 5.7. *A* communication demand *for a network with n parties, is a matrix $A \in \mathbb{N}^{n\times n}$ such that $a_{i,j}$ describes the amount of bits party i wishes to send to party j.*

Proposition 5.8. *Consider a d-regular graph G with mixing time m. Let A be a communication demand matrix, and assume that for any j, $\sum_i a_{i,j} \leq d$ and for any i, $\sum_j a_{i,j} \leq d$. Furthermore assume that every row and column in A has at most $O(d/\Lambda)$ non-zero elements, for some parameter $\Lambda > 1$. Then, for any $\varepsilon < 1/2$, there exists a communication protocol that fulfills the demand A in $O(m^2 \log m)$ rounds, and succeeds with probability $1 - n^2 2^{-\Omega(\Lambda)}$, over a network G where each link is a BSC_ε.*

Proof. The idea of the coding scheme is to send each chunk of information (i.e., each $a_{i,j}$ bits defined by the demand A), via $O(d)$ disjoint paths given by Theorem 5.3. However, note that each path given by that theorem consists of $2m$ cascaded BSC_ε. Such a cascade flips each bit with probability $(1 - (1-2\varepsilon)^{2m})/2$, and thus has a capacity $C \leq (1-2\varepsilon)^{4m}$, see e.g., [CT06, Chapter 7]. To overcome this error we can use a standard error correction, yet this will incur a blowup of $2^{O(m)}$ which we can substantially reduce by adding another layer of (trivial) encoding/decoding per BSC_ε link. Indeed, in the following assume that each bit we communicate through a BSC_ε is first encoded to length $O(\log m)$, sent over the network in $O(\log m)$ rounds and then decoded at the other side of the link. Effectively, this reduces the error at each link to $1/m$, so each link can be seen as $\mathsf{BSC}_{1/m}$. Cascading $2m$ such channels is equivalent to BSC_γ with $\gamma = (1 - (1-2/m)^{2m})/2 \approx (1 - e^{-4})/2$, that is, the error level is bounded by some constant independent of m.

The protocol for reliably delivering the communication demand A goes as follows. For any given i, j, the i-th party encodes its $a_{i,j}$ bits targeted to the j-th party, using a Shannon error correction code that succeeds with probability $1 - 2^{-\Omega(\Lambda)}$ assuming a BSC_γ. Using Lemma 2.2, there exists such a code that encodes $a_{i,j}$ bits into $\tilde{a}_{i,j} = O(a_{i,j} + \Lambda)$ bits. Observe that after this encoding, each party i holds $\sum_j \tilde{a}_{i,j} = \sum_j O(a_{i,j} + \Lambda) = O(d)$ bits to communicate, since it is guaranteed that the i-th party has at most $O(d/\Lambda)$ parties j for which $a_{i,j} \neq 0$, and that $\sum_j a_{i,j} \leq d$. Let $\tilde{A}$ be the communication demand defined by the above $\tilde{a}_{i,j}$.

Next, define $O(m)$ matrices $\{B_k\}$ such that any row and column in B_k sums up to at most $d/1600m$ and such that $\sum_k B_k = \tilde{A}$. This can be done (efficiently) by König's Theorem, using any efficient algorithm for edge coloring bipartite graphs. Communicating $\tilde{A}$ is equivalent to communicating all the demands $\{B_k\}$, which we will do in a sequential manner. Each such B_k defines a list $\mathcal{L}_k$ in which the pair (i, j) appears exactly $(b_{i,j})_k$ times in $\mathcal{L}_k$. Since the sum of each row and column in B_k is bounded by $d/1600m$, the list satisfies the conditions of Theorem 5.3. Thus, the demand described by B_k can be transmitted by a sequence of $2m$ bit-transmissions over the noisy network. Moreover, since there are at most $O(m)$ many B_k's, transmitting all of them sequentially (i.e., unreliably fulfilling the communication demand defined by $\tilde{A}$) can be done in $O(m^2)$ bit-transmissions. Recall that each bit-transmission consists of $O(\log m)$ rounds of communication, and we get that the entire process takes $O(m^2 \log m)$ rounds.

Last, we show that the entire process implies a reliable communication of the demand A. Recall that each chunk of information in $\tilde{A}$ is encoded to resist the noise γ, that is, the noise induced by transferring each bit through the path of $2m$ consecutive independent $\mathsf{BSC}_{1/m}$ links. It follows that each encoded chunk (i.e., each $\tilde{a}_{i,j}$) is decoded correctly with probability $1 - 2^{-\Omega(\Lambda)}$. A union bound on all the ($< n^2$) different encoded transmissions $\{\tilde{a}_{i,j}\}$ gives the claimed success probability. □

As a corollary of the above theorem, note that the same result holds for any communication demand A in which each row and column sums up to $K \cdot d$ rather than d, in $O(K \cdot m^2 \log m)$ rounds. In that case, after encoding each chunk we get the demand $\tilde{A}$ in which the sum of each row or column is $O(Km^2)$, and thus it can be transmitted in $O(Km^2 \log m)$ rounds.

We are now ready to complete the proof of Theorem 5.2.

Proof. (*Theorem 5.2*). Let A_0 be the communication demand induced by the neighborhood connectivity, thus for any j, $\sum_i a_{i,j} = d$ and for any i, $\sum_j a_{i,j} = d$, since the network's graph is d-regular. The number of non-zero elements in every row or column is d, so we cannot apply Proposition 5.8 directly. Instead, we perform a sequence of relays[2], using Proposition 5.8, that converts the initial communication demand A_0 into one that satisfies the conditions of Proposition 5.8, i.e., where each party has at most d bits to send to at most $O(d/\Lambda)$ different destinations. Specifically, there is a sequence of ℓ matrices $A_0, A_1, \ldots, A_\ell$ such that for any $l \in [\ell]$ we convert A_{l-1} into A_l in $O(m^2 \log m)$ rounds, and where it holds that (1) each A_l is a block-diagonal matrix with block size at most $n(d/\Lambda)^{-l}$; (2) the sum of each column in A_l is d; and (3) the sum of each row in A_l is at most $4d$.

Lemma 5.9. *Let the communication demand $A_l \in \mathbb{N}^{n \times n}$ be block diagonal with block size b, and assume that $\sum_i (a_{i,j})_l \leq d$ and $\sum_j (a_{i,j})_l \leq 4d$. Then, it is possible to reliably relay information by $O(m^2 \log m)$ rounds of communication, so that $A_{l+1} \in \mathbb{N}^{n \times n}$ describing the communication demand after the relay is block diagonal with block size at most $b\frac{\Lambda}{d}$, and it holds that $\sum_i (a_{i,j})_{l+1} \leq d$ and $\sum_j (a_{i,j})_{l+1} \leq 4d$.*

Proof. Consider the r-th block of A_l, that is, the submatrix that contains the communication between parties $P_r \equiv \{(r-1)b+1, (r-1)b+2, \ldots, rb\}$; note that these parties wish to send information only between themselves due to the block diagonal form of A_l. Therefore, we can treat each such block independently.

Split P_r into d/Λ disjoint subsets $P_{r,1}, \ldots, P_{r,d/\Lambda}$ of equal size, in an arbitrary way. For any $j = 1, \ldots, d/\Lambda$, each $p_i \in P_r$ will send *all* the information directed to parties in $P_{r,j}$, i.e. $\sum_{u \in P_{r,j}} a_{i,u}$ bits, to a single party of $P_{r,j}$. The recipient can be chosen in a "greedy" way: order the parties of $P_{r,j}$ in some order, say $p'_1, p'_2, \ldots$; iterate over all $p_i \in P_r$ in an increasing order of the demand $\sum_{u \in P_{r,j}} a_{i,u}$, and determine the recipient as the first $p' \in P_{r,j}$ that (i) is currently scheduled to receive bits from less than $2d/\Lambda$ different parties, and (ii) will receive at most $4d$ bits (including the $\sum_{u \in P_{r,j}} a_{i,u}$ bits held by p_i). Under these restrictions, we can use Proposition 5.8 to perform the relay in $O(m^2 \log m)$ rounds in a reliable way (with high probability).

To see that the greedy algorithm succeeds in accommodating the demand of all $p_i \in P_r$, split P_r into parties for which $\sum_{u \in P_{r,j}} a_{i,u} \leq \Lambda$, denoted as the subset $P_< \subseteq P_r$, and the other parties, denoted $P_>$. Parties in $P_<$ are aggregated in groups of size $2d/\Lambda$. Note that the joint demand of each such group never exceeds $2d$ bits. We need $|P_<|/(2d/\Lambda) \leq b\Lambda/2d$ parties $p' \in P_{r,j}$ to accommodate all these groups. Parties from $P_>$ are aggregated until adding an additional party causes the demand to exceed $4d$ bits of information. Since we order the parties according to an increasing order of demand, each assigned p' (maybe, except one) accommodates at least $2d$ bits of demand (and at most $4d$ bits). Also note that the total amount of bits to be sent by parties in $P_>$ (to a specific $P_{r,j}$) is at most $b\Lambda$, since each column in A_l sums up to at most d bits, and there are at most $b/(d/\Lambda)$ columns to be considered here. Thus, in order to accommodate all $P_>$ we need at most $b\Lambda/2d$ different parties $p' \in P_{r,j}$. Summing these two parts, the total number of parties in $P_{r,j}$ needed to accommodate both $P_<$ and $P_>$, is bounded by $b\Lambda/d \leq |P_{r,j}|$.

[2]In *relay* we mean that if party i wants to send a bit to party j, it can send that bit to some party k who will later relay that bit to j. Thus, after sending the bit to k, the communication demand changes so that $a_{i,j} = 0$ and $a_{k,j}$ increases by one.

The above is being performed, in parallel, for each block of A_l. Recall that the block-diagonal form of A_l implies a partition of the parties into disjoint sets $\{P_r\}$ corresponding to each block, where each relay happens only within the block. Moreover, within each block each party communicates at most $O(d)$ bits and at most $O(d/\Lambda)$ different parties. Thus the joint communication demand satisfies the conditions of Proposition 5.8, and we can perform the relay in $O(m^2 \log m)$ rounds, reliably. At the end of this process, each block of size b is transformed into d/Λ disjoint blocks of size at most $b\Lambda/d$. Note that all the parties reliably receive the information relayed to them, except with a probability of $n^2 2^{-\Omega(\Lambda)}$. □

We perform the above process of Lemma 5.9 recursively, starting from A_0 being the incidence matrix of the the underlying network (i.e., the demands induced by the Neighbor task). If we set $\Lambda = d^c$ for some constant $c < 1$, then after $\ell = m/(1-c) - O(1)$ steps we reach a matrix A_ℓ that is block diagonal with block size at most

$$n(d/\Lambda)^{-\ell} = nd^{-m}\frac{d}{\Lambda} \leq d/\Lambda$$

Specifically, each column in A_ℓ sums up to d, each row sums to $4d$, and at each row/column there are $O(d/\Lambda)$ non-zero element. We can now use Proposition 5.8 one last time to fulfill the communication demand A_ℓ and complete the proof. The entire scheme takes $O((\ell+1)\cdot m^2 \log m) = O(m^3 \log m)$ rounds, and succeeds with probability $1 - mn^2 2^{-\Omega(\Lambda)} = 1 - mn^2 2^{-\Omega(d^c)}$. To succeed with high probability we require $d > \log^{1+\alpha} n$, for some $\alpha > 0$ and $c > 1/(1+\alpha)$.

It can be easily verified that each step of the protocol is computationally efficient, which makes the entire simulation efficient. □

References

[AGS13] S. Agrawal, R. Gelles, and A. Sahai. Adaptive protocols for interactive communication. Manuscript, arXiv:1312.4182 (cs.DS), 2013.

[Alo88] N. Alon. The linear arboricity of graphs. *Israel J. Math.*, 62(3):311–325, 1988.

[Alo91] N. Alon. A parallel algorithmic version of the local lemma. *32nd Annual Symposium on Foundations of Computer Science (San Juan, PR, 1991)*, pp. 586–593. IEEE Comput. Soc. Press, Los Alamitos, CA, 1991.

[AC07] N. Alon and M. Capalbo. Finding disjoint paths in expanders deterministically and online. *48th Annual Symposium on Foundations of Computer Science, 2007)*, pp. 518–524. IEEE Comput. Soc. Press, Los Alamitos, CA, 2007.

[AS08] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., Hoboken, NJ, third edn., 2008. With an appendix on the life and work of Paul Erdős.

[BK12] Z. Brakerski and Y. T. Kalai. Efficient interactive coding against adversarial noise. *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pp. 160–166, 2012.

[BN13] Z. Brakerski and M. Naor. Fast algorithms for interactive coding. *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pp. 443–456. 2013.

[BE14] M. Braverman and K. Efremenko. List and unique coding for interactive communication in the presence of adversarial noise. *Proceedings of the IEEE Symposium on Foundations of Computer Science*, FOCS '14, pp. 236–245. 2014.

[BEGH15] M. Braverman, K. Efremenko, R. Gelles, and B. Haeupler. Constant-rate coding for multiparty interactive communication is impossible. ECCC report TR15-197, 2015. To appear in STOC'16.

[BR11] M. Braverman and A. Rao. Towards coding for maximum errors in interactive communication. *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pp. 159–166. ACM, New York, NY, USA, 2011.

[BR14] M. Braverman and A. Rao. Toward coding for maximum errors in interactive communication. *Information Theory, IEEE Transactions on*, 60(11):7248–7255, 2014.

[BFU94] A. Broder, A. Frieze, and E. Upfal. Existence and construction of edge-disjoint paths on expander graphs. *SIAM Journal on Computing*, 23(5):976–989, 1994.

[CGH13] K. Chandrasekaran, N. Goyal, and B. Haeupler. Deterministic algorithms for the lovász local lemma. *SIAM Journal on Computing*, 42(6):2132–2155, 2013.

[CPT13] K.-M. Chung, R. Pass, and S. Telang. Knowledge-preserving interactive coding. *Proceedings of the 54th annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 449–458. 2013.

[CT06] T. M. Cover and J. A. Thomas. *Elements of Information Theory (2nd edition)*. John Wiley & Sons, 2006.

[DLP12] D. Dolev, C. Lenzen, and S. Peled. tri, tri again: Finding triangles and small subgraphs in a distributed setting. M. Aguilera, ed., *Distributed Computing*, *Lecture Notes in Computer Science*, vol. 7611, pp. 195–209. Springer Berlin Heidelberg, 2012.

[EGH15] K. Efremenko, R. Gelles, and B. Haeupler. Maximal noise in interactive communication over erasure channels and channels with feedback. *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science*, ITCS '15, pp. 11–20. 2015.

[EL75] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets*, 10:609–627, 1975.

[FGOS13] M. Franklin, R. Gelles, R. Ostrovsky, and L. J. Schulman. Optimal coding for streaming authentication and interactive communication. *Advances in Cryptology – CRYPTO 2013*, *LNCS*, vol. 8043, pp. 258–276. Springer, 2013.

[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.

[GH15] R. Gelles and B. Haeupler. Capacity of interactive communication over erasure channels and channels with feedback. *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pp. 1296–1311. 2015.

[GMS11] R. Gelles, A. Moitra, and A. Sahai. Efficient and explicit coding for interactive communication. *Proceeding of the IEEE Symposium on Foundations of Computer Science*, FOCS '11, pp. 768–777. 2011.

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

[GSW14] R. Gelles, A. Sahai, and A. Wadia. Private interactive communication across an adversarial channel. *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS '14, pp. 135–144. ACM, 2014.

[GH14] M. Ghaffari and B. Haeupler. Optimal Error Rates for Interactive Coding II: Efficiency and List Decoding. *Proceedings of the IEEE Symposium on Foundations of Computer Science*, FOCS '14, pp. 394–403. 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. ACM, New York, NY, USA, 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.

[GI05] V. Guruswami and P. Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Trans. on Information Theory*, 51(10):3393–3400, 2005.

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

[HS14] W. M. Hoza and L. J. Schulman. The adversarial noise threshold for distributed protocols, 2014. ArXiv:1412.8097. [Online:] http://arxiv.org/abs/1412.8097.

[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 Symposium on theory of computing*, pp. 715–724. ACM, New York, NY, USA, 2013.

[Len13] C. Lenzen. Optimal deterministic routing and sorting on the congested clique. *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*, PODC '13, pp. 42–50. ACM, New York, NY, USA, 2013.

[MT10] R. A. Moser and G. Tardos. A constructive proof of the general lovász local lemma. *J. ACM*, 57(2):11:1–11:15, 2010.

[Pan13] D. Pankratov. On the power of feedback in interactive channels. [Online:] http://people.cs.uchicago.edu/~pankratov/papers/feedback.pdf, 2013.

[Raj94] S. Rajagopalan. *A Coding Theorem for Distributed Computation*. Ph.D. thesis, University of California at Berkeley, Berkeley, CA, USA, 1994. UMI Order No. GAX95-29464.

[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. ACM, New York, NY, USA, 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.

[Sch93] L. J. Schulman. Deterministic coding for interactive communication. *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pp. 747–756. ACM, New York, NY, USA, 1993.

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

[Sha48] 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.

[Spi95] D. A. Spielman. Linear-time encodable and decodable error-correcting codes. *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, STOC '95, pp. 388–397. ACM, New York, NY, USA, 1995.

Appendix

A Proof of Theorem 4.1

We give here the detailed proof of Theorem 4.1.

Proof. Consider the n^2 bits $\{a_{i,j}\}$ that have to be sent as the $n \times n$ matrix A. Let $k = \sqrt{n}$, and split the matrix A into n disjoint sub-matrices $B_{t,l}$, each of size $k \times k$. Specifically,

$$A = \begin{pmatrix} B_{1,1} & B_{1,2} & \cdots & B_{1,k} \\ B_{2,1} & B_{2,2} & \cdots & B_{2,k} \\ & & \ddots & \\ B_{k,1} & B_{k,2} & \cdots & B_{k,k} \end{pmatrix},$$

where for any $t, l \in [k]$ we let

$$B_{t,l} = \begin{pmatrix} a_{(t-1)k+1,(l-1)k+1} & a_{(t-1)k+1,(l-1)k+2} & \cdots & a_{(t-1)k+1,lk} \\ a_{(t-1)k+2,(l-1)k+1} & a_{(t-1)k+2,(l-1)k+2} & \cdots & a_{(t-1)k+2,lk} \\ \vdots & & \ddots & \vdots \\ a_{tk,(l-1)k+1} & a_{tk,(l-1)k+2} & \cdots & a_{tk,lk} \end{pmatrix}.$$

Associate each block $B_{t,l}$ with a responsible party $f(t,l) \in [n]$ in a bijective way, without loss of generality, we can take $f(t,l) = k(t-1) + l$. The protocol proceeds in two steps, each of which takes $O(1)$ rounds. In the first step, each party $f(t,l)$ learns all the bits that belong to $B_{t,l}$ with high probability. In the second step, each such party distributes the bits of the appropriate matrix $B_{t,l}$ to their destinations.

The protocol uses an error correcting code $\mathsf{ECC} : \{0,1\}^k \to \{0,1\}^{k'}$ that, assuming a BSC_ε, fails with probability at most $2^{-\Omega(k)}$; note that $k' = O_\varepsilon(k)$ (see Lemma 2.2).

To explain the first step suppose, first, that we do not use the code and each party i wishes to send the bit $a_{i,j}$ to party j. This can be done in two rounds as follows. In the first round, party i sends the bit $a_{i,j}$ to party $(ki + j) (\mathrm{mod}\ n)$. Note that here, crucially, each party i sends exactly one bit to each other party, and that for each block $B_{t,l}$, no two bits of $B_{t,l}$ reach the same party. Thus the bits of $B_{t,l}$ reach all the n parties, each getting exactly one bit. In the second step, each party sends the unique bit from $B_{t,l}$ that it received to the responsible party $f(t,l)$. Clearly after these two steps, $f(t,l)$ receives all bits of the block $B_{t,l}$. A similar routing technique appears in [DLP12, Len13].

In the actual realization of the first step, each party i applies the code from Lemma 2.2 to the k bits in each block in its row before transmitting them. Thus the bits $a_{i,(l-1)k+1}, a_{i,(l-1)+2}, \ldots, a_{i,lk}$ are first encoded to get k' bits, which are sent using the above procedure, simulating each of the two rounds by $\lceil k'/k \rceil$ rounds. Once the party $f(t,l)$ gets the bits of the encoded rows of the block $B_{t,l}$, he can decode them, using the error correcting code, and get all the bits of the block $B_{t,l}$ correctly, with high probability.

The second step is performed in the same way, reversing the directions. Ignoring the encoding, this can be done in two rounds as follows. In the first round, each party $f(t,l)$ responsible to the bits in the block $B_{t,l}$ sends the bit $a_{i,j}$ of this block to party number $(i + kj)(mod\ n)$. Note that no two bits of the block have to be sent to the same destination, and hence this can indeed be performed in one round. In the second round the party that got the bit $a_{i,j}$ sends it to its destination: party number j. Since for distinct i, i', $(i + kj)(mod\ n)$ is not equal to $(i' + kj)(mod\ n)$ this can also be done in one round. To ensure that with high probability no errors will occur, the party $f(t,l)$ encodes each column of its block $B_{t,l}$ before sending the bits of the encoded message. As before, after the encoding each of these two rounds can be simulated by a constant number of rounds. This completes the proof. □