Electronic Colloquium on Computational Complexity, Report No. 111 (2019)

Noisy Beeps

Klim Efremenko* Ben-Gurion University

Gillat Kol† Princeton University

Raghuvansh R. Saxena‡ Princeton University

Abstract

We study the effect of noise on the n-party *beeping model*. In this model, in every round, each party may decide to either 'beep' or not. All parties hear a beep if and only if at least one party beeps. The beeping model is becoming increasingly popular, as it offers a very simple abstraction of wireless networks and is very well suited for studying biological phenomena. Still, the noise resilience of the beeping model is yet to be understood.

Our main result is a lower bound, showing that making protocols in the beeping model resilient to noise may have a large performance overhead. Specifically, we give a protocol that works over the (noiseless) beeping model, and prove that any scheme that simulates this protocol over the beeping model with correlated stochastic noise will blow up the number of rounds by an $\Omega(\log n)$ multiplicative factor.

We complement this result by a matching upper bound, constructing a noise-resilient simulation scheme with $\mathcal{O}(\log n)$ overhead for any noiseless beeping protocol.

1 Introduction

In the beeping model [CK10], a set of n parties interact by beeping in synchronous rounds. In every round, each party can decide to beep (emit a signal) or not to beep (be silent). If at least one party beeps, all parties hear a beep, otherwise, all parties hear silence. The (noiseless) beeping model, and other related models, have received a lot of attention in recent years [AABJ11, SJX13, AAB+13, HM13, FSW14, GN15, MRZ15, BKK+16, HL17, DBB18, *etc.*], largely as they provide an abstraction capturing the simplest possible communication primitive: a detectable burst of 'energy'. This abstraction is very well suited for describing wireless networks. In addition, one of the main motivations for studying the beeping model is due to its connections to signal-driven biological systems [AAB+11, NB15], *e.g.*, cells

*Supported by the Israel Science Foundation (ISF) through grant No. 1456/18.

†Research supported by an Alfred P. Sloan Fellowship, the National Science Foundation CAREER award CCF-1750443, and by the E. Lawrence Keyes Jr. / Emerson Electric Co. Award.

‡Research supported by the National Science Foundation CAREER award CCF-1750443.

ISSN 1433-8092

communicating by secreting proteins and other chemical markers that are diffused and sensed by neighboring cells, or fireflies reacting to flashes of light from nearby fireflies.

The beeping model can also be viewed as a multi-party generalization of Blackwell's binary *multiplication channel*, an extensively studied channel in the information theory community. In this channel, there are two parties, and in every round, each party sends a bit and both parties receive the 'and' (or, equivalently, the 'or') of the two sent bits. By viewing a beeping party as sending a 1, and a silent party as sending a 0, we get that, in each round, the beeping model outputs the 'or' of the bits sent.

In this paper, we initiate the study of the noise tolerance of the beeping model. For a constant $\epsilon > 0$, we define the *n-party ϵ-noisy beeping model* as a model where, in every round, all the parties receive the 'or' of the n bits sent in this round, with probability $1 - \epsilon$, and with probability ϵ, the parties receive the negation of this 'or', independently for all rounds. We then consider the question: let Π be a protocol that was designed to work over the (noiseless) beeping model. Is it possible to simulate Π by a protocol Π', such that Π' is not much longer than Π, and works over the ϵ-noisy beeping model?

1.1 Our Results

While, as mentioned above, one of the attractive features of the beeping model is its simplicity and robustness, it is still prone to errors. Our main result is that making protocols over the beeping model resilient to noise may require a substantial performance overhead. Specifically, simulating a noiseless protocol Π by a noise resilient protocol can require a logarithmic (in n) blowup in length.

Theorem 1.1 (main). *Let $\epsilon \geq 0$ be a constant. There exists a (deterministic) protocol Π over the n-party noiseless beeping model of length $T(n)$, such that any protocol Π' (even with shared randomness) that simulates Π in the n-party ϵ-noisy beeping model, requires $\Omega(T(n) \log n)$ rounds.*

We mention that our lower bound is proved for the more relaxed one-sided noise model, where the channel may only flip the 'or' when it evaluates to 0 (*i.e.,* can change a 0 to a 1, but never changes a 1 to a 0), see subsubsection A.1.2 and Theorem C.1. Curiously, this statement is asymmetric - one can construct a simulation protocol with constant blowup when the noisy channel only changes 1s to 0s, see section 2.

We complement this lower bound result by a matching upper bound: a simulation protocol with a blowup of at most $\mathcal{O}(\log(n))$ to the number of rounds. While simulations with such blowups for any protocol length[1] are typically easy to obtain using the, by now standard, rewind-if-error mechanism, the protocol in Theorem 1.2 requires an additional idea, see subsection 2.1.

[1] Observe that protocols of length polynomial in n can trivially be simulated by repeating every round $\mathcal{O}(\log(n))$ times and taking the majority.

Theorem 1.2. *Let $\epsilon \geq 0$ be a constant. For every protocol Π over the n-party noiseless beeping model of length $T(n)$, there exists a protocol Π' that simulates[2] Π in the n-party ϵ-noisy beeping model, and only requires $\mathcal{O}(T(n) \log n)$ rounds.*

1.2 Noise Type

This paper considers the beeping model in the presence of *correlated noise*: when a bit is flipped due to the noise in the channel, all parties receive the flipped bit. The channel ensures that all parties receive the same bit in every round, and thus, they have the same transcript. This stands in contrast to the *independent noise* model, where, in each round, each party gets an independent ϵ-noisy copy of the 'or' of the bits sent by the parties in that round. In this model, parties may witness different transcripts. The correlated noise model describes the situation where there are global interferences (*e.g.*, global network problems due to weather, contaminated environment, *etc.*), while independent noise is better suited to describe situations where the noise source is local.

The correlated and the independent noise models are (at least seemingly) incomparable, and it is not clear how to "translate" a protocol assuming one of these models to a protocol in the other model. From a protocol design point of view, the correlated noise model has the advantage of the parties agreeing on the (noisy) transcript of the protocol. However, this also means that, for some of the rounds, no party knows the correct 'or'. In contrast, while the independent noise model suffers from uncorrelated transcripts, the correct 'or' of every round will be received by almost all the parties, and these may be able to convey it to the rest of the parties.

Understanding the resilience of the beeping model under independent noise is an obvious interesting problem left open by our work. While Theorem 1.2 applies to the independent noise setting as well, the proof of Theorem 1.1 breaks. Furthermore, inspired by [Gal88], one can show that the protocol Π that proves Theorem 1.1, admits a simulation protocol with only $\mathcal{O}(\log \log(n))$ blowup in length in the independent noise model. However, it is our belief that with a different example (*e.g.*, a variant of pointer chasing), a super-constant lower bound on the blowup can be proved for independent noise as well.

1.3 Related Works

Interactive Coding. The field of coding for interactive communication was introduced in the seminal papers of Schulman [Sch92, Sch93, Sch96]. Various aspects of two-party interactive coding (such as computational efficiency, interactive channel capacity, noise tolerance, list decoding, different channel types, *etc.*) were considered in recent years [GMS11, BR11, BKN14, Bra12, MS14, BE14, GMS14, GH14, GHK+16, EGH16, BGMO17, EKS18, to cite a few].

[2] except with probability polynomially small in n.

While two-party models for interactive coding were studied extensively, until recently, multi-party models were barely explored in this context. We view this paper as a part of a general effort by the authors and others to better understand the noise tolerance of multi-party (distributed) models.

Relation to [EKS18]. The work most relevant to this paper is the work of [EKS18], studying the noise tolerance of the broadcast channel. This work shows that every (non-adaptive) protocol over the (single-hop) broadcast channel can be compiled to a noise resilient protocol, by incurring only a constant multiplicative overhead. We point out that the [EKS18] protocol works for both the correlated and the independent noise models and makes no assumptions about the outputs of the channel if it is not the case that exactly one party broadcasts (*i.e.,* in case of collision, where more than one party is broadcasting in the same round, or silence, where no party is broadcasting, all the parties get adversarial bits).

While the noisy beeping model assumed by this paper is more powerful than the noisy broadcast model assumed by [EKS18] (here, collisions and silences are not adversarial, but are only subject to random noise), the [EKS18] simulation does not carry over to our setting (as is evident by our lower bound). The reason is that the additional power of the beeping model can also be used by the noiseless protocol we wish to simulate, and this is indeed the case for the protocol Π from Theorem 1.1. More generally, given two communication models, even if it is known that every protocol in the second model can be run over the first model as well, the interactive coding questions for these models are incomparable.

The Noise Resilience of Other Distributed Models. The first distributed model to be studied in the context of interactive coding is the point-to-point (*a.k.a.* private channels) model, for which a beautiful constant rate coding scheme was given by [RS94]. Variants of the point-to-point model were later explored by [JKL15, ABE+16, BEGH16, HS16, GK17, CGH18, GKR19].

As described above, the noisy broadcast channel has also been studied in the works of [Gam87, Gal88, GKS08, EKS18]. The noise tolerance of the broadcast channel was also studied in [FK00, KM05, New04, GGKN09, GHM18]. Interactive coding for multi-hop radio networks (in the "collision-as-silence" and related models) is studied in [CHHZ17, CHHZ18, EKS19]. The related dual graph model, where some of the links are unreliable, is studied by [KLN+10, GHLN12, CGK+14, GLN13], and [KKP01, KPP10] consider radio networks with faulty nodes.

We mention that, while some techniques can be carried over between models, interactive coding schemes, and even more so, interactive coding lower bounds, for each of the above models require novel ideas capturing the unique nature of the model.

2 Proof Sketch

2.1 The [EKS18] Interactive Coding Scheme

As mentioned in subsection 1.3, the main result in [EKS18] is a constant rate interactive coding scheme for simulating every (non-adaptive) noiseless broadcast protocol over the noisy adaptive broadcast channel. The scheme follows the popular 'rewind-if-error' approach [Sch92, *etc.*, for a survey, see [Gel17]]. The idea is to break the noiseless protocol into small chunks and simulate each chunk separately such that the probability of failure of a given chunk is polynomially small in the length of the chunk. After one chunk of the protocol is simulated, the parties run a 'verification phase' to verify that the chunk was simulated correctly. If it was, the parties proceed to simulate the next chunk of the noiseless protocol. Otherwise, the parties re-simulate this chunk.

During the verification phase of [EKS18] (interpreted for correlated noise), each party checks if the parties' shared noisy transcript agrees with the bits it communicated. This can be done easily, as they assume that the noiseless protocol is non-adaptive, meaning that the order of turns is pre-defined and each party 'owns' a disjoint set of bits in the transcript. A party that finds a mismatch in one of the bits it owns 'raises an error flag', causing the verification phase to fail. This idea is implemented by having each party compute a bit that evaluates to 1 if and only if it detects an error. The authors of [EKS18] then design a protocol, with polylogarithmic number of rounds, allowing the parties to compute the 'or' of these bits. Furthermore, they argue that an efficient (sublinear) protocol for the 'or' function is essential for our approach to work.

One property of the beeping channel is that there is an (extremely) efficient protocol for the 'or' of n bits, *i.e.*, there exists an efficient protocol that computes $\bigvee_i b^i$ where the bit b^i is known to player i. Thus, at first glance, it may seem that there is a constant rate interactive coding scheme for the beeping channel as well.

However, delving deeper reveals a problem: how will a party verify that the shared transcript π respects its beeping pattern? We say that a party beeped a 1 in a given round if it decided to beep, otherwise we say that it beeped a 0. If, for a round m, the bit $\pi_m = 0$, then, this is easily verifiable: each party can check that it indeed beeped a 0 in round m. But, it is unclear how to efficiently verify $\pi_m = 1$: consider the case that all parties beeped 0 in round m (but $\pi_m = 1$ due to noise). Each party may assume that one of the other parties beeped a 1. In other words, if the channel noise flipped the value of round m from a 1 to a 0, there will be at least one party that is able to detect the error by itself. However, it is unclear that a flip of 0 to 1 can be detected in the same manner.

The upshot of the discussion above is that in the beeping channel, the errors that change a 1 to a 0 are, in some sense, easier to handle than the errors that change a 0 to a 1. In particular, if there were no errors of the latter type, then a constant rate interactive coding scheme may be developed on the lines of [EKS18].

In the upper bound protocol, we show that with an $\Omega(\log n)$ overhead to the length of the

protocol, 0 to 1 flips can also be detected. An $\Omega(\log n)$ overhead scheme simulating chunks of polynomial length with inverse polynomial error can easily be obtained by repetition. However, to prove our upper bound we describe a procedure that not only simulates a given chunk to obtain a transcript π, but also computes an 'owner' for every round m where at least one party beeped a 1. This owner is one of the parties that beeped 1 in round m. Simulating a chunk this way allows us to implement the verification phases needed to make the rewind-if-error approach work for simulating protocols of all lengths. In our verification phase, it is the responsibility of the party that owns a given round m to raise an error flag for that round (*i.e.,* verify that $\pi_m = 1$). An error flag for rounds with no owner (all parties beeped 0) can be raised by any player.

2.2 The Lower Bound Construction

Our lower bound shows that detecting 0 to 1 flips cannot be done with $o(\log n)$ overhead. We turn this detection problem into a concrete communication task witnessing our lower bound. We draw inspiration from the problem of verifying a transcript π above. However, instead of trying to *verify* a transcript π, we consider the harder problem of *computing* such a π. We construct the following simple communication task: for all $m \in [2n]$, all players i have a bit b^i_m. The task is to compute $\pi_m = \bigvee_i b^i_m$ for all $m \in [2n]$. Observe how $b^i_1 b^i_2 \cdots b^i_{2n}$ corresponds to the sequence of bits beeped by party i in some noiseless protocol (and therefore also known to party i), while $\pi_1\pi_2\cdots\pi_{2n}$ corresponds to the correct transcript of this protocol (sequence of channel outputs).

The above task has a trivial beeping protocol with $2n$ rounds in the noiseless setting: in round $m \in [2n]$, party i beeps b^i_m.

To prove a lower bound on the length of a protocol solving the above task in the noisy setting, it turns out to be enough to consider a restricted version of this task. Namely, we show that π is hard to compute in the presence of noise, even under the promise that for all i, the bit b^i_m is 1 for exactly one m. In this case, the input of a player i can be equivalently described by the index $m \in [2n]$ such that $b^i_m = 1$. Indeed, this version, where all players get as input a number $x^i \in [2n]$ and compute $\mathcal{L}(x) = \{x^i \mid i \in [n]\}$[3] is the version we work with in this paper.

Interestingly, this lower bound construction lacks many of the complexities that a general interactive coding scheme has to tackle. For example, if indeed the bit b^i_m corresponds to the bit beeped by player i in round m of a protocol over the noiseless beeping channel, then this bit may depend on the bits $\pi_1\pi_2\cdots\pi_{m-1}$ received by player i. In our example, we assume that there is no such dependence and the bit b^i_m is fixed in advance. In other words, our example captures the problem of interactively coding very simple protocols, where the bits sent in round m are independent of the transcript received in the first $m-1$ rounds. We interpret the fact that the best possible lower bound can be obtained for a communication

[3]We use x to denote the input vector $x^1x^2\cdots x^n$.

task as simple as $\mathcal{L}$, as showing that even basic operations over the beeping model are becoming expensive in the presence of noise.

2.3 Sensitivity of the Function $\mathcal{L}$

The main reason why the function $\mathcal{L}$ is hard for a noisy beeping protocol is that it is highly sensitive to a change in one of its n inputs. Specifically, for a constant fraction of the inputs x, there exists a subset I of players such that $|I| = \Theta(n)$, and for any player $i \in I$, changing the input of player i changes the output of the function. To see why, let I be the set of players i such that only player i has the input x^i, *i.e.*, for all players $j \neq i$, it holds that $x^j \neq x^i$. It is easily seen that $|I| = \Theta(n)$ for a constant fraction of the inputs x. It also holds that changing the input of any player $i \in I$ changes the output of the function (as x^i is no longer an element of the output).

With the above in mind, define two inputs x, x' to be *neighbors* if there is at most 1 player i such that $x^i \neq x'^i$. Further, define $\mathcal{N}(x)$ to be the set of all neighbors x' of x such that $\mathcal{L}(x) \neq \mathcal{L}(x')$. The above discussion says that for a constant fraction of inputs x, we have $|\mathcal{N}(x)| = \Theta(n^2)$. For the rest of this proof sketch, we only work with inputs x such that $\mathcal{N}(x) = \Theta(n^2)$. We note that our actual proof does not deal with $\mathcal{N}$ explicitly and the use of $\mathcal{N}$ is limited to this sketch only.

2.4 The Proof of the Lower Bound

While our lower bound proof is not lengthy, it is technically quite challenging. The proof defines a measure for the information that the protocol obtains about the input. The measure is of the form $\frac{\Pr(x|\pi)}{\sum_{x' \in \mathcal{N}'(x)} \Pr(x'|\pi)}$, where $\mathcal{N}'(x) \subseteq \mathcal{N}(x)$ (see the definition of ζ in subsection C.2). Intuitively, protocols that solve $\mathsf{InputSet}_n$ should be able to differentiate between x and $x' \in \mathcal{N}'(x)$, as $\mathcal{L}(x) \neq \mathcal{L}(x')$. This is captured by Theorem C.3, showing that the above ratio is large for correct protocols. Theorem C.2 proves that short protocols can only lead to a small ratio, as, because of the noise, a round can only change the relative probabilities of x and x' by a constant. By combining Theorem C.2 and Theorem C.3, we get that any correct protocol is long, and the assertion follows. The rest of the sketch is devoted to highlighting the main ideas that go into the selection of the subset $\mathcal{N}'(x)$ in the definition of our measure, as well as to the proofs of Theorem C.2 and Theorem C.3.

2.4.1 Information about x Extracted from Ones in the Transcript

Let us recall our lower bound construction. All the players i have as input a number $x^i \in [2n]$ and they have to compute the set $\mathcal{L}(x) = \{x^i : i \in [n]\}$. We consider a uniform distribution over the inputs x. Suppose that Π is a protocol that correctly computes $\mathcal{L}(x)$ on a 0.9 fraction of inputs x.

We can assume, without loss of generality, that the output of player 1 in Π is determined solely by the transcript of Π. This can be done because the input of player 1 is short ($\mathcal{O}(\log n)$

bits), and thus, can be included as a part of the transcript with only an additive $\mathcal{O}(\log n)$ overhead to the number of rounds in Π.

Since we assume that the protocol Π is correct with probability at least 0.9, we can expect a typical transcript π to be such that, conditioned on the transcript being π, the probability that the protocol is correct is at least 0.9. This holds only in expectation and there is no reason to believe that this holds for any given transcript. However, it does hold for many transcripts π. Throughout the rest of this section, we illustrate how to handle the 1s in any such transcript π by assuming that π is the all ones transcript. In the next section, we describe how to deal with the 0s in π.

Let us now, and for the rest of this section, fix the transcript π to be the all ones transcript and let L be the set output by player 1 when the transcript is π. Let x be the input given to the parties. Under our assumption, conditioned on π, the probability that $\mathcal{L}(x) = L$ is at least 0.9. This means that the expression

$$\frac{\sum_{x:\mathcal{L}(x)=L} \Pr(x,\pi)}{\sum_{x:\mathcal{L}(x)\neq L} \Pr(x,\pi)} = \frac{\sum_{x:\mathcal{L}(x)=L} \Pr(x|\pi)}{\sum_{x:\mathcal{L}(x)\neq L} \Pr(x|\pi)} \tag{1}$$

is at least a constant (assuming that the denominator is non-zero). By the discussion in subsection 2.3, for every x that appears in the numerator, there are $\Theta(n^2)$ neighbors $x' \in \mathcal{N}(x)$ that appear in the denominator.

We wish to focus on only one x and all its neighbors $\mathcal{N}(x)$ and for that we would like to take the sum over x 'out' of the numerator and the denominator in the expression above. As will be explained below, we can in fact bound Equation 1 by

$$n \cdot \frac{\sum_{x:\mathcal{L}(x)=L} \Pr(x,\pi)}{\sum_{x:\mathcal{L}(x)=L} \sum_{x'\in\mathcal{N}(x)} \Pr(x',\pi)}. \tag{2}$$

This will allow us to conclude that there exists an x such that the probability of x given π is comparable to the sum of the probabilities of *all* its neighbors given π. The rest of the proof is divided into two steps. First, we show how to convert Equation 1 to Equation 2. Then, we show how to conclude a lower bound on the length of π from Equation 2

From Equation 1 to Equation 2. The numerators in Equation 1 and Equation 2 are the same, and we solely focus on the denominators. The definition of $\mathcal{N}(x)$ ensures that if $\mathcal{L}(x) = L$ then $\mathcal{L}(x') \neq L$ for all $x' \in \mathcal{N}(x)$ and consequently, every term in the denominator of Equation 2 appears in the denominator of Equation 1. However, it may happen that a term may appear more than once in the denominator of Equation 2 while appearing only once in the denominator of Equation 1.

We claim that a term in the denominator of Equation 1 can appear at most n times in the denominator of Equation 2. Indeed, if a term $\Pr(x',\pi)$ is to appear more than n times, then there are more than n different $x \in \mathcal{N}(x')$ such that $\mathcal{L}(x) = L$. However, since $\mathcal{L}(x') \neq L$, there is at least one mismatch between $\mathcal{L}(x')$ and L. If a neighbor x of x' changes the input

to player i, it must change it in a way to correct this mismatch. This means that, for all i, there is at most one way to change it to an x satisfying $\mathcal{L}(x) = L$ implying a total of at most n ways (one for each $i \in [n]$).

Thus, we are able to conclude that

$$n \cdot \frac{\sum_{x:\mathcal{L}(x)=L} \Pr(x,\pi)}{\sum_{x:\mathcal{L}(x)=L} \sum_{x' \in \mathcal{N}(x)} \Pr(x',\pi)} \geq \frac{\sum_{x:\mathcal{L}(x)=L} \Pr(x,\pi)}{\sum_{x:\mathcal{L}(x)\neq L} \Pr(x,\pi)}.$$

We do not know if losing a factor of n is necessary while going from Equation 1 to Equation 2. However, the rest of our analysis is strong enough to deal with this loss.

From Equation 2 to a Lower Bound. For any input x, the neighbors $x' \in \mathcal{N}(x)$ can be divided into $\Theta(n)$ categories based on the player i whose input is changed. Let us denote the subset of neighbors that differ in the input of player i using $\mathcal{N}^i(x)$.

From Equation 2, we can conclude that there exists an x such that given π, the probability of x is at least $\Omega\left(\frac{1}{n}\right)$ times the probability of $\mathcal{N}(x)$. This means that there exists an i such that given π, the probability of x is at least $\Omega(1)$ times the probability of $\mathcal{N}^i(x)$. It turns out that the hardest case to analyze is when this holds for *all* the possible values of i. We assume this for the rest of this subsection.

Prior to executing the protocol Π, since the inputs were sampled uniformly, the probability of x is $\mathcal{O}\left(\frac{1}{n}\right)$ times the probability of $\mathcal{N}^i(x)$. Conditioning on π changes $\mathcal{O}\left(\frac{1}{n}\right)$ to $\Omega(1)$. What is it about the all ones transcript π that makes x so much more likely than its neighbors $x' \in \mathcal{N}^i(x)$?

The inputs x' differ from x only in the input of player i and, consequently, the only way π can differentiate between x and x' is if all the players other than i beep 0. Furthermore, since π is the all ones transcript, the only way x can be made much *more* likely relative to x' is when there are many rounds where player i was the only player beeping a 1. In this case, since x' has a different input for player i, player i may no longer beep 1 in the corresponding round, and thus may lower the probability of receiving a 1 by a constant factor (as the noise rate is constant).

Since every such round increases the probability of x relative to x' by a constant factor, and we need to increase by a factor of n, there should be $\Omega(\log n)$ rounds where player i is the only player beeping 1. Since there are $\Theta(n)$ possible values for i, Π must have $\Omega(n \log n)$ rounds.

2.4.2 Information about x Extracted from Zeros in the Transcript

The previous subsection was dedicated to handling the ones in π, and we assumed that π is the transcript that has all 1s. Are things a lot different when π is allowed to have 0s?

When the transcript π is allowed to have zeros, x can be made more likely relative to its neighbors $x' \in \mathcal{N}^i(x)$ in another way: consider rounds m such that $\pi_m = 0$ and none of the players beeped a 1 (when the input was x). Since the input to player i is different in

x', player i may beep 1 in round m, lowering the probability of receiving a 0 by a constant factor. Superficially, this may look the same as before. However, something very important has changed.

Earlier, the relevant rounds m were such that when the input was x, only player i was beeping in round m. This means that, in round m, the relative probability of x and x' is affected *only* for neighbors $x' \in \mathcal{N}^i(x)$. As explained above, the number of rounds required to sufficiently increase the probability of x versus the probability of neighbors $x' \in \mathcal{N}^i(x)$, for a given i, is $\Omega(\log n)$, and the hard case is when the probability of x increases versus any $x' \in \mathcal{N}(x)$ (*i.e.*, for any i). Observe that, earlier, *the relevant rounds for different i were disjoint* and we were able to conclude that the total number of rounds is $\Omega(n \log n)$.

However, when the transcript π is allowed to have zeros, even rounds m where all the players beep 0 are relevant and these rounds may affect the relative probability of x and *any neighbor* $x' \in \mathcal{N}(x)$ (independently of i) as changing any player's input may change the 0 to a 1. It is not clear how to show that the rounds for different i are 'sufficiently disjoint' for us to get the factor of n we need for the lower bound.

Since this line of attack does not give us a lot of traction, we exploit a different weakness of the 0s in the transcript.

2.4.3 Handling the Zeros in the Transcript

In subsection 2.1, we argued that the zeros of the transcript are easier to verify than the ones. This is because if a 0 was supposed to be a 1, the player who was beeping the 1 knows that, and can immediately raise an error flag. If no player raised an error flag, the players can be confident that the bit was actually a 0. Thus, a mechanism can be built on the lines of, say, [EKS18] that ensures that all the 0s received in Π are noise-free without increasing the number of rounds in Π by a lot. For the rest of this sketch, we work under a one-sided error assumption, where when the parties receive the bit 0, they know that it is actually 0.

Since the players know the protocol Π and the current transcript π, the knowledge that the bit was actually a 0 allows them to rule out certain input tuples. The subset of inputs that is ruled out is precisely those inputs for which the parties would have beeped 1. Define, for a transcript π, the set $S^i(\pi)$ to be the subset of inputs to player i that are *not* ruled out in this way. In other words, if a player i has an input from the set $S^i(\pi)$, then, for all rounds m such that $\pi_m = 0$, player i beeps 0 in round m.

Observe that for a given transcript π, if we solely focus on the neighbors x' of an input x such that $x'^i \in S^i(\pi)$ for all players i, then in a round m where all players beep 0 when the input is x, then all players beep 0 even when the input is x'. Thus, the problem described at the beginning of subsubsection 2.4.2 does not arise. This allows us to work as described in subsubsection 2.4.1.

There is one last piece in this puzzle which is the size of the sets $S^i(\pi)$. Recall that we got the $\Omega(\log n)$ factor in the complexity because the size of the sets $\mathcal{N}^i(x)$ was n. Now, we are working with a subset of the neighbors x' in $\mathcal{N}^i(x)$ for which $x'^i \in S^i(\pi)$. Does this

mean we get a worse lower bound? The answer is no, and the reason is that we can ensure that for most i, the sets $S^i(\pi)$ are at least polynomially large ($\geq \sqrt{n}$, say). This is due to a careful information theoretic argument which is as follows:

Initially, the input to our problem has entropy $\Theta(n \log n)$. Since our protocol is short, its transcript cannot give us more that $c \cdot n \log n$ bits of information about the input, for some small c. Thus, even after conditioning on the transcript π, the input distribution should have entropy $\Theta(n \log n)$. We know that inputs not in $S^i(\pi)$ are no longer possible given the transcript. Thus, the sets $S^i(\pi)$ cannot all be very small (say, sub-polynomial). As long as the sets $S^i(\pi)$ are polynomially large, we have an $\Omega(\log n)$ overhead term, as desired.

For the formal definitions and the proof, please see the appendices below.

References

[AAB+11] Yehuda Afek, Noga Alon, Omer Barad, Eran Hornstein, Naama Barkai, and Ziv Bar-Joseph. A biological solution to a fundamental distributed computing problem. *Science*, 331(6014):183–185, 2011. 1

[AAB+13] Yehuda Afek, Noga Alon, Ziv Bar-Joseph, Alejandro Cornejo, Bernhard Haeupler, and Fabian Kuhn. Beeping a maximal independent set. *Distributed Computing*, 26(4):195–208, 2013. 1

[AABJ11] Yehuda Afek, Noga Alon, and Ziv Bar-Joseph. Beeping an MIS. *Manuscript*, 2011. 1

[ABE+16] Noga Alon, Mark Braverman, Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Reliable communication over highly connected noisy networks. In *Symposium on Principles of Distributed Computing (DISC)*, pages 165–173. ACM, 2016. 4

[BE14] Mark Braverman and Klim Efremenko. List and unique coding for interactive communication in the presence of adversarial noise. In *Foundations of Computer Science (FOCS)*, pages 236–245, 2014. 3

[BEGH16] Mark Braverman, Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Constant-rate coding for multiparty interactive communication is impossible. In *Symposium on Theory of Computing (STOC)*, pages 999–1010. ACM, 2016. 4

[BGMO17] Mark Braverman, Ran Gelles, Jieming Mao, and Rafail Ostrovsky. Coding for interactive communication correcting insertions and deletions. *IEEE Transactions on Information Theory*, 63(10):6256–6270, 2017. 3

[BKK+16] Philipp Brandes, Marcin Kardas, Marek Klonowski, Dominik Pajak, and Roger Wattenhofer. Approximating the size of a radio network in beeping model. In *Structural Information and Communication Complexity (SIROCCO)*, pages 358–373, 2016. 1

[BKN14] Zvika Brakerski, Yael Tauman Kalai, and Moni Naor. Fast interactive coding against adversarial noise. *Journal of the ACM (JACM)*, 61(6):35, 2014. 3

[BO15] Mark Braverman and Rotem Oshman. The communication complexity of number-in-hand set disjointness with no promise. *Manuscript*, 2015. 26

[BR11] Mark Braverman and Anup Rao. Towards coding for maximum errors in interactive communication. In *Symposium on Theory of computing (STOC)*, pages 159–166. ACM, 2011. 3

[Bra12] Mark Braverman. Towards deterministic tree code constructions. In *Innovations in Theoretical Computer Science (ITCS)*, pages 161–167. ACM, 2012. 3

[CGH18] Keren Censor-Hillel, Ran Gelles, and Bernhard Haeupler. Making asynchronous distributed computations robust to channel noise. In *Innovations in Theoretical Computer Science Conference (ITCS)*, pages 50:1–50:20, 2018. 4

[CGK+14] Keren Censor-Hillel, Seth Gilbert, Fabian Kuhn, Nancy A. Lynch, and Calvin C. Newport. Structuring unreliable radio networks. *Distributed Computing*, 27(1):1–19, 2014. 4

[CHHZ17] Keren Censor-Hillel, Bernhard Haeupler, D. Ellis Hershkowitz, and Goran Zuzic. Broadcasting in noisy radio networks. In *Symposium on Principles of Distributed Computing (PODC)*, pages 33–42, 2017. 4

[CHHZ18] Keren Censor-Hillel, Bernhard Haeupler, D. Ellis Hershkowitz, and Goran Zuzic. Erasure correction for noisy radio networks. *CoRR*, abs/1805.04165, 2018. 4

[CK10] Alejandro Cornejo and Fabian Kuhn. Deploying wireless networks with beeps. In *International Symposium on Distributed Computing (DISC)*, pages 148–162, 2010. 1

[DBB18] Fabien Dufoulon, Janna Burman, and Joffroy Beauquier. Beeping a deterministic time-optimal leader election. In *International Symposium on Distributed Computing (DISC)*, pages 20:1–20:17, 2018. 1

[EGH16] Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Maximal noise in interactive communication over erasure channels and channels with feedback. *IEEE Transactions on Information Theory*, 62(8):4575–4588, 2016. 3

[EKS18] Klim Efremenko, Gillat Kol, and Raghuvansh Saxena. Interactive coding over the noisy broadcast channel. In *Symposium on Theory of Computing (STOC)*, pages 507–520. ACM, 2018. 3, 4, 5, 10, 28

[EKS19] Klim Efremenko, Gillat Kol, and Raghuvansh Saxena. Radio network coding requires logarithmic overhead. *Manuscript*, 2019. 4

[FK00] Uriel Feige and Joe Kilian. Finding OR in a noisy broadcast network. *Information Processing Letters*, 73(1-2):69–75, 2000. 4

[FSW14] Klaus-Tycho Förster, Jochen Seidel, and Roger Wattenhofer. Deterministic leader election in multi-hop beeping networks. In *International Symposium on Distributed Computing (DISC)*, pages 212–226, 2014. 1

[Gal88] Robert G. Gallager. Finding parity in a simple broadcast network. *IEEE Transactions on Information Theory*, 34(2):176–180, 1988. 3, 4

[Gam87] Abbas El Gamal. Open problems presented at the 1984 workshop on specific problems in communication and computation sponsored by bell communication research. *Appeared in "Open Problems in Communication and Computation", by Thomas M. Cover and B. Gopinath (editors). Springer-Verlag*, 1987. 4

[Gel17] Ran Gelles. Coding for interactive communication: A survey. *Foundations and Trends® in Theoretical Computer Science*, 13(1–2):1–157, 2017. 5

[GGKN09] Seth Gilbert, Rachid Guerraoui, Dariusz R. Kowalski, and Calvin C. Newport. Interference-resilient information exchange. In *International Conference on Computer Communications (INFOCOM)*, pages 2249–2257, 2009. 4

[GH14] Ran Gelles and Bernhard 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*, pages 1296–1311. SIAM, 2014. 3

[GHK+16] Ran Gelles, Bernhard Haeupler, Gillat Kol, Noga Ron-Zewi, and Avi Wigderson. Towards optimal deterministic coding for interactive communication. In *Symposium on Discrete Algorithms (SODA)*, pages 1922–1936. Society for Industrial and Applied Mathematics, 2016. 3

[GHLN12] Mohsen Ghaffari, Bernhard Haeupler, Nancy A. Lynch, and Calvin C. Newport. Bounds on contention management in radio networks. In *International Symposium on Distributed Computing (DISC)*, pages 223–237, 2012. 4

[GHM18] Ofer Grossman, Bernhard Haeupler, and Sidhanth Mohanty. Algorithms for noisy broadcast under erasures. *CoRR*, abs/1808.00838, 2018. 4

[GK17] Ran Gelles and Yael Tauman Kalai. Constant-rate interactive coding is impossible, even in constant-degree networks. In *Innovations in Theoretical Computer Science Conference (ITCS)*, pages 21:1–21:13, 2017. 4

[GKR19] Ran Gelles, Yael Tauman Kalai, and Govind Ramnarayan. Efficient multiparty interactive coding for insertions, deletions and substitutions. *CoRR*, abs/1901.09863, 2019. 4

[GKS08] Navin Goyal, Guy Kindler, and Michael Saks. Lower bounds for the noisy broadcast problem. *SIAM Journal on Computing*, 37(6):1806–1841, 2008. 4

[GLN13] Mohsen Ghaffari, Nancy A. Lynch, and Calvin C. Newport. The cost of radio network broadcast for different models of unreliable links. In *Principles of Distributed Computing (PODC)*, pages 345–354, 2013. 4

[GMS11] Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient and explicit coding for interactive communication. In *Foundations of Computer Science (FOCS)*, pages 768–777. IEEE, 2011. 3

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

[GN15] Seth Gilbert and Calvin C. Newport. The computational power of beeps. In *International Symposium on Distributed Computing (DISC)*, pages 31–46, 2015. 1

[HL17] Stephan Holzer and Nancy A. Lynch. Brief announcement: Beeping a maximal independent set fast. In *International Symposium on Distributed Computing (DISC)*, 2017. 1

[HM13] Bojun Huang and Thomas Moscibroda. Conflict resolution and membership problem in beeping channels. In *International Symposium on Distributed Computing (DISC)*, pages 314–328, 2013. 1

[HS16] William M. Hoza and Leonard J. Schulman. The adversarial noise threshold for distributed protocols. In *Symposium on Discrete Algorithms (SODA)*, pages 240–258, 2016. 4

[JKL15] Abhishek Jain, Yael Tauman Kalai, and Allison Bishop Lewko. Interactive coding for multiparty protocols. In *Innovations in Theoretical Computer Science (ITCS)*, pages 1–10, 2015. 4

[KKP01] Evangelos Kranakis, Danny Krizanc, and Andrzej Pelc. Fault-tolerant broadcasting in radio networks. *Journal of Algorithms*, 39(1):47–67, 2001. 4

[KLN+10] Fabian Kuhn, Nancy A. Lynch, Calvin C. Newport, Rotem Oshman, and Andréa W. Richa. Broadcasting in unreliable radio networks. In *Symposium on Principles of Distributed Computing (PODC)*, pages 336–345, 2010. 4

[KM05] Eyal Kushilevitz and Yishay Mansour. Computation in noisy radio networks. *SIAM J. Discrete Math.*, 19(1):96–108, 2005. 4

[KPP10] Evangelos Kranakis, Michel Paquette, and Andrzej Pelc. Communication in random geometric radio networks with positively correlated random faults. *Ad Hoc & Sensor Wireless Networks*, 9(1-2):23–52, 2010. 4

[MRZ15] Yves Métivier, John Michael Robson, and Akka Zemmari. On distributed computing with beeps. *CoRR*, abs/1507.02721, 2015. 1

[MS14] Cristopher Moore and Leonard J Schulman. Tree codes and a conjecture on exponential sums. In *Innovations in theoretical computer science (ITCS)*, pages 145–154. ACM, 2014. 3

[NB15] Saket Navlakha and Ziv Bar-Joseph. Distributed information processing in biological and computational systems. *Communications of the ACM*, 58(1):94–102, 2015. 1

[New04] Ilan Newman. Computing in fault tolerance broadcast networks. In *Computational Complexity Conference (CCC)*, pages 113–122, 2004. 4

[RS94] Sridhar Rajagopalan and Leonard J. Schulman. A coding theorem for distributed computation. In *Symposium on the Theory of Computing (STOC)*, pages 790–799, 1994. 4

[Sch92] Leonard J Schulman. Communication on noisy channels: A coding theorem for computation. In *Foundations of Computer Science (FOCS)*, pages 724–733. IEEE, 1992. 3, 5

[Sch93] Leonard J Schulman. Deterministic coding for interactive communication. In *Symposium on Theory of computing (STOC)*, pages 747–756. ACM, 1993. 3

[Sch96] Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996. 3

[SJX13] Alex Scott, Peter Jeavons, and Lei Xu. Feedback from nature: an optimal distributed algorithm for maximal independent set selection. In *Symposium on Principles of Distributed Computing (PODC)*, pages 147–156, 2013. 1

Appendix

A The Beeping Model and The Problem

Throughout this paper, we will work with the n-party beeping model described below. We will use $\mathcal{X}^i$ to denote the inputs of party i and $\mathcal{Y}^i$ to denote the output space.

A.1 The Model

A.1.1 Noisy Beeping Definitions

In the section, we define the n-party correlated noise beeping models assumed by this paper, by defining communication protocols in these models. We first define the noisy beeping model and the one-sided noisy beeping model for a noise parameter ϵ. The noiseless beeping model and the noiseless one-sided beeping model are obtained from the noisy versions by setting $\epsilon = 0$.

Let $\epsilon \geq 0$. A *(deterministic) protocol* Π *over the n-party ϵ-noisy beeping model* is defined by a tuple $\left(T, \{f_m^i\}_{i\in[n],m\in[T]}, \{g^i\}_{i\in[n]}\right)$. Here, $T > 0$ is a parameter that denotes the length of Π, f_m^i are functions of the type $f_m^i : \mathcal{X}^i \times \{0,1\}^{m-1} \to \{0,1\}$ (broadcast functions), and g^i are functions of the type $g^i : \mathcal{X}^i \times \{0,1\}^T \to \mathcal{Y}^i$ (output functions).

The protocol Π executes as follows. Suppose that party i has input $x^i \in \mathcal{X}^i$. Before round m, the parties share a transcript $\pi_{<m}$. Define $\pi_{<1} = \varepsilon$, the empty string. In round $m \in [T]$, we say that party i beeps the bit $b^i = f_m^i(x^i, \pi_{<m})$. Define the output of the channel for round m to be

$$\pi_m = N_\epsilon \oplus \bigvee_{i\in[n]} b^i,$$

where, N_ϵ is an ϵ-noisy bit that is independent of all other randomness in the execution. That is, $N_\epsilon = 0$ with probability $1-\epsilon$, and $N_\epsilon = 1$ with probability ϵ. The parties append π_m to $\pi_{<m}$ and continue the execution of the protocol. After T rounds, party i outputs $g^i(x^i, \pi_{\leq T})$.

A (deterministic) protocol Π over the n-party *one-sided* ϵ-noisy beeping model is defined similarly to the ϵ-noisy beeping model, with one difference: In round $m \in [T]$, if $\bigvee_{i\in[n]} b^i = 1$ then $\pi_m = 1$ (no noise is introduced). Otherwise, if $\bigvee_{i\in[n]} b^i = 0$, then $\pi_m = N_\epsilon \oplus \bigvee_{i\in[n]} b^i$, as before.

A *randomized* protocol over the n-party ϵ-noisy beeping model is a distribution over deterministic protocols in the n-party ϵ-noisy beeping model. A *randomized* protocol over the one-sided n-party ϵ-noisy beeping model is a distribution over deterministic protocols in the one-side n-party ϵ-noisy beeping model.

A.1.2 One-Sided vs Two-Sided Noise

We next claim that every protocol Π over the n-party $\frac{1}{4}$-noisy beeping model can be simulated over the one-sided n-party $\frac{1}{3}$-noisy beeping model where the parties share a random string: parties run Π over the one sided n-party $\frac{1}{3}$-noisy beeping model round-by-round. If the parties receive the bit 1, they flip it to 0 with probability $\frac{1}{4}$ using the shared randomness. Now, the probability of the updated output being 0 if one of the parties beeps a 1 is $\frac{1}{4}$, as the one-sided noise never flips a 1 to a 0, but the parties will flip it by themselves with probability $\frac{1}{4}$. The probability of the updated output being 1 if all parties beeps 0 is also $\frac{1}{4}$: with probability $\frac{1}{3}$, the one-sided noise flips the 0 to a 1, and then with probability $\frac{3}{4}$ the parties do not flip the 1 back to a 0. Since these events are independent, we get that a probability of the updated output being 1 if all parties beeps 0 is $\frac{1}{3} \cdot \frac{3}{4} = \frac{1}{4}$.

The above claim shows that a lower bound on the length of a protocol solving a communication task over the n-party one-sided $\frac{1}{3}$-noisy beeping model with shared randomness, implies a similar lower bound for the n-party $\frac{1}{4}$-noisy beeping model.

Another way to see that the one sided error model gives the parties more power than the two sided error model is two consider the two sided error model but with the presence of an adversary that can 'correct' any of the bits flipped by the channel (but cannot introduce new errors). The presence of such an adversary essentially prohibits the parties from using a protocol that relies on the noise being exactly what it is. Since adversary can correct every time the channel flips a 1 to a 0, the two sided error model becomes a one sided one.

A.2 The Communication Task

The Communication Problem $\mathsf{InputSet}_n$. For $n > 0$, we consider the $\mathsf{InputSet}_n$ communication task, which is described next. In the task $\mathsf{InputSet}_n$, all of the n parties get as input a number in $[2n]$, *i.e.*, party i gets input $x^i \in [2n]$. The numbers x^i are chosen uniformly at random, and independently for all the n parties. We will use $x = x^1 x^2 \cdots x^n$ to denote the vector of all inputs. The parties need to compute the set $\mathcal{L}(x) = \{x^i \mid i \in [n]\}$. We stress that party i may not know x^j for any $j \neq i \in [n]$. Rather, if $y \in \mathcal{L}(x)$, then party i knows that *some* party j received y as input.

Noiseless Protocol for $\mathsf{InputSet}_n$. The task $\mathsf{InputSet}_n$ has a $2n$ length deterministic protocol in the noiseless beeping model. In a round $m \in [T]$, party i beeps 1 if and only if $x^i = m$. Observe that $\pi_m = 1 \iff m \in \mathcal{L}(x)$. Thus, the parties can output $\mathcal{L}(x)$ at the end of the protocol.

The rest of this paper is devoted to showing that for a constant $\epsilon > 0$, any (even randomized) protocol for $\mathsf{InputSet}_n$ in the ϵ-noisy beeping model must consist of at least $\frac{n \log n}{1000}$ rounds in the worst-case. We note that since $\mathsf{InputSet}_n$ is a distributional communication task, the worst-case length of a protocol (over the selection of inputs, channel noise, and the randomness used by the players) is only a constant factor away form its expected length (for

a related error parameter), as the protocol can always be truncated to twice its expected length, incurring only a constant blowup to the error.

B Technical Preliminaries

B.1 Notation

All logarithms in this paper are to the base 2. We denote random variables using capital letters and their realizations using the corresponding lower case letters. For example, a random variable X may take the value x. We often write $\Pr(\cdot \mid X = x)$, $\Pr(X = x)$, $\mathbb{E}(\cdot \mid X = x)$, *etc.*, as $\Pr(\cdot \mid x)$, $\Pr(x)$, and $\mathbb{E}(\cdot \mid x)$. We use the notation $\sum_{x \in S} \Pr(x) = \Pr(S)$. Finally, the support of X, denoted $\mathrm{supp}(X)$, is the set $\{x \mid \Pr(x) > 0\}$ and the concatenation of two strings s_1 and s_2 is denoted using $s_1 s_2$.

B.2 Information Theory

Definition B.1 (Entropy)**.** *The (binary) entropy of a discrete random variable X is defined as*

$$\mathbb{H}(X) = \sum_{x \in \mathrm{supp}(X)} \Pr(x) \log \frac{1}{\Pr(x)} = \mathop{\mathbb{E}}_{x \sim X} \left[\log \frac{1}{\Pr(x)} \right].$$

Definition B.2 (Conditional Entropy)**.** *The entropy of a discrete random variable X given another random variable Y is defined as*

$$\mathbb{H}(X \mid Y) = \mathop{\mathbb{E}}_{y \sim Y} \left[\mathbb{H}(X \mid Y = y) \right].$$

Definition B.3 (Mutual Information)**.** *The mutual information between two discrete random variables X and Y is defined as*

$$\mathbb{I}(X : Y) = \mathbb{H}(X) - \mathbb{H}(X \mid Y) = \mathbb{H}(Y) - \mathbb{H}(Y \mid X).$$

We can also define conditional mutual information as

$$\mathbb{I}(X : Y \mid Z) = \mathbb{H}(X \mid Z) - \mathbb{H}(X \mid YZ) = \mathbb{H}(Y \mid Z) - \mathbb{H}(Y \mid XZ).$$

Fact B.4. *If the random variable X takes values in the set Ω, it holds that*

$$0 \leq \mathbb{H}(X) \leq \log|\Omega|.$$

Fact B.5 (Conditioning reduces entropy)**.** *We have $0 \leq \mathbb{I}(X : Y) \leq \mathbb{H}(X)$.*

Fact B.6 (Subadditivity of entropy)**.** *We*[4] *have $\mathbb{H}(XY) = \mathbb{H}(X) + \mathbb{H}(Y \mid X) \leq \mathbb{H}(X) + \mathbb{H}(Y)$. Equality holds if X and Y are independent.*

[4] We use $\mathbb{H}(XY)$ to denote the joint entropy of the tuple (X, Y).

B.3 Some Lemmas

Lemma B.7. *Let $k > 0$ and a_i, b_i be positive numbers for $i \in [k]$. It holds that*

$$\frac{\left(\sum_{i\in[k]} a_i\right)^2}{\sum_{i\in[k]} b_i} \leq \sum_{i\in[k]} \frac{a_i^2}{b_i}$$

Proof. Applying the Cauchy-Schwarz inequality to the sequences $\frac{a_i}{\sqrt{b_i}}$ and $\sqrt{b_i}$, we get

$$\left(\sum_{i\in[k]} a_i\right)^2 \leq \left(\sum_{i\in[k]} \frac{a_i^2}{b_i}\right)\left(\sum_{i\in[k]} b_i\right).$$

□

Lemma B.8. *Let $k > 0$ and $\{X_i\}_{i=1}^k$ be mutually independent random variables uniformly distributed over a finite set S. Define the random set $I = \{i \in [k] \mid X_i \neq X_j, \forall j \neq i\}$. If $k < |S|$,*

$$\Pr\left(|I| \leq \frac{k}{3}\right) \leq \frac{3}{2} \cdot \left(1 - \mathrm{e}^{-\frac{k}{|S|}}\right).$$

Proof. Let $\alpha = \Pr\left(|I| \leq \frac{k}{3}\right)$. We have

$$\mathbb{E}[|I|] \leq \alpha \cdot \frac{k}{3} + (1-\alpha)k = k\left(1 - \frac{2\alpha}{3}\right).$$

Define Y_i, for $i \in [k]$, to be indicator random variable that is 1 if and only if $i \in I$. By linearity of expectation, we have

$$\mathbb{E}[|I|] = \sum_{i\in[k]} \mathbb{E}[Y_i] = k\,\mathbb{E}[Y_1],$$

where the last step follows as the variables Y_i are identically distributed. By direct calculation,

$$\mathbb{E}[|I|] = k\,\mathbb{E}[Y_1] = k\left(1 - \frac{1}{|S|}\right)^{k-1} \geq k\mathrm{e}^{-\frac{k-1}{|S|-1}} \geq k\mathrm{e}^{-\frac{k}{|S|}},$$

where we use $\mathrm{e}^{-1} \leq \left(1 - \frac{1}{n}\right)^{n-1}$, for all $n > 1$. Combining the two bounds for $\mathbb{E}[Y_1]$, we have

$$\alpha \leq \frac{3}{2} \cdot \left(1 - \mathrm{e}^{-\frac{k}{|S|}}\right).$$

□

C Logarithmic Overhead Lower Bound (Proof of Theorem 1.1)

For the rest of this text, we fix $n > 0$ and a protocol Π for the task $\mathsf{InputSet} = \mathsf{InputSet}_n$ in the ϵ-noisy beeping model that has length $T < \frac{n \log n}{1000}$. For ease of exposition, we assume that $\epsilon = \frac{1}{3}$. We assume that the inputs to the n parties are drawn uniformly and independently from $[2n]$. Since the input of the task $\mathsf{InputSet}_n$ is randomized, we can assume, without loss of generality, that the protocol Π is deterministic. Formally, we prove the following theorem, and Theorem 1.1 follows.

Theorem C.1. *Any deterministic protocol that computes* $\mathsf{InputSet}_n$ *with error probability* $\epsilon < 1/4$ *over the* n*-party one-sided* ϵ*-noisy beeping model (over uniformly sampled inputs), requires* $\Omega(n \log n)$ *rounds.*

C.1 Notation

We let X^i be the random variable that represents the input x^i of party i. Recall that the variables X^i are independent and uniformly distributed over the set $[2n]$. We will use $X = X^1 X^2 \cdots X^n$ to denote the vector of all the parties' inputs. Recall that realization of random variables are denoted using the corresponding lower case letters. For $y \in [2n]$, we use $x^{i=y}$ to denote the input which is the same as x except at the coordinate i, where it is y.

We reserve m to index the rounds of Π. For $m \in [T]$, we use Π_m to denote the random variable representing the bit received by the players in round m of Π. The randomness in Π_m is the randomness in the parties' inputs and the noise added by the beeping model. The notation $\Pi_{\leq m}$ will mean $\Pi_1 \Pi_2 \cdots \Pi_m$. We often omit the subscript when $m = T$, *e.g.*, Π denotes $\Pi_{\leq T}$.

C.2 Our Framework

In our proof, we make two simplifying assumptions that are without loss of generality. Firstly, we assume that when any of the parties beeps 1, then this is received correctly by all the parties. In other words, the noise can only change a 0 to a 1. This assumption also means that in any round where the parties *receive* a 0, they can be sure that all parties beeped 0.

The second assumption that we make is that the output of player 1 is determined by the transcript π of the protocol, and does not require knowing the input of player 1 (*i.e.*, $g^1(x^1, \pi)$ is only a function of π). This does not compromise generality as the input of player 1 is small and can be written using $\mathcal{O}(\log n)$ bits. Thus, for any (noiseless) protocol Π that solves $\mathsf{InputSet}$, we can construct a (noiseless) protocol Π', that is at most $\mathcal{O}(\log n)$ longer than Π, in which player 1 beeps its input in the first rounds (no other player beeps during these rounds), and then the players execute Π. The transcript of Π' contains the input of player 1, thus player 1's output is only a function of the transcript.

Correct executions. Define the set $\mathcal{C}$ of correct executions as the set of tuples (x, π) such that player 1 outputs $\mathcal{L}(x)$ when the transcript is π. This is well defined due to our assumption that the output of player 1 is determined by π. Our proof works by upper bounding $\Pr(\mathcal{C})$.

Feasible sets. Let $m \in [T]$. For a realization $\pi_{\leq m} = \pi_1 \pi_2 \cdots \pi_m$ of $\Pi_{\leq m}$, let $J = \{j \in [m] : \pi_j = 0\}$ be the set of coordinates of $\pi_{\leq m}$ that are 0. Since we assume that the noise in our channel can only change a 0 to a 1, $\pi_j = 0$ implies that all the parties beeped 0 in round j. For $j \in J$, let $S^i_j = \{y : f^i_j(y, \pi_{<j}) = 0\}$[5] be the set of all inputs of party i such that party i beeps 0 in round j if the transcript received in the first $j-1$ rounds is $\pi_{<j}$. Define the feasible set of party i and a transcript $\pi_{\leq m}$ as

$$S^i(\pi_{\leq m}) = \bigcap_{j \in J} S^i_j.$$

Good players. Let $x = x^1 x^2 \cdots x^n$ be an input to the n parties and let π be a transcript. Recall the feasible sets $S^i(\pi)$ defined above. Let $G_1(x) = \{i \in [n] \mid x^i \neq x^j, \forall j \neq i\}$ be the set of all parties with unique inputs. Let $G_2(\pi) = \{i \in [n] \mid |S^i(\pi)| > \sqrt{n}\}$ be the set of parties with more than $\sqrt{n}$ feasible inputs given the transcript π. The set of good players G for the input x and transcript π is defined as

$$G(x, \pi) = G_1(x) \cap G_2(\pi).$$

Define the event $\mathcal{G}$ (over the selection of inputs and the channel noise) as

$$\mathcal{G} \equiv |G(x, \pi)| \geq \frac{n}{4}.$$

A progress measure. Let $i \in [n]$. For any input x and any π, let $G(x, \pi)$ and $S^i(\pi)$ be as defined above. Define:

$$Z(x, \pi) = \sum_{i \in G(x,\pi)} \mathop{\mathbb{E}}_{y \sim S^i(\pi)} \left[\Pr(x^{i=y}, \pi)\right].$$

$$\zeta(x, \pi) = \begin{cases} 0 & , \Pr(x, \pi) = 0 \\ \frac{\Pr(x,\pi)}{Z(x,\pi)} & , \Pr(x, \pi) > 0 \end{cases}.$$

The probabilities in the above definitions are over the selection of inputs and the channel noise. We note that ζ is well defined as if $\Pr(x, \pi) > 0$ then $Z(x, \pi) \neq 0$. The reason is that since x can happen with π and since $y \in S^i(\pi)$, it holds that $x^{i=y}$ can also happen with π.

[5] Recall from Appendix A the definition of f^i_j.

C.3 The Proof of Theorem 1.1

C.3.1 Short Protocols Imply Small ζ

We prove Theorem 1.1 in two steps. First, we show that if T is short, then $\zeta(x,\pi)$ is small.

Theorem C.2. *For all x,π such that $\mathcal{G}$ happens and $\Pr(x,\pi)>0$,*

$$\zeta(x,\pi) \le \frac{4}{n}\cdot 3^{\frac{4T}{n}}.$$

Proof. Fix $\pi=\pi_1\pi_2\cdots\pi_T$ and $x=x^1x^2\cdots x^n$ be such that $\mathcal{G}$ happens and $\Pr(x,\pi)>0$. By the chain rule, we have

$$\Pr(x,\pi)=\Pr(x)\prod_{m\in[T]}\Pr(\pi_m \mid x,\pi_{<m}).$$

For $m\in[T]$, let $B_m=B_m(x,\pi)$ be the set of all players that beeped 1 in round m. Note that one can indeed be retrieve B_m given x and π, as each party decides what to beep in the next round based on its input and the shared transcript of the prior rounds. Also note that B_m may be the empty set in case all the players beeped 0. We partition the rounds into disjoint sets:

$$A_0=\{m\in[T]\mid \pi_m=0\} \qquad A_0'=\{m\in[T]\mid \pi_m=1\wedge B_m=\emptyset\}.$$

$$A_i=\{m\in[T]\mid B_m=\{i\}\} \qquad A_{n+1}=[T]\setminus\left(A_0\cup A_0'\cup\bigcup_{i\in[n]}A_i\right).$$

Recall our assumption that a 1 is not affected by noise. Since we assume that $\Pr(x,\pi)>0$, $B_m\neq\emptyset \implies \pi_m=1$. On the other hand, we have $B_m=\emptyset$ means that $\Pr(\pi_m=0\mid x,\pi_{<m})=\frac{2}{3}$ and $\Pr(\pi_m=1\mid x,\pi_{<m})=\frac{1}{3}$ (recall that ϵ is fixed to $1/3$). Direct calculation gives:

$$\Pr(x,\pi)=\Pr(x)\left(\frac{2}{3}\right)^{|A_0|}\left(\frac{1}{3}\right)^{|A_0'|}.$$

We now focus on lower bounding $\Pr(x^{i=y},\pi)$ for some $i\in G(x,\pi), y\in S^i(\pi)$. Since $y\in S^i(\pi)$, for any round $m\in A_0$, all the parties beep 0 in round m even when the input is $x^{i=y}$. Thus, these rounds behave exactly as before and the terms carry over.

We claim that for any round $m\in A_0'$, it holds that $\Pr(\pi_m=1\mid x^{i=y},\pi_{<m})\ge\frac{1}{3}$. Recall that for such rounds $B_m=\emptyset$, meaning that given the input x, all parties beeped 0 in round m. Given the input $x^{i=y}$, it is possible that all parties still beep 0, in which case $\Pr(\pi_m=1\mid x^{i=y},\pi_{<m})=\frac{1}{3}$, or at least one party decides to beep 1 in this round, implying $\Pr(\pi_m=1\mid x^{i=y},\pi_{<m})=1$.

Among the other rounds, the only ones where the probabilities may be affected are the rounds in A_i, as only party i has a different input in $x^{i=y}$. In A_i, party i is the only one beeping a 1. However, even if party i decides to beep a 0 given its new input y, then due

to the noise, these rounds will still produce the same transcript with probability $\frac{1}{3}$. We can thus lower bound $\Pr(x^{i=y}, \pi)$ as follows:

$$\Pr(x^{i=y}, \pi) \geq \Pr(x^{i=y}) \left(\frac{2}{3}\right)^{|A_0|} \left(\frac{1}{3}\right)^{|A_0'|} \left(\frac{1}{3}\right)^{|A_i|}.$$

Since the distribution over inputs is uniform, we have

$$\begin{aligned}
\zeta(x, \pi) &= \frac{\Pr(x, \pi)}{\sum_{i \in G(x,\pi)} \mathbb{E}_{y \sim S^i(\pi)} \left[\Pr(x^{i=y}, \pi)\right]} \\
&\leq \frac{1}{\sum_{i \in G(x,\pi)} 3^{-|A_i|}} \\
&\leq \frac{1}{|G(x,\pi)|} \cdot 3^{\frac{\sum_{i \in G(x,\pi)} |A_i|}{|G(x,\pi)|}} && \text{(Convexity of } f(x) = 3^{-x}\text{)} \\
&\leq \frac{4}{n} \cdot 3^{\frac{4T}{n}} && \text{(Since } \mathcal{G} \text{ occurs and by the disjointness of } A_i\text{)}
\end{aligned}$$

□

C.3.2 Correct (and Short) Protocols Have Large ζ

We finish by showing that the correctness of Π implies that $\zeta(x, \pi)$ cannot be that small.

Theorem C.3. *If* $\Pr(\mathcal{C}) \geq \frac{2}{3} + n^{-\frac{1}{8}}$ *and* $T \leq \frac{n \log n}{1000}$*, we have*

$$\mathbb{E}\left[\zeta(x, \pi) \mid \mathcal{G}\right] \geq n^{-\frac{3}{4}}.$$

The proof of Theorem C.3 uses the following claims:

Observation C.4. *Due to subadditivity of entropy (Fact B.6) and Fact B.4, we have* $\mathbb{H}(X \mid \pi) \leq \sum_i \mathbb{H}(X^i \mid \pi) \leq \sum_i \log(|S^i(\pi)|)$*. In particular,* $|G_2(\pi)| \leq \frac{19n}{20} \implies \mathbb{H}(X \mid \pi) \leq \frac{19n}{20} \log(2n) + \frac{n}{40} \log n$*.*

Lemma C.5. *If* $T \leq \frac{n \log n}{1000}$*, then,* $\Pr(\overline{\mathcal{G}}) \leq \frac{2}{3}$*.*

Proof. If (x, π) is such that $\overline{\mathcal{G}}$ happens, then, $|G(x, \pi)| < \frac{n}{4}$. Thus, either $|G_1(x)| < \frac{n}{3}$ or $|G_2(\pi)| < \frac{19n}{20}$. By Lemma B.8, the probability of the former is at most $\frac{3}{5}$. Let $\mathcal{G}_2$ be the event $|G_2(\pi)| \leq \frac{19n}{20}$. We now show that $\Pr(\mathcal{G}_2) \leq \frac{1}{50}$ using information theory finishing the proof. We have

$$\begin{aligned}
T \geq \mathbb{H}(\Pi) &\geq \mathbb{I}(X : \Pi) && \text{(By Fact B.4 and Fact B.5)} \\
&= \mathbb{H}(X) - \mathbb{H}(X \mid \Pi) \\
&= n \log(2n) - \mathop{\mathbb{E}}_{\pi} \left[\mathbb{H}(X \mid \pi)\right].
\end{aligned}$$

For all π such that $\mathcal{G}_2$ happens, we upper bound $\mathbb{H}(X \mid \pi)$ using Observation C.4. For the rest, we use the trivial bound of $n \log(2n)$ given by Fact B.4. This gives:

$$T \geq \frac{n \log n}{40} \Pr(\mathcal{G}_2),$$

implying the result. □

Proof of Theorem C.3. In this proof, we refer to $\mathcal{G}$ as both an event and a set of (x, π). Observe that

$$\mathbb{E}\left[\zeta(x,\pi) \mid \mathcal{G}\right] = \sum_{x,\pi} \Pr(x, \pi \mid \mathcal{G})\zeta(x,\pi) = \frac{1}{\Pr(\mathcal{G})} \sum_{(x,\pi)\in\mathcal{G}} \Pr(x,\pi)\zeta(x,\pi).$$

We continue using that fact $\zeta(\cdot,\cdot)$ is non-negative.

$$\begin{aligned}
\mathbb{E}\left[\zeta(x,\pi) \mid \mathcal{G}\right] &\geq \sum_{(x,\pi)\in\mathcal{G}\cap\mathcal{C}} \Pr(x,\pi)\zeta(x,\pi) \\
&\geq \frac{\left(\sum_{(x,\pi)\in\mathcal{G}\cap\mathcal{C}} \Pr(x,\pi)\right)^2}{\sum_{(x,\pi)\in\mathcal{G}\cap\mathcal{C}} Z(x,\pi)} && \text{(Lemma B.7)} \\
&\geq \frac{\Pr(\mathcal{G}\cap\mathcal{C})^2}{\sum_{(x,\pi)\in\mathcal{C}} Z(x,\pi)} \geq \frac{\left(\Pr(\mathcal{C}) - \Pr(\overline{\mathcal{G}})\right)^2}{\sum_{(x,\pi)\in\mathcal{C}} Z(x,\pi)}.
\end{aligned}$$

The numerator is easily bounded from below by $((\frac{2}{3} + n^{-\frac{1}{8}}) - \frac{2}{3})^2 = n^{-\frac{1}{4}}$ using Lemma C.5. We solely focus on the denominator now. We have

$$\sum_{(x,\pi)\in\mathcal{C}} Z(x,\pi) = \sum_{(x,\pi)\in\mathcal{C}} \sum_{i\in G(x,\pi)} \mathop{\mathbb{E}}_{y\sim S^i(\pi)} \left[\Pr(x^{i=y},\pi)\right].$$

For all $i \in G(x,\pi)$, we have $|S^i(\pi)| > \sqrt{n}$ implying

$$\sum_{(x,\pi)\in\mathcal{C}} Z(x,\pi) \leq \frac{1}{\sqrt{n}} \sum_{(x,\pi)\in\mathcal{C}} \sum_{i\in G(x,\pi)} \sum_{y\in S^i(\pi)} \Pr(x^{i=y},\pi). \tag{3}$$

Now, define the set

$$T(\pi) = \{x' \mid \exists x, i, y : (x,\pi) \in \mathcal{C} \text{ and } i \in G(x,\pi) \text{ and } y \in S^i(\pi) \text{ and } x' = x^{i=y}\}.$$

By definition, all the terms in sum in Equation 3 are of the form $\Pr(x',\pi)$ for $x' \in T(\pi)$. However, a given term may be repeated multiple times. We now claim that the number of repetitions is at most n. Formally,

Claim. *Fix a tuple (x',π). For any $i \in [n]$, there is at most one tuple (x,y) such that*

$$(x,\pi) \in \mathcal{C} \text{ and } i \in G(x,\pi) \text{ and } x' = x^{i=y}.$$

Proof. Proof by contradiction. Suppose there are (x_1, y_1) and (x_2, y_2) satisfying the given

properties. Observe that x_1 and x_2 differ only in x_1^i and x_2^i. Also, $i \in G(x_1, \pi)$ implies that $x_1^i \in \mathcal{L}(x_1)$ and $x_1^i \notin \mathcal{L}(x_2)$. Thus, $\mathcal{L}(x_1) \neq \mathcal{L}(x_2)$. Since we assume that the output of player 1 is determined solely by the transcript, $\mathcal{L}(x_1) \neq \mathcal{L}(x_2)$ implies that (x_1, π) and (x_2, π) cannot both be in $\mathcal{C}$, a contradiction.

□

Due to the foregoing claim, we have $\sum_{(x,\pi)\in\mathcal{C}} Z(x,\pi) \leq \sqrt{n}$. This implies

$$\mathbb{E}\left[\zeta(x,\pi) \mid \mathcal{G}\right] \geq \frac{\left(\Pr(\mathcal{C}) - \Pr(\overline{\mathcal{G}})\right)^2}{\sum_{(x,\pi)\in\mathcal{C}} Z(x,\pi)} \geq \frac{n^{-\frac{1}{4}}}{\sqrt{n}} = n^{-\frac{3}{4}}.$$

□

C.3.3 Proof of Theorem C.1 Given Theorem C.2 and Theorem C.3

Proof of Theorem C.1 . We assume that n is large, say $n > 10^{100}$. Assume, for the sake of contradiction, that there is a protocol Π of length $T < \frac{n \log n}{1000}$ that solves $\mathsf{InputSet}_n$ correctly with probability at least $\frac{3}{4}$. Thus, $\Pr(\mathcal{C}) \geq \frac{3}{4}$. By Theorem C.3, we have $\mathbb{E}[\zeta(x,\pi) \mid \mathcal{G}] \geq n^{-\frac{3}{4}}$. However, all the terms $\zeta(x,\pi)$ in this expectation such that $\Pr(x,\pi) > 0$ satisfy $\zeta(x,\pi) \leq \frac{4}{n} \cdot n^{0.01} < n^{-0.90}$ (Theorem C.2). This clearly contradicts $\mathbb{E}[\zeta(x,\pi) \mid \mathcal{G}] \geq n^{-\frac{3}{4}}$. □

D Coding Scheme With Logarithmic Overhead (Proof of Theorem 1.2)

In this section, we give an interactive coding scheme for simulating any noiseless beeping protocol with an $\mathcal{O}(\log n)$ overhead such that the error of the simulation is polynomially small in n. The scheme works for both the correlated noise and the independent noise models.

It is easy to see that protocols in the n-party noiseless beeping model that are polynomial in length (say at most n^{100}) can be simulated over the noisy beeping model with an $\mathcal{O}(\log n)$ overhead. This is done by repeating each transmission in the noiseless beeping model $c \log n$ times, for a constant c large enough so that a majority of the receptions are correct with probability polynomially small (say at most n^{-200}). The correctness of the protocol then follows using a union bound.

As argued in the sketch in subsection 2.1, in order to simulate noiseless beeping protocols of arbitrary length, one can follow the 'rewind-if-error' approach where the long protocol is partitioned into small chunks and the protocol is simulated chunk by chunk. For the beeping model, our discussion there showed that merely simulating a chunk to get a transcript π is not sufficient. Instead, one also needs to compute an 'owner' for every 1 in the transcript π.

We next describe such a simulation procedure a for a given chunk. The owner should be a player that beeped 1 in that round.

D.1 Simulating One Chunk

We now describe our simulation procedure for a chunk. We take our chunks to be of size n. First, we simulate this chunk by repeating every round $\mathcal{O}(\log n)$ times and obtaining a transcript π. Next, we run a second phase that computes an owner for every 1 in π, *i.e.*, a person who beeped 1 in the corresponding round. This phase (similar to the protocol in [BO15]) works as follows: Starting with player 1, all the players beep all the 1s that they can own. Since the transcript π has n rounds, any 1 in π can be identified with at most $\log n + 1$ bits. The players also encode these using constant rate error correcting codes to ensure correct reception except with polynomially small probability. Once player 1 has listed all the 1s in π that he can own, he beeps a special $\mathcal{O}(\log n)$ length sequence that indicates that player 2 should start listing. Thereafter, it is player 2's turn to beep all the 1s that he can own, without repeating those already beeped by player 1, and so on.

We formalize the above in Algorithm 1, where Π is used to denote a chunk of the noiseless protocol to be simulated. Algorithm 1 has two phases as above. In the *simulation phase*, the parties simulate Π by repetition, *i.e.*, every round in Π is repeated $c \cdot \log n$ times (for large enough c) and players take the majority of the bits received. For simplicity, we omit this detail in the description below and assume that all the parties have a correctly simulated transcript π in the second phase. In the second 'finding owners' phase, the players compute owners for every 1 in the transcript π computed in the simulation phase.

In Algorithm 1, for a transcript π, we use $J = \{j \mid \pi_j = 1\}$ to denote the set of 1s in π. This deviates from our notation in subsection C.2 where we use J to denote the 0s instead. We use $C : [n] \cup \{\mathsf{Next}\} \to \{0,1\}^{c \log n}$ to denote a constant rate error correcting code of relative distance 0.99. The constant c is chosen in the analysis to ensure correct decoding with high probability.

It is readily seen that this procedure takes $\mathcal{O}(n \log n)$ rounds as desired. We now formally analyze the finding owners phase and show that it works as desired.

Theorem D.1. *For $i, m \in [n]$, let b^i_m be an input to player i in the finding owners phase in Algorithm 1. Also, define a transcript $\pi \in \{0,1\}^n$ such that $\pi_m = \bigvee_{i \in [n]} b^i_m$. Except with probability at most n^{-10}, it holds at the end of Algorithm 1 that:*

For all $j \in J$, the variables o^i_j have the same value o_j for all players i. Additionally, this value o_j satisfies $b^{o_j}_j = 1$.

Proof. We choose C to be an error correcting code such that the probability that all the players decode correctly in all the $2n$ iterations is at least $1 - n^{-10}$. We condition on this event for the rest of the proof.

It is easily seen that if all the players decode correctly in all the iterations, then the value the variables T^i, turn^i, and o^i_j in our protocol is the same for all players. This proves the

Algorithm 1 Simulating one chunk Π of length n

Simulation phase:

Simulate Π round by round and obtain a transcript π. For $i, m \in [n]$, let the bit b_m^i be the bit beeped by player i in round m.

Finding owners:

Input: For $i, m \in [n]$, player i has a bit b_m^i. The players also share a transcript π where $\pi_m = \bigvee_{i \in [n]} b_m^i$.

Output: For all $j \in J$, player i outputs $o_j^i \in [n]$. After a correct execution, o_j^i are the same for all i and indicate an owner of round j.

$T^i \leftarrow \emptyset, \forall i \in [n]$.
$\text{turn}^i \leftarrow 1, \forall i \in [n]$.
for $l \in [2n]$ **do**
    if $\text{turn}^i = i$ **then**
        if $\exists j \notin T^i : b_j^i = 1$ **then**
            Party i beeps $C(j)$.
        else
            Party i beeps $C(\mathsf{Next})$.
        end if
    end if
    All players decode what was received. Let σ^i be the value decoded.
    If $\sigma^i = \mathsf{Next}$, increment turn^i. Else, let $\sigma^i = j \in [n]$. Set $T^i \leftarrow T^i \cup \{j\}$ and $o_j^i \leftarrow \text{turn}^i$.
end for
Player i outputs o_j^i for all $j \in J$.

first part of our claim. We denote these common values by T, turn, and o_j respectively.

We now claim that the value o_j is updated at least once for all $j \in J$. Suppose, for contradiction, that for some $j \in J$, the value of o_j is never updated. Then, $j \notin T$ and from our assumption that the players always decode correctly, we get that $C(j)$ was never beeped. This means that $b_j^i = 0$ for all values i taken by turn. However, turn takes all values in $[n]$ as after each iteration, either turn is incremented or T gets an additional element. Since T can get at most n elements and there are $2n$ iterations. We can conclude that turn takes all values in $[n]$. Thus, $\forall i, b_j^i = 0 \implies \pi_j = 0 \implies j \notin J$, a contradiction.

When o_j is updated, it gets the value turn. Furthermore, o_j is updated only when player turn beeped $C(j)$ implying $b_j^{o_j} = 1$. □

D.2 Rest of the Simulation

The rest of the simulation proceeds along the lines of [EKS18]. We define a hierarchy of simulation protocols $\mathcal{A}_l$ for $l \geq 0$, where $\mathcal{A}_l$ simulates 2^l chunks and generates a transcript π of length $2^l \cdot n$ with owners for all the 1s in π. The protocol $\mathcal{A}_l$, for $l > 0$, has two executions of $\mathcal{A}_{l-1}$ followed by a progress check. If π_1 and π_2 are the transcripts generated by these two executions, then, the progress check fails with probability exponentially small in l and returns the longest prefix of $\pi_1\pi_2$ that was successfully simulated.

In the progress check, the parties find the longest prefix that was correctly simulated by binary search. To decide whether or not a prefix is correctly simulated, the parties see if they have any of the bits in the prefix does not match what they beeped in that round. For bits in the prefix that are 1, only the owner preforms this check. For the bits that are 0, this check is performed by all the parties. This can be repeated $\mathcal{O}(l)$ times to ensure correctness except with probability exponentially small in l.

To analyze this scheme, we define a progress measure that is the expected length of the prefix that has been correctly simulated and for which, owners have been correctly computed. By Theorem D.1 the expectation of this measure is at least $n(1 - o(1))$ for $\mathcal{A}_0$. Going from $\mathcal{A}_{l-1}$ to $\mathcal{A}_l$, this measure almost doubles (with a loss exponentially small in l). Since a sum of exponentially small terms converges, we get scheme with only logarithmic overhead for arbitrary lengths. For further technical details, see [EKS18].

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