Optimal Message-Passing with Noisy Beeps

Peter Davies
peter.w.davies@durham.ac.uk
Durham University
Durham, UK

ABSTRACT

Beeping models are models for networks of weak devices, such as sensor networks or biological networks. In these networks, nodes are allowed to communicate only via emitting beeps: unary pulses of energy. Listening nodes only the capability of *carrier sensing*: they can only distinguish between the presence or absence of a beep, but receive no other information. The noisy beeping model further assumes listening nodes may be disrupted by random noise.

Despite this extremely restrictive communication model, it transpires that complex distributed tasks can still be performed by such networks. In this paper we provide an optimal procedure for simulating general message passing in the beeping and noisy beeping models. We show that a round of Broadcast CONGEST can be simulated in $O(\Delta \log n)$ round of the noisy (or noiseless) beeping model, and a round of CONGEST can be simulated in $O(\Delta^2 \log n)$ rounds (where Δ is the maximum degree of the network). We also prove lower bounds demonstrating that no simulation can use asymptotically fewer rounds.

This allows a host of graph algorithms to be efficiently implemented in beeping models. As an example, we present an $O(\log n)$-round Broadcast CONGEST algorithm for maximal matching, which, when simulated using our method, immediately implies a near-optimal $O(\Delta \log^2 n)$-round maximal matching algorithm in the noisy beeping model.

A full-length preprint version of this paper is also available [11].

CCS CONCEPTS

• **Networks** → **Network algorithms**; • **Theory of computation** → **Distributed algorithms**; • **Mathematics of computing** → Coding theory.

KEYWORDS

Message Passing, Beeping Model, Superimposed Codes

ACM Reference Format:
Peter Davies. 2023. Optimal Message-Passing with Noisy Beeps. In *ACM Symposium on Principles of Distributed Computing (PODC '23), June 19–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583668.3594594

PODC '23, June 19–23, 2023, Orlando, FL, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0121-4/23/06.
https://doi.org/10.1145/3583668.3594594

1 INTRODUCTION

Beeping models were first introduced by Cornejo and Kuhn [8] to model wireless networks of weak devices, such as sensor networks and biological networks [2]. These models are characterised by their very weak assumptions of communication capabilities: devices are assumed to communicate only via *carrier sensing*. That is, they have the ability to distinguish between the presence or absence of a signal, but not to gain any more information from the signal.

1.1 Models

The models we study all have the same basic structure: a network of devices is modeled as a graph with n nodes (representing the devices) and maximum degree Δ, where edges represent direct reachability between pairs of devices. We will assume that all nodes activate simultaneously, and therefore have shared global clock (some prior work on beeping models instead allow nodes to activate asynchronously). Time then proceeds in synchronous rounds, in which nodes can perform some local computation and then can communicate with neighboring devices. The defining characteristic of each model is the communication capability of the nodes.

Noiseless Beeping Model. In each round, each node chooses to either beep or listen. Listening nodes then hear a beep iff at least one of their neighbors beeped, and silence otherwise. Nodes do not receive any other information about the number or identities of their beeping neighbors.

Noisy Beeping Model. The noisy beeping model, introduced by Ashkenazi, Gelles, and Leshem [4], is similar to the noiseless version, except that the signal each listening node hears (beep or silence) is *flipped*, independently uniformly at random, with some probability $\varepsilon \in (0, \frac{1}{2})$.

Within these beeping models, our aim will be to simulate more powerful message-passing models, in which nodes have the ability to send longer messages to each other, and these messages are received without interference:

Broadcast CONGEST Model. In rounds of the Broadcast CONGEST model, nodes may send the same $O(\log n)$-bit message to each of their neighboring nodes, and each node hears the messages from all of its neighbors.

CONGEST Model. The CONGEST model is similar to Broadcast CONGEST, but allows nodes to send (potentially) different $O(\log n)$-bit messages to each of their neighboring nodes. Again, each node hears the messages from all of its neighbors.

The communication capabilities in the Broadcast CONGEST and CONGEST models are clearly much more powerful than that of either beeping model, and CONGEST in particular has a broad literature of efficient algorithms. Our aim in this work is to provide an efficient generic simulation of Broadcast CONGEST and CONGEST in the beeping models, so that these existing algorithms can be applied out-of-the-box to networks of weak devices.

1.2 Prior Work

Beeping models. The (noiseless) beeping model was introduced by Cornejo and Kuhn [8], who also gave results for an interval coloring task used for synchronization. Classical local graph problems have been studied in the model, with Afek et al. [1] giving an $O(\log^2 n)$-round maximal independent set algorithm, and Beauqier et al. [7] giving $O(\Delta^2 \log n + \Delta^3)$-round deterministic algorithms for maximal independent set and $(\Delta + 1)$-coloring.

Global communication problems (those requiring coordination across the entire network, and therefore with running times parameterized by the diameter D of the network) have also been studied. Single-source broadcast of a b-bit message can be performed in $O(D + b)$ rounds using the simple tool of 'beep waves', introduced by Ghaffari and Haeupler [19] and formalized by Czumaj and Davies [9]. Leader election, another fundamental global problem, has seen significant study in the model. Ghaffari and Haeupler [19] gave a randomized algorithm requiring $O(D + \log n \log\log n) \cdot \min\{\log\log n, \log \frac{n}{D}\}$, while Förster, Seidel and Wattenhofer [16] gave an $O(D \log n)$-round *deterministic* algorithm. Czumaj and Davies [10] gave a simple randomized algorithm with $O(D \log n)$ worst-case round complexity but $O(D + \log n)$ expected complexity. Finally, Dufoulon, Burman and Beauquier [12] settled the complexity of the problem with a deterministic algorithm with optimal $O(D + \log n)$ round complexity.

On other global problems, Czumaj and Davies [9] and Beauqier et al. [6] gave results for broadcasting from multiple sources, and Dufoulon, Burman and Beauqier [13] study synchronization primitives for the model variant where nodes activate asynchronously.

Message passing models. Message passing models, and CONGEST in particular, have seen a long history of study and have a rich literature of algorithms for problems including (among many others) local problems such as $\Delta + 1$-coloring[20], global problems such as minimum spanning tree[24], and approximation problems such as approximate maximum matching[3]. Broadcast CONGEST is less well-studied, though some dedicated algorithms have also been developed for it, e.g. [21]. There is an obvious way to simulate CONGEST algorithms in Broadcast CONGEST at an $O(\Delta)$-factor overhead: nodes simply broadcast the messages for each of their neighbors in turn, appending the ID of the intended recipient. In general this is the best that can be done (as can be seen from our bounds on simulating beeping models), but for specific problems this $\Theta(\Delta)$ complexity gap is often not necessary.

Simulating message passing with beeps. Two works have previously addressed the task of simulating messaging passing in beeping models. The first was by Beauquier et al. [7], and gave a generic simulation for CONGEST in the noiseless beeping model. Their algorithm required Δ^6 setup rounds, and then $\Delta^4 \log n$ beep-model rounds per round of CONGEST. This result was improved by Ashkenazi, Gelles, and Leshem [4], who introduced the noisy beeping model, and gave an improved simulation of CONGEST which requires $O(\Delta^4 \log n)$ rounds of setup, and then simulates each CONGEST round in $O(\Delta \log n \cdot \min\{n, \Delta^2\})$ rounds of noisy beeps.

1.3 Our Results

We give a randomized simulation of Broadcast CONGEST which requires $O(\Delta \log n)$ rounds in the noisy beep model per round of Broadcast CONGEST, with no additional setup cost. We will call this per-round cost the *overhead* of simulation. This implies a simulation of CONGEST with $O(\Delta^2 \log n)$ overhead in the noisy beep model. We therefore improve over the previous best result of [4] by reducing the overhead by a $\Theta(\min\{\frac{n}{\Delta}, \Delta\})$ factor, and removing the large setup cost entirely. We prove that these bounds are tight for both Broadcast CONGEST and CONGEST by giving matching lower bounds (even for the noiseless beeping model). This has the potentially surprising implication that introducing noise into the beeping model does not asymptotically increase the complexity of message-passing simulation at all.

This simulation result allows many CONGEST and Broadcast CONGEST algorithms to be efficiently implemented with beeps. As an example, we show an $O(\log n)$-round Broadcast CONGEST algorithm for the task of maximal matching, which via our simulation implies an $O(\Delta \log^2 n)$-round algorithm in the noisy beeping model. We show that this is almost optimal by demonstrating an $\Omega(\Delta \log n)$ lower bound (even in the noiseless model).

1.4 Our Approach

We summarize our approach to simulating CONGEST in the noiseless beeping model (the noisy case will follow naturally, as we will see later). First, let us mention the general approach of the previous results of [7] and [4]: there, the authors use a coloring of G^2 (i.e., a coloring such that no nodes within distance 2 in G receive the same color) to sequence transmissions. They iterate through the color classes, with nodes in each class transmitting their message (over a series of rounds, with a beep or silence representing each bit of the message). Since nodes have at most one neighbor in each color class, they hear that neighbor's message undisrupted.

The disadvantage of such an approach is that the coloring of G^2 requires a large setup time to compute, and also necessitates at least $\min\{n, \Delta^2\}$ color classes. This is the cause of the larger overhead in the simulation result of [4].

Instead of having nodes transmitting at different times, our solution is to have them all transmit at once, and use superimposed codes to ensure that the messages are decipherable. The definition of a classic superimposed code is as follows:

Definition 1 (Superimposed Codes). *An (a, k)-superimposed code of length b is a function $C : \{0, 1\}^a \to \{0, 1\}^b$ such that any superimposition (bitwise OR) of at most k codewords is unique.*

The connection between superimposed codes and beeping networks is that, if some subset of a node v's neighbors all transmit a message simultaneously (using beeps to represent **1**s and silence to represent **0**s), then v (if it were to listen every round) would hear the bitwise OR superimposition of all the messages. If this superimposition is unique, then v is able to identify the set of messages that were transmitted (and this set contain precisely those messages with no **1** in a position where the superimposition has **0**).

Superimposed codes of this form were first introduced by Kautz and Singleton [23], who showed a construction with $b = O(k^2 a)$. This definition is equivalent to cover-free families of sets, which is the terminology used in much of the prior work. A lower bound $b = \Omega(\frac{k^2 a}{\log k})$ was found by D'yachkov and Rykov [14], with a combinatorial proof later given by Ruszinkó [28], and another, simple proof given by Füredi [17]. The $\log k$ gap between upper and lower bounds remains open.

This presents a problem to applying such codes for message passing in the beep model. If all nodes are transmitting their message (of $O(\log n)$ bits) at once, then we would need to use an $(O(\log n), \Delta)$-superimposed code for the messages to be decodable. Using Kautz and Singleton's construction [23] results in a length of $O(\Delta^2 \log n)$ (and length corresponds directly to rounds in the beeping model). This would result in the same $O(\Delta^2)$-factor overhead as from using a coloring of G^2, so would not improve over [4]. Furthermore, even if we were to find improved superimposed codes, the lower bound implies that any such improvement would be only minor.

To achieve codes with better length, we weaken the condition we require. Rather than requiring that all superimpositions of at most k codewords are unique, we only require that *most* are. Specifically, if the k codewords are chosen at random, then their superimposition will be unique (and hence decodable) with high probability. We show the existence of short codes with this weakened property. Constructions with similar properties (though not quite suitable for our uses) were also given in [15].

This raises a new problem: using these shorter codes, we can efficiently have all nodes send a *random* message to their neighbors, but how does this help us send a specific message?

Our answer is that if we repeat the transmission (using the same random codewords for each node), then every node v already knows exactly when its neighbors should be beeping[1], and in particular, v knows when a neighbor u should be beeping *alone* (i.e., not at the same time as any other neighbor of v). If u now beeps only in a *subset* of the rounds indicated by its codeword, then it can pass information to v in this way. So, our final algorithm uses a secondary *distance* code to specify what this subset should be in order to ensure that all neighbors of u can determine u's message. The aim of this distance code is that codewords are sufficiently large Hamming distance apart that u's neighbors can determine u's message, even though they only hear a subset of the relevant bits, and these bits can be flipped by noise in the noisy model.

1.5 Notation

Our protocols will be heavily based on particular types of binary codes, which we will communicate in the beeping model via beeps and silence. In a particular round, in the noiseless beeping model, we will say that a node v receives a $\mathbf{1}$ if it either listens and hears a beep, or beeps itself. We will say that v receives a $\mathbf{0}$ otherwise. In the noisy model, what v hears will be this bit, flipped with probability ε.

We will use logic operators to denote operations between two strings: for $s, s' \in \{0,1\}^a$, $s \wedge s' \in \{0,1\}^a$ is the logical AND of the two strings, with $\mathbf{1}$ in each coordinate iff both s and s' had $\mathbf{1}$ in that coordinate. Similarly, $s \vee s' \in \{0,1\}^a$ is the logical OR of the two strings, with $\mathbf{1}$ in each coordinate iff s or s' (or both) had $\mathbf{1}$ in that coordinate.

Definition 2. *We will use $\mathbf{1}(s)$ to denote the number of $\mathbf{1}$s in a string $s \in \{0,1\}^a$. We will say that a string $s \in \{0,1\}^a$ d-intersects another string $s' \in \{0,1\}^a$ if $\mathbf{1}(s \wedge s') \geq d$.*

For a set of strings $S \in \{0,1\}^a$, we will use $\vee(S)$ as shorthand for the superimposition $\bigvee_{s \in S} s$.

2 BINARY CODES

The novel type of superimposed code on which our algorithm is mainly based is defined as follows:

Definition 3. *An (a, k, δ)-beep code of length b is a function $C : \{0,1\}^a \to \{0,1\}^b$ such that:*

- *all $s \in C$ have $\mathbf{1}(s) = \frac{\delta b}{k}$.*
- *the number of size-k subsets $S \subseteq C$ whose superimpositions $\vee(S)$ $\frac{5\delta^2 b}{k}$-intersect some $s \in C \setminus S$ is at most $\binom{2^a}{k} 2^{-2a}$*

(here we slightly abuse notation by using C to denote the set of codewords, i.e. the image $C(\{0,1\}^a)$ of the beep code function).

In other words, all codewords have exactly $\frac{\delta b}{k}$ $\mathbf{1}$s, and only a 2^{-2a}-fraction of the $\binom{2^a}{k}$ size-k subsets of codewords have a superimposition that $\frac{5\delta^2 b}{k}$-intersects some other codeword. This first criterion is only a technicality to aid our subsequent application; the important point is the second, which will imply that a superimposition of k *random* codewords will, with probability at least $1 - 2^{-2a}$, be decodable (even under noise, since avoiding $\frac{5\delta^2 b}{k}$-intersection will provide us with sufficient redundancy to be robust to noise). Note that for such a code to exist, $\frac{\delta b}{k}$ must be an integer, which we will guarantee in our construction.

Theorem 4. *For any any $a, k, c \in \mathbb{N}$, there exists an $(a, k, 1/c)$-beep code of length $b = c^2 k a$.*

Proof. The proof will be by the probabilistic method: we will randomly generate a candidate code C, and then prove that it has the desired properties with high probability in 2^a. Then, a code with such properties must exist, and the random generation process we use implies an efficient algorithm to find such a code with high probability (though *checking* the code is correct would require $2^{O(ak)}$ computation).

To generate our candidate code, we choose each codeword independently, uniformly at random from the set of all b-bit strings with $\frac{b}{ck}$ $\mathbf{1}$s. This clearly guarantees the first property.

For a fixed size-k set S of codewords, and a fixed codeword $x \in C \setminus S$, we now analyze the probability that $\vee(S)$ $\frac{5b}{c^2 k}$-intersects x.

Clearly we have $\mathbf{1}(\vee(S)) \leq k \cdot \frac{b}{ck} = b/c$. Consider the process of randomly choosing the positions of the $\mathbf{1}$s of x. Each falls in the same position as a $\mathbf{1}$ of $\vee(S)$ with probability at most $1/c$, even independently of the random choices for the other $\mathbf{1}$s. The probability that $\vee(S)$ $\frac{5b}{c^2 k}$-intersects x is therefore at most

$$\binom{\frac{b}{ck}}{\frac{5b}{c^2 k}} \cdot c^{-\frac{5b}{c^2 k}} \leq \left(\frac{ec}{5}\right)^{\frac{5b}{c^2 k}} \cdot c^{-\frac{5b}{c^2 k}} \leq \left(\frac{5}{e}\right)^{-\frac{5c^2 ka}{c^2 k}} \leq 2^{-4a}$$

[1] Technically, v does not know which neighbor corresponds to which codeword, but this is not required by our approach.

Taking a union bound over all codewords $s \in C \setminus S$, we find that the probability that $\vee(S)$ $\frac{5b}{c^2k}$-intersects any such codeword is at most 2^{-4a}. Then, the expected number of size-k sets S that $\frac{5b}{c^2k}$-intersect any $s \in C \setminus S$ is at most $\binom{2^a}{k}2^{-3a}$. By the probabilistic method, there therefore *exists* a an $(a, k, 1/c)$-beep code in which the number of size-k sets S that $\frac{5b}{c^2k}$-intersect any $s \in C \setminus S$ is at most $\binom{2^a}{k}2^{-3a}$.

However, since we also want an efficient algorithm to *find* an $(a, k, 1/c)$-beep code, we note that by Markov's inequality the probability that more than $\binom{2^a}{k}2^{-2a}$ size-k sets S that $\frac{5b}{c^2k}$-intersect any $s \in C \setminus S$ is at most 2^{-a}, and therefore the process of choosing codewords uniformly at random from all strings with $\frac{b}{ck}$ **1**s gives an $(a, k, 1/c)$-beep code with probability at least $1 - 2^{-a}$.

□

Notice that, while the theorem holds for any $c \in \mathbb{N}$, it is trivial for $c \leq 2$: in this case, codewords cannot $\frac{5b}{c^2k}$-intersect any string, since they contain only $\frac{b}{ck}$ **1**s. Our application will set c to be a sufficiently large constant.

Our algorithm will also make use of *distance codes*. These codes have the simple criterion that every pair of codewords is sufficiently far apart by Hamming distance (which we will denote d_H). Distance codes are an example of error-correcting codes, which have a wealth of prior research (see e.g. [22] for an extensive survey); here we just require a very simple object, for which we give a proof in a similar style to that of Theorem 4 for consistency:

Definition 5. *An (a, δ)-distance code of length b is a function $D : \{0,1\}^a \rightarrow \{0,1\}^b$ such that all pairs $s \neq s' \in D$ have $d_H(s, s') \geq \delta b$.*

Lemma 6. *For any $\delta \in (0, \frac{1}{2})$, $a \in \mathbb{N}$, and $c_\delta \geq 12(1-2\delta)^{-2}$, there exists an (a, δ)-distance code of length $b = c_\delta a$.*

Proof. We randomly generate a candidate code by choosing each codeword's entries independently uniformly at random from $\{0, 1\}$. For any pair of codewords $s, s' \in D$, the probability that they differ on any particular entry is $\frac{1}{2}$. The expected distance is therefore $\frac{b}{2}$, and by a Chernoff bound,

$$\begin{aligned}\mathbf{Pr}\left[d_H(s,s') \leq \delta b\right] &= \mathbf{Pr}\left[d_H(s,s') \leq 2\delta \mathbf{E}\left[d_H(s,s')\right]\right] \\ &\leq e^{\frac{-(1-2\delta)^2 \mathbf{E}\left[d_H(s,s')\right]}{2}} = e^{\frac{-(1-2\delta)^2 c_\delta a}{4}} \quad .\end{aligned}$$

Since $c_\delta \geq 12(1-2\delta)^{-2}$,

$$\mathbf{Pr}\left[dist(s,s') \leq \delta b\right] \leq e^{-3a} \leq 2^{-4a} \quad .$$

Taking a union bound over all $\binom{2^a}{2} \leq 2^{2a}$ pairs $s, s' \in D$, we find that the probability that any pair has $dist(s, s') \leq \delta b$ is at most 2^{-2a}. Therefore, the random generation process generates an (a, δ)-distance code with probability at least $1 - 2^{-2a}$. □

This construction can also be checked relatively efficiently, since one need only check the distance of $O(2^{2a})$ codeword pairs, which can be performed in $2^{O(a)}$ computation.

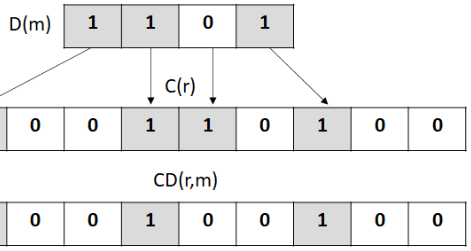

Figure 1: Combined code construction

3 SIMULATION ALGORITHM

We now arrive at our main simulation algorithm. We give an algorithm for simulating a single communication round in Broadcast CONGEST using $O(\Delta \log n)$ rounds of the noisy beep model. What we mean by this simulation is that each node v begins with a $\gamma \log n$-bit message m_v to transmit to all neighbors (where γ is some constant, a parameter of the Broadcast CONGEST model), and by the end of our beeping procedure, all nodes should be able to output the messages of all their neighbors.

Let c_ε be a constant to be chosen based on ε, the noise constant. Our algorithm will make use of two codes (instantiations of those defined in the previous section):

- a $(\gamma \log n, \frac{1}{3})$-distance code D of length $c_\varepsilon^2 \gamma \log n$, given by Lemma 6 (so long as we choose $c_\varepsilon \geq 108$);
- a $(c_\varepsilon \gamma \log n, \Delta{+}1, 1/c_\varepsilon)$-beep code C of length $c_\varepsilon^3 \gamma(\Delta{+}1) \log n$ given by Theorem 4 .

The codewords in the beep code C contain exactly $c_\varepsilon^2 \gamma \log n$ **1**s. The purpose of using these two codes is to combine them in the following manner:

Notation 7. *For a binary string s, let $\mathbf{1}_i(s)$ denote the position of the i^{th} **1** in s (and Null if s contains fewer than i **1**s).*

Let $CD : \{0,1\}^{c_\varepsilon \gamma \log n} \times \{0,1\}^{\gamma \log n} \rightarrow \{0,1\}^{c_\varepsilon^3 \gamma(\Delta+1) \log n}$ be the combined code defined as follows:

$$CD(r,m)_j = \begin{cases} \mathbf{1} & \text{if } \exists i \in [c_\varepsilon^2 \gamma \log n],\ \mathbf{1}_i(C(r)) = j, \text{ and } D(m)_i = \mathbf{1} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

That is, $CD(r, m)$ is the code given by writing the codeword $D(m)$ in the positions where $C(r)$ is **1** (and leaving the other positions as **0**): see Figure 1.

The algorithm is then as follows (Algorithm 1):

Algorithm 1 Simulation of a Broadcast CONGEST round in the noisy beeping model

```
Each node $v$ picks $r_v \in \{0,1\}^{c_\varepsilon \gamma \log n}$ independently uniformly at random
for $i = 1$ to $c_\varepsilon^3 \gamma(\Delta+1)\log n$, in round $i$, do
    Node $v$ beeps iff $C(r_v)_i = 1$
end for
for $i = 1$ to $c_\varepsilon^3 \gamma(\Delta+1)\log n$, in round $i + c_\varepsilon^3 \gamma(\Delta+1)\log n$, do
    Node $v$ beeps iff $CD(r_v, m_v)_i = 1$
end for
```

So, each node picks a random codeword from the beep code, and transmits it bitwise using beeps and silence. By the properties of the beep code, with high probability the superimposition of messages each node receives will be decodable. Then, to actually convey the message m_v, v uses the combined code, which transmits m_v, encoded with a distance code, in the positions where the beep codeword r_v used in the first round was $\mathbf{1}$. Neighbors u of v know when these positions are from the first round. Of course, there are some rounds when other neighbors of u will be beeping, some rounds when u must beep itself and cannot listen, and some rounds when the signal from v is flipped by noise. However, we will show that, by a combination of the properties of our two codes, there is sufficient redundancy to overcome all three of these obstacles, and allow u to correctly decode v's message.

4 DECODING THE CODE

In the first phase, each node v hears[2] a string we will denote $\tilde{x}_v$, which is the string $x_v := \bigvee_{u \in N(v)} C(r_u)$ with each bit flipped with probability $\varepsilon \in (0, \frac{1}{2})$, and the aim is for v to decode this string in order to determine the set $R_v := \{r_u : u \in N(v)\}$.

We first show that, before considering noise, with high probability the superimposition of random codewords chosen by each node's inclusive neighborhood is decodable.

Lemma 8. *With probability at least $1 - n^{3-c_\varepsilon\gamma}$, for every node $v \in V$ and every $r \in \{0,1\}^{c_\varepsilon\gamma\log n}$, $C(r)$ does not $5c_\varepsilon\gamma\log n$-intersect $\bigvee_{r_u \in R_v \setminus \{r\}} C(r_u)$.*

Proof. First, we see that with probability at least $1 - \frac{n^2}{2^{c_\varepsilon\gamma\log n}} = 1 - n^{2-c_\varepsilon\gamma}$, all nodes choose different random strings. For the rest of the proof we condition on this event.

For each $v \in V$, $r \in \{0,1\}^{c_\varepsilon\gamma\log n}$, let $R_{v,r}$ be a set of nodes' random strings defined as follows: starting with $R_v \setminus \{r\}$ (which is a set of input messages of size at most $\Delta+1$), add arbitrary r_x from nodes $x \notin (N(v) \cup \{w\})$ until the set is of size exactly $\Delta+1$. Since we are conditioning on the event that all nodes generate different random strings, $R_{v,w}$ is a set of $\Delta+1$ distinct random strings from $\Delta+1$ distinct nodes, none of which are w.

By the properties of a $(c_\varepsilon\gamma\log n, \Delta+1, 1/c_\varepsilon)$-beep code, therefore, the probability that $C(r)$ $5c_\varepsilon\gamma\log n$-intersects $\bigvee_{r_u \in R_{v,w}} C(r_u)$ is at most $2^{-2c_\varepsilon\gamma\log n} = n^{-2c_\varepsilon\gamma}$. If $C(r)$ does not $5c_\varepsilon\gamma\log n$-intersect $\bigvee_{r_u \in R_{v,w}} C(r_u)$, then since $R_{v,w}$ is a superset of $R_v \setminus \{r\}$, $C(r)$ also does not $5c_\varepsilon\gamma\log n$-intersect $\bigvee_{r_u \in R_v \setminus \{r\}} C(r_u)$, .

The number of possible pairs $v \in V$, $r \in \{0,1\}^{c_\varepsilon\gamma\log n}$ is $n^{1+c_\varepsilon\gamma}$. Taking a union bound over all of these, we find that $C(r)$ does not $5c_\varepsilon\gamma\log n$-intersect $\bigvee_{r_u \in R_v \setminus \{r\}} C(r_u)$ for any pair with probability at least $1 - n^{1+c_\varepsilon\gamma-2c_\varepsilon\gamma} = 1 - n^{1-c_\varepsilon\gamma}$ by a union bound. Finally, removing the conditioning on the event that nodes' random strings are all different, we reach the condition of the lemma with probability at least $1 - n^{1-c_\varepsilon\gamma} - n^{2-c_\varepsilon\gamma} \geq 1 - n^{3-c_\varepsilon\gamma}$. □

Next we must analyze how noise affects the bitstrings that nodes hear. For any node v, let x_v denote the string v heard, i.e., $\bigvee_{u \in N(v)} C(r_u)$, after each bit is flipped with probability $\varepsilon \in (0, \frac{1}{2})$. To decode the set R_v, v will take

$$\tilde{R}_v = \{r \in \{0,1\}^{c_\varepsilon\gamma\log n} : C(r) \text{ does not } \frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n\text{-intersect } \neg\tilde{x}_v\} \ .$$

That is, it includes all codewords which have fewer than $\frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n$ $\mathbf{1}$s in positions where $\tilde{x}_v$ does not.

Notice that, in the absence of noise, all $C(r)$ for $r \in R_v$ have zero $\mathbf{1}$s in positions where x_v did not, and all $C(r)$ for $r \notin R_v$ have at least $c_\varepsilon(c_\varepsilon - 5)\gamma\log n$, since $C(r)$ contains exactly $c_\varepsilon^2\gamma\log n$ $\mathbf{1}$s and, by Lemma 8, fewer than $5c_\varepsilon\gamma\log n$ of them intersect x_v. So, the goal of our next lemma is to show that noise does not disrupt this by too much.

Lemma 9. *For sufficiently large constant c_ε, with probability at least $1 - n^{4-c_\varepsilon\gamma}$, for all nodes v, $\tilde{R}_v = R_v$.*

Proof. Conditioning on the event of Lemma 8, all $C(r)$ for $r \notin R_v$ $c_\varepsilon(c_\varepsilon - 5)\gamma\log n$-intersect $\neg x_v$. Then, for such an r to be in $\tilde{R}_v$, more than $\mathbf{1}(C(r) \wedge \neg x_v) - \frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n$ of the intersection positions would have to be flipped by noise. The probability of this is clearly minimized when $\mathbf{1}(C(r) \wedge \neg x_v)$ is as low as possible, i.e., $c_\varepsilon(c_\varepsilon-5)\gamma\log n$. Then, $c_\varepsilon(c_\varepsilon-5)\gamma\log n - \frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n = (\frac{3-2\varepsilon}{4}c_\varepsilon - 5)c_\varepsilon\gamma\log n$ positions must be flipped, and the expected number of such flipped positions is $\mu := \varepsilon(c_\varepsilon - 5)c_\varepsilon\gamma\log n$.

To show a low probability of failure, we need that the number of positions that must be flipped for r to be incorrectly categorized is more than its expectation. To do so, we bound the ratio of the two quantities:

$$\begin{aligned}
\frac{(\frac{3-2\varepsilon}{4}c_\varepsilon - 5)c_\varepsilon\gamma\log n}{\varepsilon(c_\varepsilon - 5)c_\varepsilon\gamma\log n} &= \frac{\frac{3-2\varepsilon}{4}c_\varepsilon - 5}{\varepsilon(c_\varepsilon - 5)} \\
&\geq \frac{\frac{3-2\varepsilon}{4}c_\varepsilon - 5}{\frac{c_\varepsilon}{2}} \qquad \text{since } \varepsilon \in (0, \frac{1}{2}) \\
&= \frac{3}{2} - \varepsilon - \frac{10}{c_\varepsilon} \ .
\end{aligned}$$

We will set $c_\varepsilon \geq \frac{60}{1-2\varepsilon}$. Then,

$$\frac{(\frac{3-2\varepsilon}{4}c_\varepsilon - 5)c_\varepsilon\gamma\log n}{\varepsilon(c_\varepsilon - 5)c_\varepsilon\gamma\log n} \geq \frac{3}{2} - \varepsilon - \frac{1-2\varepsilon}{6} = \frac{4-2\varepsilon}{3} > 1 \ .$$

Now that we have bounded the ratio above 1, we can apply a Chernoff bound:

$$\begin{aligned}
&\mathbf{Pr}\left[\mathbf{1}(C(r) \wedge \neg\tilde{x}_v) < \frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n\right] \\
&\quad \leq \mathbf{Pr}\left[\text{more than } \frac{\frac{3-2\varepsilon}{4}c_\varepsilon - 5}{\varepsilon(c_\varepsilon - 5)}\mu \text{ flipped positions}\right] \\
&\quad \leq exp(-\left(\frac{\frac{3-2\varepsilon}{4}c_\varepsilon - 5}{\varepsilon(c_\varepsilon - 5)} - 1\right)^2 \mu/3) \\
&\quad \leq exp(-\left(\frac{4-2\varepsilon}{3} - 1\right)^2 \mu/3) \\
&\quad = exp(-(1-2\varepsilon)^2\,\varepsilon(c_\varepsilon - 5)c_\varepsilon\gamma\log n/27) \ .
\end{aligned}$$

We will now further require that $c_\varepsilon \geq \frac{54}{(1-2\varepsilon)^2\varepsilon} + 5$, which gives:

[2] For simplicity of notation, we will assume that a node counts a round in which it itself beeped as 'hearing' a $\mathbf{1}$, and in the noisy model, flips this $\mathbf{1}$ with probability ε itself. Of course, in practice this is unnecessary, and having full information about its own message can only help a node.

$$\mathbf{Pr}\left[\mathbf{1}(C(r) \wedge \neg\tilde{x}_v) < \frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n\right] \leq exp(-2c_\varepsilon\gamma\log n) \leq n^{-2c_\varepsilon\gamma} \quad .$$

Conversely, for some $r' \in R_v$, $C(r)$ does not 1-intersect $\neg x_v$ (since it is contained in the superimposition that produces x_v). So, for it to $\frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n$-intersect $\neg\tilde{x}_v$, at least $\frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n$ of the positions in which $C(r')$ has a $\mathbf{1}$ (of which there are exactly $c_\varepsilon^2\gamma\log n$, by definition) would need to be flipped in $\tilde{x}_v$. The expected number of such flipped positions is $\mu' := \varepsilon c_\varepsilon^2\gamma\log n$. Since $\varepsilon \in (0, \frac{1}{2})$, have $\frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n > \frac{4\varepsilon}{4}c_\varepsilon^2\gamma\log n = \mu'$, so we can again apply a Chernoff bound:

$$\begin{aligned}\mathbf{Pr}&\left[\mathbf{1}(C(r') \wedge \neg\tilde{x}_v) \geq \frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n\right]\\ &\leq \mathbf{Pr}\left[\text{at least } \frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n \text{ of } C(r')\text{'s } \mathbf{1}\text{s are flipped}\right]\\ &\leq exp(-\left(\frac{\frac{2\varepsilon+1}{4}}{\varepsilon}-1\right)^2\mu'/3)\\ &= exp(-\left(\frac{1}{4\varepsilon}-\frac{1}{2}\right)^2\varepsilon c_\varepsilon^2\gamma\log n/3) \quad .\end{aligned}$$

Requiring that $c_\varepsilon \geq \frac{6}{\varepsilon}\left(\frac{1}{4\varepsilon}-\frac{1}{2}\right)^{-2}$ again gives:

$$\mathbf{Pr}\left[\mathbf{1}(C(r') \wedge \neg\tilde{x}_v) \geq \frac{2\varepsilon+1}{4}c_\varepsilon^2\gamma\log n\right] \leq exp(-2c_\varepsilon\gamma\log n) \leq n^{-2c_\varepsilon\gamma} \quad .$$

So, each codeword is correctly placed in or out of $\tilde{R}_v$ with probability at least $1 - n^{-2c_\varepsilon\gamma}$. Taking a union bound over all $2^{c_\varepsilon\gamma\log n}$ codewords, we have $\tilde{R}_v = R_v$ with probability at least $1 - n^{-c_\varepsilon\gamma}$. Finally, taking another bound over all nodes $v \in V$ and removing the conditioning on the event of Lemma 8 (which occurs with probability at least $1 - n^{3-c_\varepsilon\gamma}$) gives correct decoding at all nodes with probability at least $1 - n^{4-c_\varepsilon\gamma}$. The lemma requires setting $c_\varepsilon \geq \max\{\frac{6}{\varepsilon}\left(\frac{1}{4\varepsilon}-\frac{1}{2}\right)^{-2}, \frac{54}{(1-2\varepsilon)^2\varepsilon}+5, \frac{60}{1-2\varepsilon}\}$. □

We now analyze the second stage of the algorithm, in which nodes transmit their messages using the combined code, and show that this code allows the messages to be decoded.

Lemma 10. *In the second phase of the algorithm, with probability at least $1 - n^{\gamma+6-c_\varepsilon\gamma}$, all nodes v can successfully decode $\{m_w : w \in N(v)\}$ (so long as c_ε is at least a sufficiently large constant).*

Proof. Conditioned on the event of Lemma 9, all nodes v now know R_v.

In the second stage of the algorithm, in the absence of noise v would hear the string $\bigvee_{w\in N(v)} CD(r_w, m_w)$, which we will denote y_v. To decode the message m_w, for some $w \in N(v)$, it examines the subsequence $y_{v,w}$ defined by $(y_{v,w})_j = (y_v)_i : \mathbf{1}_j(C(w)) = i$. We denote the noisy versions of these strings that v actually hears by $\tilde{y}_v$ and $\tilde{y}_{v,w}$ respectively. (Note that v does not know which neighbor w the strings r_w and $\tilde{y}_{v,w}$ belong to, but it can link them together, which is all that is required at this stage.) Node v decodes m_w as the string $\tilde{m}_w \in \{0,1\}^{\gamma\log n}$ minimizing $d_H(D(\tilde{m}_w), \tilde{y}_{v,w})$. We must show that, with high probability, $\tilde{m}_w = m_w$.

Conditioned on the event of Lemma 8, each $C(r_w)$ for $w \in N(v)$ does not $5c_\varepsilon\gamma\log n$-intersect $\bigvee_{r_u\in R_v\setminus\{r_w\}} C(r_u)$. That is, there are at least $(c_\varepsilon - 5)c_\varepsilon\gamma\log n$ positions in x_v in which $C(r_w)$ has a $\mathbf{1}$ and no other $C(r_u)$ for $u \in N(v)$ does. In these positions j, $(y_{v,w})_j = D(m_w)_j$. So, in total,

$$d_H(D(m_w), y_{v,w}) \leq 5c_\varepsilon\gamma\log n \quad .$$

Under noise, each of the positions in which $y_{v,w}$ matches $D(m_w)$ will be flipped with probability ε in $\tilde{y}_{v,w}$. So, denoting by μ the expectation $\mathbf{E}\left[d_H(D(m_w), \tilde{y}_{v,w})\right]$, we have:

$$\varepsilon c_\varepsilon^2\gamma\log n \leq \mu \leq \varepsilon c_\varepsilon^2\gamma\log n + 5c_\varepsilon\gamma\log n \quad .$$

Meanwhile,by the property of a $(\gamma\log n, \frac{1}{3})$-distance code, for any $m \neq m_w \in \{0,1\}^{\gamma\log n}$,

$$\begin{aligned}d_H(D(m), y_{v,w}) &\geq d_H(D(m), D(m_w)) - d_H(D(m_w), y_{v,w})\\ &\geq \frac{1}{3}c_\varepsilon^2\gamma\log n - 5c_\varepsilon\gamma\log n \quad .\end{aligned}$$

To lower-bound $\mathbf{E}\left[d_H(D(m), \tilde{y}_{v,w})\right]$ (which we denote by μ'), we see that $\mu' = (1-\varepsilon)d_H(D(m), y_{v,w}) + \varepsilon(c_\varepsilon^2\gamma\log n - d_H(D(m), y_{v,w}))$. Since $\varepsilon > \frac{1}{2}$, this is minimized when $d_H(D(m), y_{v,w})$ is as small as possible, i.e., $\frac{1}{3}c_\varepsilon^2\gamma\log n - 5c_\varepsilon\gamma\log n$. Then,

$$\begin{aligned}\mu' &\geq (1-\varepsilon)(\frac{1}{3}c_\varepsilon^2\gamma\log n - 5c_\varepsilon\gamma\log n)\\ &\quad + \varepsilon(c_\varepsilon^2\gamma\log n - (\frac{1}{3}c_\varepsilon^2\gamma\log n - 5c_\varepsilon\gamma\log n))\\ &= (\frac{1}{3}c_\varepsilon - \frac{2}{3}\varepsilon c_\varepsilon - 5 + 10\varepsilon + \varepsilon c_\varepsilon)c_\varepsilon\gamma\log n\\ &\geq \frac{1+\varepsilon}{3}c_\varepsilon^2\gamma\log n - 5c_\varepsilon\gamma\log n \quad .\end{aligned}$$

Since $\varepsilon < \frac{1}{2}$, we have $\frac{1+\varepsilon}{3} > \varepsilon$, and so we can see that for sufficiently large c_ε, $\mu \leq \mu'$. So, it remains to show that $d_H(D(m_w), \tilde{y}_{v,w})$ and $d_H(D(m), \tilde{y}_{v,w})$ are concentrated around their expectations.

We first show that, with high probability, $d_H(D(m_w), \tilde{y}_{v,w}) \leq \frac{1+4\varepsilon}{6}c_\varepsilon^2\gamma\log n$. Note that if we set $c_\varepsilon \geq \frac{60}{1-2\varepsilon}$, then

$$\frac{1+4\varepsilon}{6}c_\varepsilon = \varepsilon c_\varepsilon + \frac{1-2\varepsilon}{6}c_\varepsilon > \varepsilon c_\varepsilon + 5 \quad ,$$

and so

$$\frac{1+4\varepsilon}{6}c_\varepsilon^2\gamma\log n > \varepsilon c_\varepsilon^2\gamma\log n + 5c_\varepsilon\gamma\log n \geq \mu \quad .$$

Then, we can apply a Chernoff bound:

$$\begin{aligned}\mathbf{Pr}&\left[d_H(D(m_w), \tilde{y}_{v,w}) \geq \frac{1+4\varepsilon}{6}c_\varepsilon^2\gamma\log n\right]\\ &\leq \mu \cdot \mathbf{Pr}\left[d_H(D(m_w), \tilde{y}_{v,w}) \geq \frac{(1+4\varepsilon)c_\varepsilon}{6} \bigg/ (\varepsilon c_\varepsilon + 5)\right]\\ &\leq exp(-\left(\frac{(1+4\varepsilon)c_\varepsilon}{6\varepsilon c_\varepsilon+30}-1\right)^2\mu/2)\\ &= exp(-\left(\frac{(1-2\varepsilon)c_\varepsilon-30}{6\varepsilon c_\varepsilon+30}\right)^2\mu/2) \quad .\end{aligned}$$

The expression $\frac{(1-2\varepsilon)c_\varepsilon-30}{6\varepsilon c_\varepsilon+30}$ is increasing in c_ε. Therefore, if we ensure that $c_\varepsilon \geq \frac{30}{\varepsilon(1-2\varepsilon)}$, we have

$$\frac{(1-2\varepsilon)c_\varepsilon - 30}{6\varepsilon c_\varepsilon + 30} \geq \frac{\frac{30}{\varepsilon} - 30}{\frac{180}{1-2\varepsilon} + 30} = \frac{\frac{1}{\varepsilon} - 1}{\frac{6}{1-2\varepsilon} + 1} = \frac{(1-\varepsilon)(1-2\varepsilon)}{\varepsilon(7-2\varepsilon)} \quad .$$

Then,

$$\begin{aligned}
\mathbf{Pr}&\left[d_H(D(m_w), \tilde{y}_{v,w}) \geq \frac{1+4\varepsilon}{6} c_\varepsilon^2 \gamma \log n\right] \\
&\leq exp(-\left(\frac{(1-\varepsilon)(1-2\varepsilon)}{\varepsilon(7-2\varepsilon)}\right)^2 \mu/2) \\
&\leq exp(-\left(\frac{(1-\varepsilon)(1-2\varepsilon)}{\varepsilon(7-2\varepsilon)}\right)^2 c_\varepsilon^2 \gamma \log n/2) \quad .
\end{aligned}$$

Finally, if we also ensure that $c_\varepsilon \geq 6\left(\frac{(1-\varepsilon)(1-2\varepsilon)}{\varepsilon(7-2\varepsilon)}\right)^{-2}$,

$$\begin{aligned}
\mathbf{Pr}\left[d_H(D(m_w), \tilde{y}_{v,w}) \geq \frac{1+4\varepsilon}{6} c_\varepsilon^2 \gamma \log n\right] &\leq exp(-3c_\varepsilon \gamma \log n) \\
&\leq n^{-4c_\varepsilon \gamma} \quad .
\end{aligned}$$

We similarly wish to show that with high probability,

$$d_H(D(m), \tilde{y}_{v,w}) > \frac{(1+4\varepsilon)}{6} c_\varepsilon^2 \gamma \log n$$

(for $m \neq m_w$). Again, since we have set $c_\varepsilon \geq \frac{60}{1-2\varepsilon}$,

$$\frac{(1+4\varepsilon)}{6} c_\varepsilon = \frac{1+\varepsilon}{3} c_\varepsilon - \frac{1-2\varepsilon}{6} c_\varepsilon < \frac{1+\varepsilon}{3} c_\varepsilon - 5$$

So,

$$\frac{1+4\varepsilon}{6} c_\varepsilon^2 \gamma \log n < \frac{1+\varepsilon}{3} c_\varepsilon^2 \gamma \log n - 5 c_\varepsilon \gamma \log n \leq \mu' \quad .$$

Then, we can apply a Chernoff bound:

$$\begin{aligned}
\mathbf{Pr}&\left[d_H(D(m), \tilde{y}_{v,w}) \leq \frac{(1+4\varepsilon)}{6} c_\varepsilon^2 \gamma \log n\right] \\
&\leq \mathbf{Pr}\left[d_H(D(m), \tilde{y}_{v,w}) \leq \mu' \cdot \frac{(1+4\varepsilon)c_\varepsilon}{6} \Big/ \left(\frac{1+\varepsilon}{3} c_\varepsilon - 5\right)\right] \\
&\leq exp(-\left(1 - \frac{(1+4\varepsilon)c_\varepsilon}{2(1+\varepsilon)c_\varepsilon - 30}\right)^2 \mu/3) \\
&\leq exp(-\left(\frac{(1-2\varepsilon)c_\varepsilon - 30}{6\varepsilon c_\varepsilon + 30}\right)^2 \mu/3) \\
&\leq exp(-\left(\frac{(1-\varepsilon)(1-2\varepsilon)}{\varepsilon(7-2\varepsilon)}\right)^2 c_\varepsilon^2 \gamma \log n/3) \\
&\leq exp(-2c_\varepsilon \gamma \log n) \\
&\leq n^{-2c_\varepsilon \gamma} \quad .
\end{aligned}$$

Taking a union bound over all strings in $\{0,1\}^{\gamma \log n}$, we find that with probability at least $1 - n^{\gamma - 2c_\varepsilon \gamma}$, $d_H(D(m_w), \tilde{y}_{v,w}) < \frac{1+4\varepsilon}{6} c_\varepsilon^2 \gamma \log n$ and $d_H(D(m), \tilde{y}_{v,w}) > \frac{(1+4\varepsilon)}{6} c_\varepsilon^2 \gamma \log n$ for all $m \neq m_w$. So, v successfully decodes m_w. Another union bound over all $w \in N(v)$ gives probability at least $1 - n^{\gamma+1-2c_\varepsilon \gamma}$ that v correctly decodes the entire set $\{m_w : w \in N(v)\}$. Finally, removing the conditioning on the event of Lemma 9 and taking a further union bound over all nodes v, the probability that all nodes correctly decode their neighbors' messages is at least $1 - n^{\gamma+6-c_\varepsilon \gamma}$. We required that

$$c_\varepsilon \geq \max\left\{\frac{30}{\varepsilon(1-2\varepsilon)}, 6\left(\frac{(1-\varepsilon)(1-2\varepsilon)}{\varepsilon(7-2\varepsilon)}\right)^{-2}\right\} \quad .$$

□

Lemma 10 shows that Algorithm 1 successfully simulates a Broadcast CONGEST communication round with high probability. By simulating all communication rounds in sequence, we can simulate any $n^{O(1)}$ Broadcast CONGEST in its entirety at an $O(\Delta \log n)$ overhead. Note that essentially all Broadcast CONGEST (and CONGEST) algorithms are $n^{O(1)}$-round, since this is sufficient to inform all nodes of the entire input graph. So the only problems with super-polynomial round complexities would be those in which nodes are given extra input of super-polynomial size. We are not aware of any such problems having been studied, and therefore Theorem 11 applies to all problems of interest.

Theorem 11. *Any $T = n^{O(1)}$-round Broadcast CONGEST algorithm can be simulated in the noisy beeping model in $O(T\Delta \log n)$ rounds, producing the same output with with probability at least $1 - n^{-2}$.*

Proof. Each round of the Broadcast CONGEST algorithm, in which each node v broadcasts a $\gamma \log n$-bit message to all of its neighbors, is simulated using Algorithm 1 with sufficiently large constant c_ε. By Lemma 10, each simulated communication round succeeds (has all nodes correctly decode the messages of their neighbors) with probability at least $1 - n^{\gamma+6-c_e ps\gamma}$. Taking a union bound over all T rounds, and choosing c_ε sufficiently large, gives a probability of at least $1 - n^{-2}$ that all simulated communication rounds succeed. In this case, the algorithm runs identically as it does in Broadcast CONGEST, and produces the same output. The running time of Algorithm 1 is $O(\Delta \log n)$, so the overall running time is $O(T\Delta \log n)$. □

We then reach an $O(\Delta^2 \log n)$-overhead simulation for CONGEST.

Corollary 12. *Any $T = n^{O(1)}$-round CONGEST algorithm can be simulated in the noisy beeping model in $O(T\Delta^2 \log n)$ rounds, producing the same output with with probability at least $1 - n^{-2}$.*

Proof. A $T = n^{O(1)}$-round CONGEST algorithm can be simulated in $O(T\Delta)$ rounds in Broadcast CONGEST as follows: nodes first broadcast their IDs to all neighbors, and then each CONGEST communication round is simulated in Δ Broadcast CONGEST rounds by having each node v broadcast $\langle ID_u, m_{v \to u} \rangle$ to its neighbors, for every $u \in N(v)$ in arbitrary order. Then, by Theorem 11, this algorithm can be simulated in $O(T\Delta^2 \log n)$ rounds. □

5 LOWER BOUNDS

We now show lower bounds on the number of rounds necessary to simulate Broadcast CONGEST and CONGEST, based on the hardness of a simple problem we call B-bit Local Broadcast. We define the B-bit Local Broadcast problem as follows:

Definition 13 (B-Bit Local Broadcast). *Every node v is equipped with a unique identifier $ID_v \in [n]$. Every node v receives as input $\{\langle ID_u, m_{v\to u}\rangle : u \in N(v)\}$: that is, a set containing messages $m_{v\to u} \in \{0,1\}^B$ for each of v's neighbors u, coupled with the ID of u to identify the destination node. Each node v must output the set $\{\langle ID_u, m_{u\to v}\rangle : u \in N(v)\}$ (i.e. the set of messages* from *each of its neighbors, coupled with their IDs).*

Lemma 14. *B-Bit Local Broadcast requires $\Omega(\Delta^2 B)$ rounds in the beeping model (even without noise), for any algorithm succeeding with probability more than $2^{-\frac{1}{2}\Delta^2 B}$.*

Proof. The graph we use as our hard instance is as follows: we take the complete bipartite graph $K_{\Delta,\Delta}$, and add $n - 2\Delta$ isolated vertices. This graph then has n vertices and maximum degree Δ. Arbitrarily fix unique IDs in $[n]$ for each node. We will only consider the nodes of $K_{\Delta,\Delta}$ to show hardness. Arbitrarily denote one part of the bipartition L and the other R. For nodes $v \in L$, we choose each $m_{v\to u}$ independently uniformly at random from $\{0,1\}^B$. We set all other $m_{x\to x}$ to $\mathbf{0}^{\log n}$ (so, in particular, the inputs for all nodes $u \in R$ are identical).

Let $\mathcal{R}$ denote the concatenated strings of local randomness of all nodes in R (in any arbitrary fixed order). Then, the output of any node $u \in R$ must be fully deterministically dependent on the node IDs (which are fixed), $\mathcal{R}$, u's input messages (which are identically fixed to be all $\mathbf{0}$s), and the pattern of beeps and silence of nodes in L (and note that all nodes in R hear the same pattern: a beep if an node in L beeps, and silence otherwise). An algorithm running for T rounds has 2^T possible such patterns of beeps and silence.

So, the overall output of all nodes in R must be one of 2^T possible distributions, where the distribution is over the randomness of $\mathcal{R}$. The correct output for these nodes is uniformly distributed over $2^{\Delta^2 B}$ possibilities (the choices of input messages for L). The probability of a correct output is therefore at most $2^{T-\Delta^2 B}$. So, any algorithm with $T \le \frac{1}{2}\Delta^2 B$ succeeds with probability at most $2^{-\frac{1}{2}\Delta^2 B}$.

□

Having shown a lower bound on the problem in the beeping model, upper bounds in Broadcast CONGEST and CONGEST imply lower bounds on the overhead of simulation.

Lemma 15. *B-Bit Local Broadcast can be solved deterministically in $O(\Delta\lceil B/\log n\rceil)$ rounds of Broadcast CONGEST and in $O(\lceil B/\log n\rceil)$ rounds of CONGEST.*

Proof. In Broadcast CONGEST, each node v simply broadcasts the strings $\langle ID_u, m_{v\to u}\rangle$ for each $u \in N(v)$, taking $O(\Delta\lceil B/\log n\rceil)$ rounds. In CONGEST, node v instead sends $m_{v\to u}$ to node u for each $u \in N(v)$, taking $O(\lceil B/\log n\rceil)$ rounds. □

Corollary 16. *Any simulation of Broadcast CONGEST in the noiseless beeping model (and therefore also the noisy beeping model) has $\Omega(\Delta \log n)$ overhead. Any simulation of CONGEST in the noiseless (and noisy) beeping model has $\Omega(\Delta^2 \log n)$ overhead.*

6 APPLICATION: MAXIMAL MATCHING

In this section we give an example application of our simulation, to the problem of maximal matching. The problem is as follows: we assume each node has a unique $O(\log n)$-bit ID. For a successful maximal matching, each node must either output the ID of another node, or Unmatched. The outputs must satisfy the following:

- Symmetry: iff v outputs $ID(u)$, then u outputs $ID(v)$. Since each node outputs at most one ID, this implies that the output indeed forms a matching.
- Maximality: for every edge $\{u,v\}$ in the graph, u and v do not both output Unmatched.

To our knowledge, no bespoke maximal matching algorithm has previously been designed for the beeping model (either noisy or noiseless) or for Broadcast CONGEST. So, the fastest existing beeping algorithm is obtained by simulating the best CONGEST algorithms using the simulation of [4]. Since an $O(\Delta+\log^* n)$-round CONGEST algorithm for maximal matching exists [26], the running time under [4]'s simulation is therefore $O(\Delta^4 \log n+\Delta^3 \log n \log^* n)$.

We show an $O(\log n)$-round Broadcast CONGEST algorithm for maximal matching, which our simulation then converts to an $O(\Delta \log^2 n)$-round algorithm in the noisy beeping model, thereby improving the running time by around a $\Delta^3/\log n$ factor.

The base of our algorithm is Luby's algorithm for maximal independent set [25], which can be applied to produce a maximal matching (Algorithm 2). [3]

Algorithm 2 Maximal Matching: Luby's Algorithm

```
for $O(\log n)$ iterations, do
    Each edge $e$ samples $x(e)$ independently uniformly at random from $[n^9]$
    Edge $e$ joins the matching $M$ if $x(e) < x(e')$ for all $e'$ adjacent to $e$
    Endpoints of edges in $M$ drop out of the graph
end for
```

It is well-known (see [25]) that Luby's algorithm produces a maximal matching in $O(\log n)$ rounds with high probability. To implement this in Broadcast CONGEST we must make some minor changes to account for the fact that it is nodes, not edges, that communicate (Algorithm 3).

The aim of the algorithm is as follows: if, in a particular round i, an edge $\{u,v\}$ has a lower $x(\{u,v\})$ value than all neighboring edges, the following process occurs. Its higher-ID endpoint (assume W.L.O.G. that this is u) first broadcasts Propose$\langle\{u,v\}, x(\{u,v\})\rangle$. The other endpoint v then broadcasts Reply$\langle\{u,v\}\rangle$. Node u then broadcasts Confirm$\langle\{u,v\}\rangle$, and finally node v also broadcasts Confirm$\langle\{u,v\}\rangle$. These Confirm messages cause nodes adjacent to u and v to be aware that u and v will be ceasing participation (because they have been matched), and so any edges to them can be discarded from the graph.

Lemma 17. *If Algorithm 3 terminates (i.e. causes all nodes to cease participation), it outputs a maximal matching.*

[3] Often this algorithm is stated with real sampled $x(e)$ values from $[0,1]$; however, since we must communicate these values using $O(\log n)$-bit messages, we instead use integers from $[n^9]$. It can be seen that with probability at least $1 - n^{-4}$, no two of the values sampled during the algorithm are the same, so we can condition on this event for the rest of the analysis and avoid considering ties.

Algorithm 3 Maximal Matching in Broadcast CONGEST

```
Each node $v$ broadcasts its ID
Let $E_v$ be the set of $v$'s adjacent edges
Let $H_v$ be the set of $v$'s adjacent edges for which $v$ the higher-ID endpoint
for $i = 1$ to $O(\log n)$, in round $i$, do
    $v$ samples $x(e)$ independently uniformly at random from $[n^9]$ for each $e \in H_v$
    $v$ broadcasts Propose$\langle e_v, x(e_v)\rangle$, where $x(e_v)$ is the unique minimum of $v$'s sampled values (if it exists)
    Let $e'_v$ be the edge with the minimum $x(e'_v)$ for which $v$ received Propose$\langle e'_v, x(e'_v)\rangle$
    if $x(e'_v) < x(e_v)$ then
        $v$ broadcasts Reply$\langle e'_v\rangle$
    end if
    if $v$ received Reply$\langle e_v\rangle$ and did not broadcast a Reply then
        $v$ broadcasts Confirm$\langle e_v\rangle$
        $v$ outputs $e_v \in MM$ and ceases participation
    end if
    if $v$ received Confirm$\langle e'_v\rangle$ then
        $v$ broadcasts Confirm$\langle e'_v\rangle$
        $v$ outputs $e'_v \in MM$ and ceases participation
    end if
    if $v$ received Confirm$\langle\{w, z\}\rangle$ for any $w, z \neq v$ then
        $v$ removes $\{w, v\}$ and $\{z, v\}$ from $E_v$ and $H_v$ (if present).
    end if
    if $E_v$ is empty then
        $v$ ceases participation
    end if
end for
```

Proof. We first prove maximality. Nodes only cease participation when they are adjacent to an edge in MM, or when they have no remaining adjacent edges. Edges are only removed when they are adjacent to an edge in MM. So, upon termination, there are no edges in the original graph that are neither in MM nor adjacent to an edge in MM, and therefore MM is a maximal matching.

We now prove independence. Let $\{u, v\}$ be an edge which is added to MM in round i, and assume W.L.O.G. that u is the higher-ID endpoint. It is clear that, since $\{u, v\}$ is added to MM, we must have the behavior described above (u broadcasts Propose$\langle\{u, v\}, x(\{u, v\})\rangle$, v broadcasts Reply$\langle\{u, v\}\rangle$, u broadcasts Confirm$\langle\{u, v\}\rangle$, v broadcasts Confirm$\langle\{u, v\}\rangle$). Then, we can show that this excludes the possibility that any adjacent edge also joins MM in round i:

(1) u cannot act as the higher-ID endpoint of any other edge joining MM, since it only Proposes $\{u, v\}$.
(2) u cannot act as the lower-ID endpoint of any other edge joining MM, since it Confirms an edge it Proposed, and therefore cannot have broadcast any Reply.
(3) v cannot act as the higher-ID endpoint of any other edge joining MM, since it broadcasts a Reply and therefore does not Confirm$\langle e_v\rangle$.
(4) v cannot act as the lower-ID endpoint of any other edge joining MM, since only broadcasts Reply$\langle\{u, v\}\rangle$, and does not Reply for any other edge.

So, no adjacent edge to $\{u, v\}$ can join in round i. Furthermore, all nodes adjacent to u and v receive a Confirm$\langle\{u, v\}\rangle$ message and therefore all other edges adjacent to u and v are removed from the graph. So, no edge adjacent to $\{u, v\}$ can be added to MM in future rounds either. This guarantees that MM is an independent set of edges. □

Notation 18. *We will use the notation $e \sim e'$ to mean $e \cap e' \neq \emptyset$, i.e., e' shares* at least *one endpoint with e (and can be e itself). We will denote $|\{e' \in E : e \sim e'\}|$ by $d(e)$, i.e. the number of adjacent edges of e, including e itself.*

Lemma 19. *In any particular round i, the expected number of edges removed from the graph is at least $\frac{m}{2}$.*

This lemma refers to the *current* graph at round i, i.e. without all edges and nodes that have been removed in previous rounds, and m is accordingly the number of edges in the current graph.

Proof. It is easy to see that, as intended, an edge $\{u, v\}$ is added to MM if it has a lower $x(\{u, v\})$ value than all neighboring edges: its higher-ID endpoint (W.L.O.G. u), which sampled the value $x(\{u, v\})$, will denote the edge as e_u and Propose it, the $x(\{u, v\})$ value will be lower than that for which v Proposed and so v will Reply$\langle\{u, v\}\rangle$, and u will not hear a lower-valued edge to Reply and will therefore Confirm$\langle\{u, v\}\rangle$. v will also Confirm$\langle\{u, v\}\rangle$, and all edges adjacent to $\{u, v\}$ will be removed from the graph. There are $d(u) + d(v) - 1$ such edges.

The probability that $x(\{u, v\}) < x(e)$ for all $e \sim \{u, v\}$ with $e \neq \{u, v\}$ is $\frac{1}{d(u)+d(v)-1}$. So, the expected number of edges removed from the graph is at least $\frac{1}{2}\sum_{\{u,v\}\in E}(d(u)+d(v)-1)\frac{1}{d(u)+d(v)-1} = \frac{m}{2}$ (where the $\frac{1}{2}$ factor arises since each edge can be removed by either endpoint being matched, so is double-counted in the sum). □

Lemma 20. *Algorithm 3 performs maximal matching in $O(\log n)$ rounds of Broadcast CONGEST, succeeding with high probability.*

Proof. By Lemma 17, Algorithm 3 produces a maximal matching if it terminates. Conditioning on the event that all sampled values are distinct, the algorithm removes at least half of the edges in the graph in each iteration in expectation. After $4 \log n$ iterations, therefore, the expected number of edges remaining is at most $n^2 \cdot n^{-4} = n^{-2}$, and therefore by Markov's inequaility, with probability at least $1 - n^{-2}$ the number of edges remaining is 0 and the algorithm has terminated. Removing the conditioning on the event that sampled values are distinct, the algorithm terminates with probability at least $1 - n^{-2} - n^{-4}$. □

Theorem 21. *Maximal matching can be performed in $O(\Delta \log^2 n)$ rounds in the noisy beeping model, succeeding with high probability.*

Proof. Follows from applying Theorem 11 to Lemma 20 . □

This is close to optimal, since we show an $\Omega(\Delta \log n)$ bound even in the noiseless model:

Theorem 22. *Maximal matching requires $\Omega(\Delta \log n)$ rounds in the (noiseless) beeping model, to succeed with any constant probability.*

Proof. Our hard ensemble of instances is as follows: the underlying graph will be $K_{\Delta,\Delta}$, the complete bipartite graph with Δ vertices in each part. Each node's ID will be drawn independently at random from $[n^4]$.

Arbitrarily naming the two parts of the graph left and right, we consider the outputs of nodes on the right. For a correct output to maximal matching, each node on the right must uniquely output the ID of a node on the left, and so the union of outputs of the right part must be the list of IDs of the right part. The number of possible such lists (even assuming that IDs are all unique and the IDs of the right side are fixed) is $\binom{n^4-\Delta}{\Delta} \geq \binom{\frac{1}{2}n^4}{\Delta} \geq \left(\frac{n^4}{2\Delta}\right)^{\Delta} \geq n^{3\Delta}$.

We note that each right node's output must be dependent only on its ID, its local randomness, and the transcript of communication performed by left nodes during the course of the algorithm. Since the graph is a complete bipartite graph, in each round there are only two discernable possibilities for communication from the perspective of right-part nodes: either at least one left node beeps, or none do. So, the transcript for an r-round algorithm can be represented as a sequence $\{B, S\}^r$, corresponding to hearing a beep or silence in each round. There are 2^r such transcripts.

Therefore, the union of output from right nodes can be expressed as a function of the randomness of right-part nodes (both for their IDs and any local randomness used in the algorithm), and the transcript. Each transcript therefore induces a distribution of right-part outputs, over the randomness of right-part nodes.

There must be some set of left-part IDs such that under any transcript, the probability that the right-side nodes correctly output that set is at most $2^r/n^{3\Delta}$. So, if $r \leq \Delta \log n$, then the probability that the right part produces a correct output on this instance is at most $n^{\Delta}/n^{3\Delta} = n^{-2\Delta} = o(1)$. □

7 CONCLUSIONS

We have presented an optimal method for simulating Broadcast CONGEST and CONGEST in the noisy (and noiseless) beeping model. We have also presented, as an example, a maximal matching algorithm which requires $O(\log n)$ rounds in Broadcast CONGEST, and which, using our simulation, can therefore be run in $O(\Delta \log^2 n)$ rounds in the noisy beeping model.

While our general simulation method is optimal, there is still room for improvement for many specific problems in the beeping model, and the complexity picture has significant differences from the better-understood message passing models. For example, in CONGEST, the problems of maximal matching and maximal independent set have similar $O(\log \Delta + \log^{O(1)} \log n)$ randomized round complexity upper bounds [5, 18, 27], whereas in the beeping model, maximal independent set can be solved in $\log^{O(1)} n$ rounds [1] while maximal matching requires $\Omega(\Delta \log n)$ (Theorem 22). In general, the question of which problems can be solved in $O(\log^{O(1)} n)$ rounds in the beeping model, and which require $poly(\Delta)$ factors, remains mostly open.

REFERENCES

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

[2] Yehuda Afek, Noga Alon, Omer Barad, Eran Hornstein, Naama Barkai, and Ziv Bar-Joseph. 2011. A Biological Solution to a Fundamental Distributed Computing Problem. *Science* 331, 6014 (2011), 183–185. https://doi.org/10.1126/science.1193210 arXiv:https://www.science.org/doi/pdf/10.1126/science.1193210

[3] Mohamad Ahmadi, Fabian Kuhn, and Rotem Oshman. 2018. Distributed Approximate Maximum Matching in the CONGEST Model. In *32nd International Symposium on Distributed Computing (DISC 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[4] Yagel Ashkenazi, Ran Gelles, and Amir Leshem. 2022. Noisy beeping networks. *Information and Computation* (2022), 104925. https://doi.org/10.1016/j.ic.2022.104925

[5] Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. 2016. The locality of distributed symmetry breaking. *Journal of the ACM (JACM)* 63, 3 (2016), 1–45.

[6] Joffroy Beauquier, Janna Burman, Peter Davies, and Fabien Dufoulon. 2019. Optimal Multi-cast with Beeps Using Group Testing. In *Structural Information and Communication Complexity*, Keren Censor-Hillel and Michele Flammini (Eds.). Springer International Publishing, Cham, 66–80.

[7] Joffroy Beauquier, Janna Burman, Fabien Dufoulon, and Shay Kutten. 2018. Fast Beeping Protocols for Deterministic MIS and $(\Delta + 1)$-Coloring in Sparse Graphs. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*. 1754–1762. https://doi.org/10.1109/INFOCOM.2018.8486015

[8] Alejandro Cornejo and Fabian Kuhn. 2010. Deploying wireless networks with beeps. In *International Symposium on Distributed Computing*. Springer, 148–162.

[9] Artur Czumaj and Peter Davies. 2019. Communicating with beeps. *J. Parallel and Distrib. Comput.* 130 (2019), 98–109. https://doi.org/10.1016/j.jpdc.2019.03.020

[10] Artur Czumaj and Peter Davies. 2019. Leader election in multi-hop radio networks. *Theoretical Computer Science* 792 (2019), 2–11.

[11] Peter Davies. 2023. Optimal Message-Passing with Noisy Beeps. arXiv:2303.15346

[12] Fabien Dufoulon, Janna Burman, and Joffroy Beauquier. 2018. Beeping a deterministic time-optimal leader election. In *32nd International Symposium on Distributed Computing (DISC 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[13] Fabien Dufoulon, Janna Burman, and Joffroy Beauquier. 2020. Can Uncoordinated Beeps Tell Stories?. In *Proceedings of the 39th Symposium on Principles of Distributed Computing* (Virtual Event, Italy) *(PODC '20)*. Association for Computing Machinery, New York, NY, USA, 408–417. https://doi.org/10.1145/3382734.3405699

[14] Arkadii Georgievich D'yachkov and Vladimir Vasil'evich Rykov. 1982. Bounds on the length of disjunctive codes. *Problemy Peredachi Informatsii* 18, 3 (1982), 7–13.

[15] Arkadii D'yachkov, Ilya Vorobyev, Nikita Polyanskii, and Vladislav Shchukin. 2017. Almost cover-free codes and designs. *Designs, Codes and Cryptography* 82, 1 (2017), 231–247.

[16] Klaus-Tycho Förster, Jochen Seidel, and Roger Wattenhofer. 2014. Deterministic Leader Election in Multi-hop Beeping Networks. In *Distributed Computing*, Fabian Kuhn (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 212–226.

[17] Zoltán Füredi. 1996. Onr-cover-free families. *Journal of Combinatorial Theory, Series A* 73, 1 (1996), 172–173.

[18] Mohsen Ghaffari. 2016. An improved distributed algorithm for maximal independent set. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 270–277.

[19] Mohsen Ghaffari and Bernhard Haeupler. 2013. Near optimal leader election in multi-hop radio networks. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 748–766.

[20] Magnús M. Halldórsson, Fabian Kuhn, Yannic Maus, and Tigran Tonoyan. 2021. Efficient Randomized Distributed Coloring in CONGEST. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing* (Virtual, Italy) *(STOC 2021)*. Association for Computing Machinery, New York, NY, USA, 1180–1193. https://doi.org/10.1145/3406325.3451089

[21] Yael Hitron and Merav Parter. 2021. Broadcast CONGEST Algorithms against Adversarial Edges. In *35th International Symposium on Distributed Computing (DISC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[22] W Cary Huffman and Vera Pless. 2010. *Fundamentals of error-correcting codes*. Cambridge university press.

[23] William Kautz and Roy Singleton. 1964. Nonrandom binary superimposed codes. *IEEE Transactions on Information Theory* 10, 4 (1964), 363–377.

[24] Valerie King, Shay Kutten, and Mikkel Thorup. 2015. Construction and impromptu repair of an MST in a distributed network with o (m) communication. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*. 71–80.

[25] Michael Luby. 1986. A simple parallel algorithm for the maximal independent set problem. *SIAM journal on computing* 15, 4 (1986), 1036–1053.

[26] Alessandro Panconesi and Romeo Rizzi. 2001. Some simple distributed algorithms for sparse networks. *Distributed computing* 14, 2 (2001), 97–100.

[27] Václav Rozhoň and Mohsen Ghaffari. 2020. Polylogarithmic-time deterministic network decomposition and distributed derandomization. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 350–363.

[28] Miklós Ruszinkó. 1994. On the upper bound of the size of the r-cover-free families. *Journal of Combinatorial Theory, Series A* 66, 2 (1994), 302–310.