Optimal Multi-Broadcast with Beeps using Group Testing

Joffroy Beauquier, Janna Burman, Peter Davies, Fabien Dufoulon

Optimal Multi-Broadcast with Beeps using Group Testing*

Joffroy Beauquier[1], Janna Burman[1], Peter Davies[2], and Fabien Dufoulon[1]

[1] LRI, CNRS UMR 8623, Université Paris-Sud, Université Paris-Saclay, Orsay, France
{beauquier, burman, dufoulon}@lri.fr
[2] University of Warwick, Coventry, UK Peter.Davies.4@warwick.ac.uk

Abstract. The *beeping model* is an extremely restrictive broadcast communication model that relies only on carrier sensing. In this model, we obtain *time-optimal and deterministic* solutions for the fundamental communication task of multi-broadcast. The proposed solutions are completely uniform, i.e., independent of the network and problem parameters.
The originality of our approach lies in the use of *(combinatorial) group testing strategies*, originally developed in the centralized context. We improve on previous solutions to multi-broadcast by giving *efficiently constructible* solutions, that is, with local computation cost polynomial in the identifiers' range.

Keywords: Beeping Model · Group Testing · Multi-Broadcast.

1 Introduction

Wireless networks with weak communication capabilities have received a great deal of interest recently. In particular, new models assuming very severe restrictions on communication capabilities have been proposed. One of them is the *discrete beeping model* ($\mathcal{BEEP}$), introduced by Cornejo and Kuhn [7]. Due to its weak assumptions, $\mathcal{BEEP}$ has broad applicability to many different communication networks. It has strong connections with the ad-hoc radio network model, and has been used to obtain optimal results in radio networks with collision detection [15, 13]. In $\mathcal{BEEP}$, the wireless network is modeled by a *static communication graph* of diameter D, in which the n nodes represent devices and the edges represent reachability via direct transmission. Time is divided into synchronous steps (i.e., *rounds*), and in each step a node can either listen or transmit a unary signal (beep) to all its neighbors. As a beep is merely a detectable burst of energy, a listening node is not notified about the *identifiers* (IDs) of its beeping neighbors. Even more critically, a beeping node receives no feedback, while a silent (listening) one can only detect whether at least one of its neighbors beeped or all of them were silent. Although algorithms can take advantage of the synchronous nature of the rounds to transmit information using beeps, doing so impacts the time complexity in a quantifiable manner.

* Supported by the Centre for Discrete Mathematics and its Applications (DIMAP) and by EPSRC award EP/N011163/1.

Efficient solutions to fundamental communication primitives provide convenient and efficient abstractions of the actual communication mechanisms and serve as algorithmic building blocks, resulting in simpler algorithm design. Such primitives are even more important in weak communication models, such as $\mathcal{BEEP}$. In this model, simultaneous communications produce interferences, making it difficult for nodes to communicate on a global scale. Importantly, studying how these interferences impact the multi-broadcast problem in $\mathcal{BEEP}$ allows for a better understanding of this problem in stronger models. In the related and well-established radio network model with $O(\log n)$ bit messages and collision detection (a strictly stronger model than $\mathcal{BEEP}$ for which $\mathcal{BEEP}$ algorithms can be straightforwardly translated), the fastest known algorithms were designed in $\mathcal{BEEP}$ [10], and do not use the $O(\log n)$ bits of the messages. If collision detection is not available, the best multi-broadcasting randomized algorithm [2] requires $O(n \log n)$ time while the best deterministic algorithm [5] requires $O(n \log^4 n)$ time.

In the present paper we propose such communication primitives for the tasks of multi-broadcast and gossiping. In multi-broadcast, each *source* node in a subset of at most k (for some integer $k \leq n$) nodes (called *sources*) communicates its message m in $\{1, \ldots, M\}$ and its identifier id in $\{1, \ldots, L\}$ to the whole network (referred to as *multi-broadcast with provenance* in [8]). Gossiping can be seen as a variant of multi-broadcast, in which every node is a source. We present optimal and nearly optimal uniform solutions. Contrary to previous results, these solutions are *constructible*. It is important to emphasize that these results come from an entirely original approach based on *(combinatorial) group testing theory*. Group testing is a method coming from statistics, initially introduced during the Second World War to quickly detect an infection among a group of people [11]. In its original formulation (i.e., *probabilistic group testing*), the defects were assumed to follow some probability distribution, and the goal was to design a strategy identifying all defects using a small expected number of tests. Probabilistic group testing has been used for local neighbor discovery tasks in some distributed settings [19]. In the *combinatorial* context [18, 16], no assumptions are made about the distribution of the defects and the goal is to design a strategy with a small maximum number of tests (i.e., a worst-case scenario). Results from combinatorial group testing are crucial to the current work. They are used to efficiently detect all broadcasting sources, since these can be arbitrary, i.e. cannot be assumed to follow some known probability distribution.

Related Work for Multi-Broadcast. In [8], an $O(D \cdot \log L + k \log \frac{LM}{k})$ round deterministic, completely uniform (in L, D and k) algorithm for k-source multi-broadcast is presented, and the lower bound of $\Omega(D + k \log \frac{LM}{k})$ rounds is given. The multi-broadcast algorithm of [8] also provides an $O(n \log \frac{LM}{n} + D \cdot \log L)$ round solution for gossiping.

In [13], a time-optimal leader election algorithm is given and is used to slightly improve these results: $O(D \cdot \log L)$ factors are reduced to $O(D \cdot \min\{k, \log L\})$ (by executing k consecutive leader elections). Finally, in [10], the lower bound for multi-broadcast given in [8] is extended to also apply to randomized algorithms

and a time-optimal $O(D + k \log \frac{LM}{k})$ deterministic and uniform solution to multi-broadcast is proposed. However, this solution relies on a non-constructive existence proof of a complex combinatorial structure, meaning that it must be pre-computed for each possible set of network parameters, and provided to the network nodes in advance (see discussion below).

Explicitness. Algorithms in $\mathcal{BEEP}$ (and related models such as ad-hoc radio networks) generally seek to minimize the number of rounds required to complete communication tasks. As a result, the cost of local computations is often ignored. Indeed, the fastest deterministic communication algorithms in $\mathcal{BEEP}$, and in radio networks, are often *non-explicit*: they rely upon the use of combinatorial objects whose existence is only proven existentially (see e.g. [9, 10]). Although the existence proofs of the combinatorial objects involved are 'non-constructive', they do imply a naive construction: one can simply generate candidate objects randomly, if shared randomness is available, or in lexicographical order otherwise, and test if they actually satisfy the conditions of the object. However, there are exponentially many possible candidates, and testing naively whether these candidates objects are the required combinatorial objects necessitates an exponential number of computations. Such an approach thus results in an impractically high computation cost.

In some settings an argument can be made that an exponential computation cost may still be acceptable, since the construction of suitable combinatorial objects only *ever* needs to be performed once, and henceforth the object can be stored and provided whenever needed to wireless devices. However, in $\mathcal{BEEP}$ this approach poses a problem: the combinatorial objects that we need depend on the parameters of the network which are not known in advance. Hence, network nodes would have to be pre-loaded with objects for every possible set of parameters. This is again impractical, especially since our aim is to model networks of weak devices which would generally have very limited space.

Consequently, we are only concerned by computationally tractable solutions. In $\mathcal{BEEP}$, *explicit solutions* correspond to algorithms with computation time polynomial in L and k (for the nodes), and *weakly explicit solutions* to algorithms with computation time polynomial in L and exponential in k. The latter can still be computationally feasible if $k << L$ when performing multi-broadcast, and thus of practical interest.

Contributions. First, group testing strategies based on *list disjunct matrices* (see Def. 2) are shown to give efficient solutions for multi-broadcast. Then, several constructions of list disjunct matrices are presented, some novel and some from existing group testing literature, resulting in several algorithms for the multi-broadcasting task:

- An optimal $O(D + k \log \frac{LM}{k})$-time weakly explicit deterministic algorithm.
- An explicit deterministic algorithm optimal for most ranges of k and D.
- An explicit randomized algorithm optimal for $k = \Omega(\log \log L)$.

2 Group Testing

We draw from group testing theory to design efficient solutions in $\mathcal{BEEP}$ (see Section 5). The objective of *group testing* is to identify a subset of defective items in a set, by testing multiple items at a time instead of resorting to individual testing. One example is the christmas tree lighting problem: to search for a broken bulb among a group of six, one can arrange electrically in series three bulbs and apply a voltage. If they light up, then they are in good condition, and the broken bulb is one of the three others. Some classical applications of group testing are blood testing, DNA library screening, signal processing, streaming algorithms and wireless multiple-access communications [12].

Formal Definition. A formal definition of the (d, I)-*combinatorial group testing* (CGT) problem follows. Consider I items, represented by the integers in $\{1, \ldots, I\}$, and any arbitrary subset B of d items. The items in B are said to be *defective*. The only way to differentiate defective items from good (i.e., non-defective) items is through testing. For efficiency reasons, tests consider sets of items (*pools*) instead of individual items. When testing a pool, a positive result (output 1) indicates that at least one item in the pool is defective, whereas a negative result (output 0) indicates that no item in the pool is defective. Tests are considered to be error-free. A solution to the CGT problem is a *group testing strategy*, that is, a sequence of t tests (for some positive integer t) such that the set B can be computed from the results by using a *decoder*. One way of computing B is to use the *naive decoder*: a set B' is initialized to the set of all items (i.e., $\{1, \ldots, I\}$) after which for every negative test result (output 0), the items of the test's pool are removed from B'. It is important to note that the group testing strategy is tightly related to the decoder: more complex decoders could lead to fewer tests.

Explicitness in Group Testing. In group testing literature, testing strategies are devised to identify defective items from a pool, and efforts have been made to minimize the number of tests, and stages of adaptivity, required by the strategies. Again, however, it transpires that the best deterministic strategies rely on existentially-proven combinatorial objects, and so are not efficiently constructible or decodable, by the tester.

Consequently, computationally tractable solutions are sought, for practical reasons. In the group testing literature, an *explicit* strategy is one in which each test sequence can be constructed and the output decoded, in time polynomial in I and d. Also of interest is a weaker notion, which we refer to as *weak explicitness*, where construction and decoding time is polynomial in I and exponential in d. The terminology used here corresponds to that used for multi-broadcast. More precisely, when an explicit (respectively weakly explicit) testing strategy is used to obtain a solution to multi-broadcast, the result is an explicit (resp. weakly explicit) algorithm.

Related Work for Group Testing. In the most frequent setting in group testing, *non-adaptive* (i.e., offline) group testing, all tests are designed offline: a test's

outcome does not influence the following tests. Non-adaptive group testing allows tests to be performed in parallel. However, it was proven in [14] that test strategies in non-adaptive group testing require $\Omega(d^2 \cdot \frac{\log I}{\log d})$ tests. An explicit construction with $O(d^2 \cdot \log I)$ tests for the non-adaptive setting is given in [21]. On the other hand, in a *fully adaptive setting* (i.e., online setting), where each test's pool depends on the results of all previous tests, the information theoretic lower bound implies that test strategies require $\Omega(d \log \frac{I}{d})$ tests, but all tests must be performed sequentially. An optimal fully-adaptive test strategy is given in [16]. Intermediately, *adaptive* group testing refers to multiple *stages* of tests: all tests of a stage are defined independently from the results of the stage, but can depend on the results of previous stages' tests. Thus tests in the same stage can be done in parallel but successive stages must be treated sequentially. Surprisingly enough when compared with non-adaptive group testing, it is possible to construct two-stage test strategies with $\Theta(d \log \frac{I}{d})$ tests [3, 6]. In particular, a weakly explicit construction for such two-stage testing strategies (with $O(d \log \frac{I}{d})$ tests) is given in [6]. Additionally, explicit constructions are given in [4, 17, 20] with a nearly optimal number of tests. In particular, [20] gives an explicit construction for strategies with $O(d^{1+\epsilon} \log I)$ tests for any value $\epsilon > 0$.

3 Model and Definitions

3.1 Definitions

The *communication network* is represented by a simple static connected undirected graph $G = (V, E)$, where V is the node set and E the edge set. The *network size* $|V|$ is denoted by n and the *diameter* by D. Nodes have unique identifiers (IDs). This property is essential in order to break symmetry in deterministic algorithms. The *identifier* of a node $v \in V$, $id(v)$, is an integer from $\{1, \ldots, L\}$ where L is some upper bound on the identifiers unknown to nodes. Then, the *maximum length* over all identifiers in G is $\lceil \log L \rceil$ (also unknown).

We use the terminology of formal language theory. The empty word is denoted by ϵ. The operator $\|$ is for the *word concatenation*. For any positive integer i, 0^i denotes the concatenation of i symbols 0's (where $0^0 = \epsilon$). The *length* of a word x is denoted by $|x|$. For any word x and integer $j \in \{1, \ldots, |x|\}$, $x[j]$ denotes the j^{th} bit of x. For any two words x and y of the same length, we define the (bitwise OR) *superposition* of x and y (and say that x and y are (OR) *superposed*) as the binary word w of length $|w| = |x|$ such that $\forall i \in \{1, \ldots, |w|\}$, $w[i] = 0 \Leftrightarrow x[i] = y[i] = 0$. We naturally extend the superposition to the case of several words of the same length. Additionally, for any two words x and y of the same length, x is said to be *included* in y if $\forall i \in \{1, \ldots, |x|\}$, $x[i] = 1 \Rightarrow y[i] = 1$.

Multi-broadcast. Let S be a subset of k nodes (for some $k > 1$) called *sources* and having (possibly identical) messages in $\{1, \ldots, M\}$, where M is unknown to all nodes. For any node v, $m(v)$ denotes its message. If v is not a source let $m(v) = \epsilon$. Equivalently, $m(v)$ refers to an integer in $\{1, \ldots, M\}$ or to its binary

representation of length at most $\lceil \log M \rceil$.
In the *multi-broadcast* (with provenance) problem, all nodes must receive from each of the k sources its message with its ID. More precisely, they must compute the set $\{(m(v), id(v)) \mid v \text{ is a source }\}$. The *gossiping* problem is a variant of the multi-broadcast problem in which all nodes are sources.

Matrix Notations. For any $a \times b$ matrix M and any integers $i \in \{1, \ldots, a\}$ and $j \in \{1, \ldots, b\}$, the entry of M in row i and column j is denoted by $M[i, j]$. Additionally, the i^{th} row of m is denoted by $M[i, :]$ and the the j^{th} column of m is denoted by $M[:, j]$. For any integer d, let I_d be the $d \times d$ identity matrix, that is, the matrix with entry 1 on the diagonal and 0 otherwise.

3.2 Model Definitions

In the *beeping model* ($\mathcal{BEEP}$), an execution proceeds in synchronous rounds, i.e., there are synchronized local clocks and all nodes start at the same time in a *synchronous start*. In each round nodes synchronously execute the following steps. First, each node beeps or listens. Beeps are transmitted to all neighbors of the beeping node. Then, if a node beeped (in the previous step of the same round), it learns no information from its neighbors. Otherwise, it knows whether or not at least one of its neighbors beeped (during the previous step of the same round). Finally, each node performs local computations. The synchronous start assumption can be replaced by a slightly weaker variant called *wake-on-beep* [1], for an additive factor of $O(D)$ rounds.

4 A General Scheme for Multi-Broadcast

A natural solution for multi-broadcast is as follows. First, a leader node (with the maximum ID) is elected, allowing the network to rely on broadcast and convergecast (respectively, sending a message from and to the leader). Once a leader has been elected, the ID range L is known to all nodes. Relying on communications via the leader, it is now possible to efficiently compute global bounds on the network's diameter D and the message range M. Then, the k sources are identified and ordered, as efficiently as possible, by all nodes. Henceforth, this is referred to as the *source identification component*. Finally, the sources convergecast their messages to the leader (pipelined so that the messages arrive to the leader contiguously in order), and the leader broadcasts the string of messages back through the network. Since all nodes agree on the sources' order, all nodes now have all the messages together with the corresponding IDs of the sources. We outline this scheme in Alg. 1.

Algorithm 1 Multi-Broadcast Scheme

1: Perform Leader Election
2: Estimate Network Parameters
3: Perform Source Identification
4: Collect Source Messages
5: Broadcast Source Messages

All the steps of Algorithm 1, with the exception of Source Identification, can be performed efficiently, explicitly, and deterministically using known procedures from previous works on $\mathcal{BEEP}$:

- Leader election can be performed with $O(D + \log L)$ round complexity [13]. The algorithm requires unique identifiers and elects the node with the maximum identifier. The output is a boolean indicating whether the executing node is the leader or not.
- Estimating diameter D can be performed in $O(D)$ rounds [10]. The algorithm requires a leader, and outputs in all nodes an estimate $\tilde{D}$ with $D \leq \tilde{D} \leq 2D$. Henceforth, we assume that D is known because $\tilde{D}$ can be used instead of D with only a constant-factor overhead.
- Message range M can be similarly estimated in $O(D + \log M)$ time [10].
- Collecting source messages can be done using the CollectMessages procedure from [10]. This procedure requires a leader and upper bounds of D and the maximum length, in bits, of the messages to be collected, denoted by p. It takes as input a set of messages held by nodes in the network. On completion, the leader receives the OR superposition of all the messages, and the running time is $O(D + p)$ rounds.
  We apply this procedure by collecting messages of $p = k\lceil \log M \rceil$ bits, one from each source, in which source numbered i in lexicographical order places its input message into the bit interval $[i\lceil \log M \rceil, (i+1)\lceil \log M \rceil)$, with 0's in every other position (the values of k and the order i are computed during the previously performed source identification component). The superposition of these words is therefore simply the concatenation of all source messages in order. The running time is $O(D + p) = O(D + k \log M)$.
- Broadcasting source messages can be performed using the Beep-Wave procedure of [10]. This procedure allows a leader to broadcast a p-bit message to all nodes in $O(D + p)$ time. Applying the procedure to the concatenation of all k source messages in order yields an $O(D + k \log M)$ time.

All these auxiliary procedures terminate such that nodes start each subsequent procedure synchronously. Consequently, source identification is the only remaining step for which there is no efficient procedure, and it is here that the perspective of group testing allows us to make improvements. We denote the round complexity of a potential source identification algorithm by T_{SI}. Efficient source identification solutions are presented in Section 5 and their round complexities are given by Theorems 5 and 8. Moreover, the scheme for source identification when k is unknown is presented in Section 5.3.

Theorem 1. *Multi-broadcast can be solved in* $O(D + \log L + k \log M + T_{SI})$ *rounds in* $\mathcal{BEEP}$.

Proof. Applying the above procedures to the scheme in Algorithm 1, the total running time of steps 1 and 2 is $O(D + \log L + \log M)$. After these steps, a leader is elected and all nodes know common constant-factor upper bounds for D, L and M. The subsequent procedure for source identification takes T_{SI} rounds, and results in all nodes being aware of all source IDs. Finally, steps 4 and 5 are then correctly performed, completing multi-broadcast in a further $O(D + k \log M)$ rounds. The total running time is therefore $O(D + \log L + k \log M + T_{SI})$.

5 Source Identification and Group Testing

We now show how the problem of source identification can be reduced to that of combinatorial group testing (defined in Section 2). Recall that we have k source nodes with unique IDs from $[L]$, a specified leader node which is known to all nodes in the network, and universal knowledge of (linear upper bounds on) L and D. Upon completing source identification, we require that the leader node has knowledge of all the source IDs (i.e., of S).

Efficient and simple group testing strategies can be obtained by using *list disjunct matrices* (LDM). Such strategies, called LDM-strategies, are presented in Section 5.1 and are the building blocks of the source identification algorithm, described in two stages. First, a simplified scheme (when the number of sources k is known) is presented in Section 5.2. Then an extended scheme for unknown k is presented in Section 5.3. This extended scheme computes a CLDM-strategy (an extension of an LDM-strategy), and its time complexity (resp. computation cost) depends on the CLDM-strategy's parameters (resp., explicitness property). Weakly-explicit and explicit constructions of CLDM-strategies with optimal or nearly optimal parameters are proposed in Section 5.4, resulting in efficiently constructible source identification and multi-broadcast solutions.

5.1 Group Testing Strategies and LDM-strategies

Recall that the (d, I)-combinatorial group testing problem (CGT) consists of finding a subset B of d defective items within a set of I items. Good strategies for CGT use at least 2 stages (see Related work in Section 2). In a two-stage strategy, a first stage determines a subset B_1 of $\{1, \ldots, I\}$ with $B_1 \supset B$ and $|B_1| = \hat{I}$, and the second stage determines a subset B_2 of $\{1, \ldots, \hat{I}\}$ with $B_2 \supset f_1(B)$ and $|B_2| = d$ (where f_1 maps B_1 to $\{1, \ldots, \hat{I}\}$ in lexicographical order).

Definition 1. *Let* B *be some unknown subset of* d *defective items within a set of* $\hat{I}$ *items. A testing strategy using* s *stages and* t *tests over all* s *stages to determine a superset* $B' \supset B$ *of size at most* $d + \ell - 1$ *is called a* $(d, \ell, \hat{I})$ s-stage t-test testing strategy.

In group testing, it is common to build strategies using list disjunct matrices. A single list disjunct matrix defines a single stage testing strategy, and a sequence of s list disjunct matrices defines an s-stage testing strategy (for some integer s).

Definition 2. *A $(d, \ell, \hat{I}, t)$-list disjunct matrix is a $t \times \hat{I}$ binary matrix M such that for any disjoint subsets $T, R \subseteq \{1, \ldots, \hat{I}\}$ with $|T| = d$, $|R| = \ell$, there is a row i of the matrix with $\sum_{j \in T} M[i,j] = 0$ and $\sum_{j \in R} M[i,j] > 0$.*

Lemma 1. *A* $(d, \ell, \hat{I}, t)$-list disjunct matrix *defines a $(d, \ell, \hat{I})$ single stage t-test testing strategy: each row $M[i,:]$ defines the pool of the i^{th} test (for $1 \leq i \leq t$).*

Definition 3. *A* (d, I)-LDM-strategy *using s stages and t tests is a sequence $M_1, \ldots, M_s$ of list disjunct matrices with parameters $(d, \ell_1, I_1, t_1), \ldots, (d, \ell_s, I_s, t_s)$ satisfying:*

- $I_1 = I$,
- $d + \ell_i - 1 = I_{i+1}$ *for all* $1 \leq i < s$,
- $\ell_s = 1$,
- $\sum_{i \leq s} t_i = t$.

Lemma 2. *A (d, I)-LDM-strategy using s stages and t tests is a $(d, 1, I)$ s-stage t-test testing strategy and thus solves (d, I)-CGT.*

If d is known then a (d, I)-LDM-strategy can be computed (see Section 5.4 for some constructions) and this LDM-strategy defines an s-stage t-test testing strategy solving (d, I)-CGT.

5.2 Source Identification for known k

In this section, we give a simplified version (Alg. 2) of the source identification solution, in which we know the number of sources k. This assumption is removed in the extended scheme presented in Section 5.3. Alg. 2 relies on efficient constructions of LDM-strategies (for example, a 2-stage $O(k \log \frac{L}{k})$ weakly explicit LDM-strategy), which are presented later in Section 5.4.

Source Identification Scheme (Alg. 2). The source identification algorithm first computes a (k, L)-LDM-strategy $\mathcal{F}$ using s stages and t tests (which requires knowing k and L), after which sources are identified in s phases. Let $\mathcal{F} = M_1, \ldots, M_s$ where M_u (for $1 \leq u \leq s$) has parameters (k, ℓ_u, L_u, t_u), $L_1 = L$ and $\ell_s = 1$. Details on constructions of good LDM-strategies are deferred to Section 5.4. Using a weakly explicit LDM-strategy results in a weakly explicit source identification solution, and an explicit LDM-strategy in an explicit source identification solution.
Nodes start with no knowledge about which nodes could be the sources, and in each phase they obtain more information by implementing a stage of the group testing strategy defined by $\mathcal{F}$ (see Lemma 2). Let f be initialized to the identity function on $\{1, \ldots, L\}$ in the first phase. The function f is updated so that in every phase u, it renames some of the identifiers in $\{1, \ldots, L\}$ to $\{1, \ldots, L_u\}$ (including all source IDs).

The algorithm executes s phases. In each phase u (for $1 \leq u \leq s$), a node v sets $c_u(v)$ to $M_u[:, f(id(v))]$ (i.e., the $f(id(v))^{th}$ column of M_u) if it is a source, and 0^{t_u} otherwise (see lines 5-6). The superposition w of the words c_u is collected by the leader and then broadcast to all network nodes through the use of the auxiliary functions described in Section 4 (see lines 7-8). Consequently, nodes compute $S_u = \{x \in \{1, \ldots, L_u\} \mid x \text{ is included in } w\}$ and update f (see lines 11-12). More precisely, f is updated to $f_u \circ f$, where f_u renames the elements of S_u to $\{1, \ldots, L_{u+1}\}$ according to their lexicographical order: the y^{th} element of S_u is mapped to y. After all s phases are finished, nodes compute $S = f^{-1}(S_s)$.

Implementation of the testing strategy. Each phase u for $1 \leq u \leq s$ implements the stage u of the testing strategy. Nodes use the tests of stage u to determine some subset S_u of $\{1, \ldots, L_u\}$ which contains $f(S)$ (where $|f(S)| = |S|$ because no defective item is eliminated by the naive decoder, see Section 5.1). Indeed, the leader collects all messages c_u and broadcasts their superposition w to all nodes, which is the superposition of at most k columns of M_u. Each bit $w[i]$ (for $1 \leq i \leq t_u$) can be seen as the test result of test i of stage u in the testing strategy. In the last phase, S_s is a subset of $\{1, \ldots, L_s\}$ with $|S_s| = k + \ell_s - 1 = k$. Therefore, $S_s = f_{s-1} \circ \ldots \circ f_1(S)$.

Algorithm 2 Source Identification Scheme (with known k)

```
1: Inputs: k and upper bounds for L,M and D
2: Compute M_1, ..., M_s and their parameters (k, ℓ_1, L_1, t_1), ..., (k, ℓ_s, L_s, t_s)
3: f := id(v)
4: for phase u := 1 ; u ≤ s ; u++ do
5:     if v is a source node then c_u := M_u[:, f]
6:     else c_u := 0^{t_u}
7:     Collect all binary words c_u by OR superposition into w at the leader
8:     Broadcast the superposition w
9:     Get S_u = {x ∈ {1, ..., L_u} | x is included in w}
10:    if u < s then
11:        Let f_u be a function from S_u to {1, ..., L_{u+1}} in lexicographical order.
12:        if v is a source node then f = f_u(f)
13: Return S = f_1^{-1} ∘ ... ∘ f_{s-1}^{-1}(S_s)        ▷ S is the set of source IDs
```

Theorem 2. *Assume Algorithm 2 computes a (k, L)-LDM-strategy $\mathcal{F}$ using s stages and t tests. Then it solves source identification in $O(Ds + t)$ rounds in $\mathcal{BEEP}$.*

Proof. Algorithm 2 solves source identification since the testing strategy defined by $\mathcal{F}$ correctly identifies all k source nodes. In phase u ($1 \leq u \leq s$), the leader gather binary words of t_u bits from the nodes in $O(D + t_u)$ rounds. Then the leader broadcasts the superposition in $O(D + t_u)$ rounds. Over all s phases, the round complexity is $O(\sum_{u \leq s}(D + t_u)) = O(Ds + t)$ rounds.

Therefore, a good source identification solution should use an LDM-strategy with both small s and small t. The related work in Section 2 describes such strategies. However, these either require high computation cost (i.e., weak explicitness) or non-optimal (but nearly optimal) s and t [20].

5.3 Extending the Source Identification Scheme to unknown k

An extended scheme (of Alg. 2), working when k is unknown, is presented below. The scheme computes an s-stage L-CLDM-strategy (see Def. 5) instead of a (k, L)-LDM-strategy, where the former object is a sequence of constructions that produces an $(\hat{k}, L)$-LDM-strategy for any number of defective items $\hat{k} \leq L$, and can thus be computed when k is unknown. Details on constructions of good CLDM-strategies are deferred to Section 5.4.

Definition 4. *A $(\hat{d}, \hat{I})$-list disjunct matrix construction is a function $\mathcal{C}$ with input $(\hat{d}, \hat{I})$ and output (M, ℓ, t) where M is a $(\hat{d}, \ell, \hat{I}, t)$-list disjunct matrix.*

Definition 5. *A I-*CLDM-strategy *is a sequence $\mathcal{C}_1, \ldots, \mathcal{C}_s$ of constructions of list disjunct matrices satisfying: $\forall \hat{d} \leq I$, let $\mathcal{C}_1(\hat{d}, I) = (M_1, \ell_1, t_1)$ and for $1 < i \leq s$, $\mathcal{C}_i(\hat{d}, I_i) = (M_i, \ell_i, t_i)$ for $I_i = \hat{d} + \ell_{i-1} - 1$, then $M_1, \ldots, M_s$ is a $(\hat{d}, I)$-LDM-strategy.*

Scheme for Source Identification with unknown k. The extended scheme first computes an s-stage L-CLDM-strategy $\mathcal{F}_{\mathcal{C}} = \mathcal{C}_1, \ldots, \mathcal{C}_s$. Following which, sources are identified in s phases, and each phase consists of at most $\lceil \log k \rceil$ subphases. Similarly to Alg. 2, nodes start with no knowledge about which nodes could be the sources, and in each phase u they obtain more information by implementing at most $\lceil \log k \rceil$ consecutive single stage testing strategies on $\{1, \ldots, L_u\}$. Notice that the set of items $\{1, \ldots, L_u\}$ tested upon does not change throughout the different single stage testing strategies (i.e., subphases) of the phase u. Let f be initialized to the identity function on $\{1, \ldots, L\}$ in the first phase. The function f is updated so that in every phase u, it renames some of the identifiers in $\{1, \ldots, L\}$ to $\{1, \ldots, L_u\}$ (including all source IDs).

Subphase Implementation. In sub-phase r of phase u, if $r = 1$ then node v computes $\hat{k}_u^1$, as the smallest power of 2 ($\hat{k}_u^1 = 2^{g_u}$ for some integer g_u) such that $\mathcal{C}_u(\hat{k}_u^1, L_u) = (M_u^1, \ell_u^1, t_u^1)$ satisfies $t_u^1 \geq D$. This prerequisite ensures that the round complexity of phase u in this extended scheme is the same as that in Alg. 2. For any other subphase $r > 1$, node v computes $\hat{k}_u^r = 2^{r-1}\hat{k}_u^1$.
Following which, a node v first computes $\hat{k}_u^r$ and $\mathcal{C}_u(\hat{k}_u^r, L_u) = (M_u^r, \ell_u^r, t_u^r)$. Then, it sets c_u to $M_u^r[:, f(id(v))]$ (i.e., the $f(id(v))^{th}$ column of M_u^r) if it is a source, and 0^{t_u} otherwise. The superposition w of the words c_u is collected by the leader and then broadcast to all network nodes through the use of the auxiliary functions described in Section 4. Then, nodes compute $S_u^r = \{x \in \{1, \ldots, L_u\} \mid x$ is included in $w\}$. If $|B_u^r| \geq \hat{k}_u^r + \ell_u^r$, nodes execute subphase $r+1$ with $\hat{k}_u^{r+1} = 2\hat{k}_u^r$ and still on items $\{1, \ldots, L_u\}$. Otherwise, nodes finish the current phase and if

$u < s$ then nodes execute the following phase $u+1$ with $L_{u+1} = \hat{k}_u^r + \ell_u^r - 1$ (on items $\{1, \ldots, L_{u+1}\}$) and the function f is updated to $f_u \circ f$, where f_u renames the elements of S_u^r to $\{1, \ldots, L_{u+1}\}$ according to their lexicographical order: the y^{th} element of S_u is mapped to y.

The last subphase of a phase implements the only successful single stage testing strategy of the phase. Moreover, if $k_u^r > k$ then the single stage testing strategy defined by M_u^r is guaranteed to return a subset S_u^r of less than $\hat{k}_u^r + \ell_u^r - 1$ items. Consequently, each phase has at most $\lceil \log k \rceil$ subphases.

This method can be used to solve (k, L)-CGT with unknown k, at the cost of a multiplicative factor $\lceil \log k \rceil$ for both stages and tests in comparison to the corresponding (k, L)-LDM-strategy computed when k is known. Fortunately, when CLDM-strategies are used in our source identification solution, this multiplicative factor does not affect the round complexity (see Lemma 3 and Th. 3, whose proofs are deferred to the full version of this paper).

Lemma 3. *Each phase u of the extended source identification scheme takes $R_u = O(\sum_{r \leq r'} t_u^r)$ rounds for $r' = \max\{1, \lceil \log k \rceil - g_u\}$. Let t_u be defined by $\mathcal{C}_u(k, L_u)$. If $\mathcal{C}_u$ satisfies $t_u^1 = O(D)$ and if $r' > 1$, $\sum_{r \leq r'} t_u^r = O(t_u)$, then it follows that $R_u = O(D + t_u)$.*

The conditions of Lemma 3 are satisfied by all 3 CLDM-strategies proposed in Section 5.4. Consequently, the following theorem holds for each:

Theorem 3. *Assume that the s-stage L-CDM-strategy $\mathcal{F}_C$ used in the scheme satisfies Lemma 3 for each phase u ($1 \leq u \leq s$). The extended scheme solves source identification with unknown k in $O(Ds + t)$ rounds, where t is defined by the (k, L)-LDM-strategy computed by $\mathcal{F}_C$ (with $\hat{k} = k$).*

5.4 Efficiently constructible source identification solutions

Various CLDM-strategies resulting in efficient deterministic source identification solutions are presented in this section. Theorem 2 from Section 5.2 emphasizes that both stages and tests should be as low as possible. However strategies with a single stage require a non-optimal $\Omega(d^2 \cdot \frac{\log I}{\log d})$ tests (see Related work in Section 2), thus the CLDM-strategies proposed here have at least 2 stages.
Several constructions of list disjunct matrices are presented, with a trade-off between computational cost and optimal parameters (optimal number of tests). First we give a weakly explicit construction with optimal parameters, resulting in a weakly-explicit (2-stage $O(k \log \frac{L}{k})$-tests) CLDM-strategy and thus a weakly explicit round-optimal source identification solution. Following which, we give two explicit constructions with nearly optimal parameters and use them to construct two different explicit CLDM-strategies. Their combination results in an explicit nearly optimal (optimal for most ranges of D and k) source identification solution.

Lemma 4. *For any integers $\hat{k}, \hat{L}$ with $\hat{L} > \hat{k}$, the identity matrix $I_{\hat{L}}$ is a $(\hat{k}, 1, \hat{L}, \hat{L})$-list disjunct matrix. Thus, there exists a construction function $\mathcal{C}_{Ind}(\hat{k}, \hat{L}) = (I_{\hat{L}}, 1, \hat{L})$ with computation cost poly$(\hat{k}, \hat{L})$.*

The matrix construction $\mathcal{C}_{Ind}$ defines a testing strategy with individuals tests on all $\hat{L}$ items. Although this strategy is not efficient when $\hat{L} >> \hat{k}$, it is very efficient once $\hat{L} = O(\hat{k} \log \frac{\hat{L}}{\hat{k}})$. The challenging part is therefore to reduce L items which could possibly be defective to a 'shortlist' of $\hat{L} = O(k \log \frac{L}{k})$ items.

Weakly explicit construction with optimal parameters. We use an optimal weakly-explicit group testing result from [6]:

Theorem 4 ([6]). *There exists an optimal construction function* $\mathcal{C}_W(\hat{k}, \hat{L}) = (M_W, \hat{k}, O(\hat{k} \log \frac{\hat{L}}{\hat{k}}))$ *with computation cost* $O(\hat{k}^3 \hat{L}^{2\hat{k}+1} \log \hat{L})$.

The CLDM-strategy $\mathcal{F}_1 = \mathcal{C}_W, \mathcal{C}_{Ind}$ is a weakly explicit 2-stage $O(k \log \frac{L}{k})$-test CDLM-strategy. As a side note, $\mathcal{F}_1$ defines what is referred to as a *trivial two-stage testing strategy* in group testing (see Related work in Section 2): $\mathcal{C}_W$ determines most non-defective items, after which $\mathcal{C}_{Ind}$ can be used to determine the k defective items (among the remaining $O(k)$ items). When $\mathcal{F}_1$ is given to the source identification scheme in Section 5.3, the result is a weakly explicit algorithm with optimal round complexity for source identification.

Theorem 5. *The extended source identification scheme using a testing strategy defined by* $\mathcal{F}_1$ *is a weakly explicit algorithm solving source identification in optimal* $O(D+k \log \frac{L}{k})$ *rounds. Consequently, combining this result and the multi-broadcast scheme in Section 4, the result is a weakly explicit algorithm solving multi-broadcast in optimal* $O(D + k \log \frac{LM}{k})$ *rounds.*

Explicit constructions with near optimal parameters. Unfortunately, there are no known explicit constructions for group testing strategies using $O(k \log \frac{L}{k})$ tests and a constant number of stages. As a result, the best known results in group testing [20] do not give optimal multi-broadcast algorithms in $\mathcal{BEEP}$. However, by combining two explicit CLDM-strategies, we can design a multi-broadcast algorithm in $\mathcal{BEEP}$ optimal for most ranges of D and k. For $D >> k \log L$ we can use an existing explicit construction from [20]:

Theorem 6 ([20]). *For any constant* $\epsilon > 0$*, there exists a construction function* $\mathcal{C}_E(\hat{k}, \hat{L}) = (M_E, \hat{k}^{1+\epsilon}, \hat{k}^{1+\epsilon} \log \hat{L})$ *with computation cost* $poly(\hat{k}, \hat{L})$.

For $D << k \log L$ we present a new construction (proof deferred to the full version of this paper):

Theorem 7. *Given integers* $\hat{k}, \hat{L}$ *with* $\hat{L} \geq 2\hat{k}$*, let* q *denote* $\lfloor \log_{2\hat{k}} \hat{L} \rfloor$*. There exists a construction function* $\mathcal{C}_{DIG}(\hat{k}, \hat{L}) = (M_{DIG}, \hat{k}^q, 2\hat{k}q)$ *with computation cost* $poly(\hat{k}, \hat{L})$.

Two explicit CLDM-strategies are presented here:

- The first strategy $\mathcal{F}_2 = \mathcal{C}_E, \mathcal{C}_{Ind}$ is an explicit 2-stage $O(k^{1+\epsilon} \log L)$-test CLDM-strategy. It is, similarly to $\mathcal{F}_1$, a trivial two-stage testing strategy. When the source identification scheme in Section 5.3 uses a testing strategy defined by $\mathcal{F}_2$, the result is an explicit algorithm for source identification with optimal round complexity when $D = \Omega(k^{1+\epsilon} \log L)$.

– The second strategy $\mathcal{F}_3$ is a sequence of $O(\log k \log \frac{\log L}{\log k}) + 1$ constructions, where constructions $\mathcal{C}_i = \mathcal{C}_{DIG}$ for $1 \leq i \leq O(\log k \log \frac{\log L}{\log k})$ and the last construction is $\mathcal{C}_{Ind}$. $\mathcal{F}_3$ is an explicit CLDM-strategy using $O(\log k \log \frac{\log L}{\log k})+$ 1 stages and $O(k \log \frac{L}{k})$ tests. When the source identification scheme in Section 5.3 uses a testing strategy defined by $\mathcal{F}_3$, the result is an explicit algorithm for source identification with optimal round complexity when $D = O(\frac{k \log \frac{L}{k}}{\log k \log \frac{\log L}{\log k}})$.

By executing these two source identification solutions (one defined by $\mathcal{F}_2$, the other by $\mathcal{F}_3$) in parallel (i.e., one round of the first algorithm, then one of the second, and so on), the following result can be obtained.

Theorem 8. *Source identification can be solved using an explicit algorithm with optimal round complexity when either* $D = O(\frac{k \log \frac{L}{k}}{\log k \log \frac{\log L}{\log k}})$ *or* $D = \Omega(k^{1+\epsilon} \log L)$ *(for any constant* $\epsilon > 0$*). As a result, multi-broadcast can be solved using an explicit algorithm with optimal round complexity for most ranges of* k *and* D*.*

6 Explicit Solutions using Randomized Group Testing

While asymptotically optimal explicit 2-stage randomized group testing strategies exist (e.g. constructing a $(\hat{d}, O(\hat{d}), \hat{I}, O(\hat{d} \log \frac{\hat{I}}{\hat{d}}))$ list-disjunct matrix by setting each entry to 1 independently with probability $\Theta(1/\hat{d})$), these strategies are not implementable in our $\mathcal{BEEP}$ framework. This is because they rely on *shared randomness*, i.e. the tester must have access to the randomness used to construct the matrix in order to decode it. However, one practical way to achieve this in $\mathcal{BEEP}$ is to have the leader node generate the random bits to be used, and broadcast them to the network. This will result in a time cost (in rounds) equivalent to the number of the generated random bits. To minimize this cost and obtain an efficient randomized multi-broadcast algorithm in $\mathcal{BEEP}$, we present a new group testing result demonstrating that an optimal testing strategy can be generated using very few random bits:

Theorem 9. *Given* $\hat{d}, \hat{I}$ *with* $\hat{I} \geq 2\hat{d}$*, and* $O(\log \hat{I}(1 + \frac{\log \log \hat{I}}{\log \hat{d}}))$ *independent uniformly random bits, one can construct an explicit* 2*-stage group testing strategy* $\mathcal{F}_P$ *such that for any set* T *of* $\hat{d}$ *defective items, the strategy recovers* T *using* $O(\hat{d} \log \frac{\hat{I}}{\hat{d}})$ *tests and succeeding with high probability* $(1 - 1/poly(\hat{I}))$*.*

This strategy can be used in the same source identification framework as those in section 5, starting with an estimate $\hat{k}$ such that $\hat{k} \log \frac{L}{\hat{k}} = \Theta(D)$, and successively doubling until the algorithm succeeds. The resulting algorithm solves source identification in $O(D + k \log \frac{L}{k} + \log L \log \log L)$ rounds, with high probability (i.e., with probability $(1 - 1/poly(L))$). The proofs of Theorems 9 and 10 are deferred to the full version of this paper.

Theorem 10. *Source identification can be solved in $\mathcal{BEEP}$ with an explicit randomized algorithm in $O(D + k \log \frac{L}{k} + \log L \log\log L)$ rounds, succeeding with high probability. This round complexity is optimal whenever $k = \Omega(\log\log L)$.*

References

1. Afek, Y., Alon, N., Bar-Joseph, Z., Cornejo, A., Haeupler, B., Kuhn, F.: Beeping a maximal independent set. Distributed Computing **26**(4), 195–208 (Aug 2013)
2. Bar-Yehuda, R., Israeli, A., Itai, A.: Multiple communication in multihop radio networks. SIAM Journal on Computing **22**(4), 875–887 (1993)
3. Bonis, A., Gasieniec, L., Vaccaro, U.: Optimal two-stage algorithms for group testing problems. SIAM Journal on Computing **34**(5), 1253–1270 (2005)
4. Cheraghchi, M.: Noise-resilient group testing: Limitations and constructions. Discrete Applied Mathematics **161**(1), 81 – 95 (2013)
5. Chlebus, B.S., Kowalski, D.R., Pelc, A., Rokicki, M.A.: Efficient distributed communication in ad-hoc radio networks. In: ICALP. pp. 613–624 (2011)
6. Cicalese, F., Vaccaro, U.: Superselectors: Efficient constructions and applications. In: ESA. pp. 207–218 (2010)
7. Cornejo, A., Kuhn, F.: Deploying wireless networks with beeps. In: DISC. pp. 148–162 (2010)
8. Czumaj, A., Davies, P.: Communicating with Beeps. In: OPODIS. pp. 1–16 (2016)
9. Czumaj, A., Davies, P.: Deterministic communication in radio networks. SIAM Journal on Computing **47**(1), 218–240 (2018)
10. Czumaj, A., Davies, P.: Communicating with beeps. Journal of Parallel and Distributed Computing (2019)
11. Dorfman, R.: The detection of defective members of large populations. Ann. Math. Statist. **14**(4), 436–440 (1943)
12. Du, D.Z., Hwang, F.K.: Combinatorial Group Testing and Its Applications. World Scientific (1993)
13. Dufoulon, F., Burman, J., Beauquier, J.: Beeping a Deterministic Time-Optimal Leader Election. In: DISC. pp. 20:1–20:17 (2018)
14. D'yachkov, A., Rykov, V., Rashad, A.: Superimposed distance codes. Problems Control Inform. Theory **18**(4), 237–250 (1989)
15. Ghaffari, M., Haeupler, B.: Near optimal leader election in multi-hop radio networks. In: SODA. pp. 748–766 (2013)
16. Hwang, F.K.: A method for detecting all defective members in a population by group testing. Journal of the American Statistical Association **67**(339), 605–608 (1972)
17. Indyk, P., Ngo, H.Q., Rudra, A.: Efficiently decodable non-adaptive group testing. In: SODA. pp. 1126–1142 (2010)
18. Li, C.H.: A sequential method for screening experimental variables. Journal of the American Statistical Association **57**(298), 455–477 (1962)
19. Luo, J., Guo, D.: Neighbor discovery in wireless ad hoc networks based on group testing. In: 46th Annual Allerton Conference on Communication, Control, and Computing. pp. 791–797 (2008)
20. Ngo, H.Q., Porat, E., Rudra, A.: Efficiently decodable error-correcting list disjunct matrices and applications. In: ICALP. pp. 557–568 (2011)
21. Porat, E., Rothschild, A.: Explicit nonadaptive combinatorial group testing schemes. IEEE Transactions on Information Theory **57**(12), 7982–7989 (2011)
22. Vadhan, S.P.: Pseudorandomness. Foundations and Trends in Theoretical Computer Science **7**(1–3), 1–336 (2012)

A Proofs for Section 5

Proof (Proof of Lemma 1). Let M be a $(d, \ell, \hat{I}, t)$-list disjunct matrix. Assume by contradiction that the t-test testing strategy defined by the t rows of M is not a a $(d, \ell, \hat{I})$ single stage t-test testing strategy. Consider the set B' returned by the naive decoder applied on the results of these t tests: B' is initialized at $\{1, \ldots, \hat{I}\}$ and each negative test eliminates all items involved in the tests from B'. Assume by contradiction that $|B'| \geq d + \ell$. Note that the d defective items are never eliminated by the naive decoder and are thus in B'. For the sake of analysis, we can decompose B' into two disjoint subsets, the defective items B and the remaining items R with $|B| = d$ and $|R| \geq \ell$. From the list disjunctness property of M, there is a row i in M such that $\sum_{j \in B} M[i, j] = 0$ and $\sum_{j \in R} M[i, j] > 0$. As a result, the test corresponding to this row is negative and there is a column $j \in R$ such that $M[i, j] = 1$. Consequently, the naive decoder eliminates one of the items in R, hence a contradiction.

Proof (Proof of Lemma 2). Consider a (d, I)-LDM-strategy $\mathcal{F}$ using s stages and t tests. Then $\mathcal{F}$ is a sequence of s list disjunct matrices $M_1, \ldots, M_s$ with parameters $(d, \ell_1, I_1, t_1), \ldots, (d, \ell_s, I_s, t_s)$. By Lemma 1, M_1 defines a single stage t_1-test testing strategy. The naive decoder returns a set B_1 such that the set of defective items $B \subset B_1$ and $|B_1| \leq d + \ell_1 - 1 = I_2$. The items of B_1 are mapped to $\{1, \ldots, I_2\}$ according to their lexicographical order (represented by function f_1). Notice that the defective item set B is mapped to $f_1(B)$ (where $|f_1(B)| = |B|$), and that the subsequent stage seeks to determine a superset B_2 of $f_1(B)$ (and not of B). After which, Lemma 1 is similarly applied to $M_2, \ldots, M_{s-1}$, thus defining functions $f_2, \ldots, f_{s-1}$.
Finally, M_s defines the tests of stage s by Lemma 1 and the naive decoder returns $B_s \subset \{1, \ldots, I_s\}$. Since B_s is a superset of $f_{s-1}(\ldots f_1(B))$ and $|B_s| \leq d + \ell_s - 1 = d$, $B_s = f_{s-1}(\ldots f_1(B))$. Therefore, as $B = f_1^{-1}(\ldots f_{s-1}^{-1}(B_s))$ then an s-stage t-test strategy defined by $\mathcal{F}$ solves (d, I)-CGT.

Proof (Proof of Lemma 3). Consider a phase u (for $1 \leq u \leq s$). The phase takes $R_u = O(\sum_{r \leq r'} t_u^r)$ rounds for $r' = \max\{1, \lceil \log k \rceil - g_u\}$, since in each subphase r (for $1 \leq r \leq r'$), $t_u^r \geq D$ and nodes gather binary words of t_u^r bits at the leader in $O(D + t_u^r) = O(t_u^r)$ rounds, which then broadcasts the superposition in $O(D + t_u^r) = O(t_u^r)$ rounds.

Proof (Proof of Theorem 3). A phase in the extended scheme gives the same correctness guarantees as a phase in Alg. 2. Therefore, correctness of the extended scheme follows from that of Alg. 2.
By Theorem 2, Alg. 2 takes $O(Ds + t)$ rounds. Moreover, by Lemma 3 each phase in the extended scheme has the same round complexity as in Alg. 2 (given some properties on the CLDM-strategy used). Therefore, the extended scheme takes $O(Ds + t)$ rounds.

Proof (Proof of Theorem 5). Consider $\mathcal{F}_1 = \mathcal{C}_W, \mathcal{C}_{Ind}$ and the extended source identification scheme presented in Section 5.3. It is simple to prove that $\mathcal{C}_W$ and

$\mathcal{C}_{Ind}$ satisfy the conditions of Lemma 3. Therefore, we can use Theorem 3 to prove that the extended source identification scheme computing $\mathcal{F}_1$ is a weakly explicit algorithm solving source identification in optimal $O(D + k \log \frac{L}{k})$ rounds.

Proof (Proof of Theorem 7). Write each $j \in [\hat{L}]$ in base $2\hat{k}$, i.e. $j = j_0 j_1 j_2 \dots j_q$, and each digit j_i is an integer between 0 and $2\hat{k} - 1$. For each $x \in [q]$, define the $2\hat{k} \times \hat{L}$ matrix M_x by $M_x[i, j] = 1$ iff $j_x = i$. Then, we let M_{DIG} be the $2\hat{k}q \times \hat{L}$ matrix obtained by vertically concatenating all M_x. We will show that M_{DIG} is a $(\hat{k}, 2^q, \hat{L}, 2\hat{k}q)$-list disjunct matrix.

Let T be a subset of $[\hat{L}]$ with $|T| = \hat{k}$. For each $x \in [q]$, $|DIG_x := \{i : \exists j \in T \text{ with } j_x = i\}| \leq \hat{k}$ i.e. at most $\hat{k}$ different values for digit x are held by the $\hat{k}$ elements of T. For any $j' \in [\hat{L}]$ which has $j'_x \notin DIG_x$, we have $M_x[j'_x, j'] = 1$ and $M_x[j'_x, j] = 0$ for all $j \in T$.
So, for any element j' *not* in the set $DIG := DIG_1 \times DIG_2 \times \dots \times DIG_q$, there is a row in M_{DIG} where j' has value 1 and all elements of T have value 0. DIG is therefore the set of remaining possible defectives, and its size is at most $\hat{k}^q$.

Proof (Proof of Theorem 8). Consider $\mathcal{F}_2 = \mathcal{C}_E, \mathcal{C}_{Ind}$ and the extended source identification scheme presented in Section 5.3. It is simple to prove that $\mathcal{C}_E$ and $\mathcal{C}_{Ind}$ satisfy the conditions of Lemma 3. Therefore, we can use Theorem 3 to prove that the extended source identification scheme computing $\mathcal{F}_2$ is an explicit algorithm solving source identification in nearly optimal $O(D+k^{1+\epsilon} \log L)$ rounds (for any constant $\epsilon > 0$). As a result, if $D = \Omega(k^{1+\epsilon} \log L)$ (for any constant $\epsilon > 0$), then the round complexity above is optimal for source identification.

Similarly, we prove that the extended source identification scheme computing $\mathcal{F}_3$ is an explicit algorithm solving source identification in nearly optimal $O(D \log k \log \frac{\log L}{\log k} + k \log \frac{L}{k})$ rounds. When $D = O(\frac{k \log \frac{L}{k}}{\log k \log \frac{\log L}{\log k}})$, the round complexity above is optimal for source identification.

B Proofs for Section 6

We first show a construction of a testing matrix to be used in our randomized group testing strategy:

Theorem 11. *Given $\hat{d}, \hat{I}$ with $\hat{I} \geq 2\hat{d}$, and $O(\log \hat{I}(1 + \frac{\log \log \hat{I}}{\log \hat{d}}))$ independent uniformly random bits, one can construct an $O(\hat{d} \log \frac{\hat{I}}{\hat{d}}) \times \hat{I}$ matrix such that the matrix eliminates all but $O(\hat{d})$ non-defectives with high probability (i.e. with only $1/poly(\hat{I})$ probability of failure).*

Proof. We will need the following classic results on hashing (see e.g. [22]):

Definition 6. *A family of functions $\mathcal{H}$ mapping $\{1, \dots, N\}$ to $\{1, \dots, M\}$ is ϵ-almost pairwise independent if for every $x_1 \neq x_2 \in \{1, \dots, N\}$, $y_1, y_2 \in \{1, \dots, M\}$, we have*

$$\mathbf{Pr}\left[H(x_1) = y_1 \text{ and } H(x_2) = y_2\right] \leq \frac{1}{M^2} + \epsilon \ .$$

Here the randomness is over uniformly random choice of H from $\mathcal{H}$.

Definition 7. *For $N, M, k \in \mathbb{N}$ with $k \leq N$, A family of functions $\mathcal{G}$ mapping $\{1, \ldots, N\}$ to $\{1, \ldots, M\}$ is k-wise independent if for every distinct $x_1, \ldots, x_k \in \{1, \ldots, N\}$, the values $G(x_1) \ldots, G(x_k)$ are independent and uniformly distributed in $\{1, \ldots, M\}$, when G is drawn uniformly at random from $\mathcal{G}$.*

Theorem 12. *There exists an explicit ϵ-almost pairwise independent family $\mathcal{H}$ of functions $H : \{1, \ldots, N\} \to \{1, \ldots, M\}$ such that any $H \in \mathcal{H}$ can be specified using $O(\log \log N + \log M + \log \frac{1}{\epsilon})$ bits.*

Theorem 13. *There exists an explicit k-wise independent family $\mathcal{G}$ of functions $G : \{1, \ldots, N\} \to \{1, \ldots, M\}$ such that any $G \in \mathcal{G}$ can be specified using $O(k \log NM)$ bits.*

(We omit some details here such as requiring the domain and range of the functions to be integer powers of 2, but since we are concerned with asymptotic complexity, this does not affect the results).

We use these functions to minimize the amount of random bits necessary for our construction. Let c be a sufficiently large constant. We also require that $\hat{I}$ and $\hat{d}$ are sufficiently large, but again this does not affect asymptotic results.

Let G be an explicit $\frac{4 \log \hat{I}}{\log \hat{d}}$-wise independent family of functions $\{1, \ldots, O(\log \frac{\hat{I}}{\hat{d}})\} \to \{1, \ldots, 2^{O(\log \hat{d} + \log \log \hat{I})}\}$ described by Theorem 13, with functions specified by $O(\frac{4 \log \hat{I}}{\log \hat{d}} \log(\log \frac{\hat{I}}{\hat{d}} \cdot 2^{O(\log \hat{d} + \log \log \hat{I})})) = O(\log \hat{I}(1 + \frac{\log \log \hat{I}}{\log \hat{d}})$ bits.

Let H be an explicit $\frac{1}{\hat{d}^3}$-almost pairwise independent family of functions $h : \{1, \ldots, \hat{I}\} \to \{1, \ldots, c\hat{d}\}$, with functions specified using $O(\log \log \hat{I} + \log c\hat{d} + \log \hat{d}^3) = O(\log \log \hat{I} + \log \hat{d})$ bits.

Using the $O(\log \hat{I}(1 + \frac{\log \log \hat{I}}{\log \hat{d}}))$ random bits provided to the algorithm, select a random function $g \in G$. Then, for $x \in \{1, \ldots, c \log \frac{\hat{I}}{\hat{d}}\}$, let $h_x : \{1, \ldots, \hat{I}\} \to \{1, \ldots, c\hat{d}\}$ be the function from H specified by the $O(\log \log \hat{I} + \log \hat{d})$-bit string $g(x)$. We then define a $c\hat{d} \times \hat{I}$ matrix M_x by $M_x[i, j] = 1$ iff $h_x(j) = i$. Finally, let our testing matrix M be the $c^2 \hat{d} \log \frac{\hat{I}}{\hat{d}} \times \hat{I}$ matrix obtained by vertically concatenating all M_x.

Let T be our arbitrary set of defective items, i.e. a subset of $\{1, \ldots, \hat{I}\}$ with $|T| = \hat{d}$.

For each $x \in \{1, \ldots, c \log \frac{\hat{I}}{\hat{d}}\}$, let S_x be the set of non-defective items which are not eliminated by a matrix M_y with $y < x$ (i.e. S_x is the set of all items $j' \in \{1, \ldots, \hat{I}\} \setminus T$ such that there is no $y < x$ and $i \leq t$ with $M_y[i, j'] = 1$ and $M_y[i, j] = 0 \ \forall j \in T$). The $S_{x+1} \setminus S_x$ is the set of all items which *are* eliminated by matrix M_x.

Clearly $S_1 = \{1, \ldots, \hat{I}\} \setminus T$. We now wish to show that for any $x \in \{1, \ldots, c \log \frac{\hat{I}}{\hat{d}}\}$, the probability that $|S_{x+1}| > \frac{|S_x|}{2}$ is at most $\frac{9}{c\hat{d}}$.

Fix some $x \in \{1, \ldots, c \log \frac{\hat{I}}{\hat{d}}\}$. We assume that $|S_x| \geq c\hat{d}$, since otherwise we have already eliminated sufficient items. For $j \in S_x$, denote by $\mathbf{1}_j$ the indicator variable that $j \notin S_{x+1}$, i.e. that j is eliminated by matrix M_x. Notice that by symmetry, these $\mathbf{1}_j$ are identically distributed for all $j \in S_x$ (though they are not independent, or even pairwise independent). Denote the expectation of these indicator variables by μ.

We first bound μ from below. For any item j in S_x, the probability that all elements $j' \in T$ have $h_x(j') \neq h_x(j)$ (i.e. have value 0 on row $h_x(j)$ is bounded by:

$$\begin{aligned}
\mathbf{Pr}\left[\bigcap_{j' \in T} \{h_x(j') \neq h_x(j)\}\right] &\geq 1 - \sum_{j' \in T} \mathbf{Pr}\left[\{h_x(j') = h_x(j)\}\right] \\
&\geq 1 - \hat{d} \cdot \left(\frac{1}{c\hat{d}} + \frac{1}{\hat{d}^2}\right) \\
&\geq \frac{c-2}{c} \ ,
\end{aligned}$$

where the initial inequality follows from a union bound and the first equality by $\frac{1}{\hat{d}^3}$-almost pairwise independence of h_x. In this event $\mathbf{1}_j = 1$, so $\mu \geq \frac{c-2}{c}$. By linearity of expectation, $\mathbf{E}\left[|S_x \setminus S_{x+1}|\right] = \sum_{j \in S_x} \mathbf{E}\left[\mathbf{1}_j\right] = \mu |S_x|$.

We must now show a concentration bound on $|S_x \setminus S_{x+1}|$. To do so, we will need the following lemma:

Lemma 5. *For any $i \neq j \in S_x$, $\mathbf{E}\left[\mathbf{1}_i \mathbf{1}_j\right] \leq \frac{\mu}{2\hat{d}} + \mu^2$.*

Proof.

$$\begin{aligned}
\mathbf{E}\left[\mathbf{1}_i \mathbf{1}_j\right] &= \mathbf{Pr}\left[\mathbf{1}_i = \mathbf{1}_j = 1\right] \\
&= \mathbf{Pr}\left[\mathbf{1}_i = \mathbf{1}_j = 1 \wedge h_x(i) = h_x(j)\right] + \mathbf{Pr}\left[\mathbf{1}_i = \mathbf{1}_j = 1 \wedge h_x(i) \neq h_x(j)\right] \\
&\leq \mathbf{Pr}\left[\mathbf{1}_i = 1 \wedge h_x(i) = h_x(j)\right] + \mathbf{Pr}\left[\mathbf{1}_i = \mathbf{1}_j = 1 \mid h_x(i) \neq h_x(j)\right] \\
&\leq \mathbf{Pr}\left[\mathbf{1}_i = 1\right] \mathbf{Pr}\left[h_x(i) = h_x(j)\right] + \mu^2 \\
&= \mu\left(\frac{1}{c\hat{d}} + \frac{1}{\hat{d}^2}\right) + \mu^2 \leq \frac{\mu}{(c-1)\hat{d}} + \mu^2 \ .
\end{aligned}$$

We use this bound on the correlation of the indicator variables $\mathbf{1}_j$ to bound the variance of their sum:

Lemma 6. $\mathbf{Var}\left[\sum_{j \in S_x} \mathbf{1}_j\right] \leq \frac{2\mu |S_x|^2}{c\hat{d}}$.

Proof.

$$\begin{aligned}
\mathbf{Var}\left[\sum_{j\in S_x}\mathbf{1}_j\right] &= \mathbf{E}\left[\left(\sum_{j\in S_X}\mathbf{1}_j - \mathbf{E}\left[\sum_{j\in S_x}\mathbf{1}_j\right]\right)^2\right] \\
&= \mathbf{E}\left[\left(\sum_{j\in S_x}(\mathbf{1}_j-\mu)\right)^2\right] \\
&= \sum_{i,j\in S_x}\mathbf{E}\left[(\mathbf{1}_i-\mu)(\mathbf{1}_j-\mu)\right] \\
&= \sum_{j\in S_x}\mathbf{E}\left[(\mathbf{1}_j-\mu)^2\right] + \sum_{i\neq j\in S_x}\mathbf{E}\left[(\mathbf{1}_i-\mu)(\mathbf{1}_j-\mu)\right] \\
&= |S_x|\mu(1-\mu) + \sum_{i\neq j\in S_x}\mathbf{E}\left[\mathbf{1}_i\mathbf{1}_j - \mu(\mathbf{1}_i+\mathbf{1}_j)+\mu^2\right] \\
&\leq \frac{2}{c}|S_x|\mu + \sum_{j_1\neq j_2\in S_x}\left(\frac{\mu}{(c-1)\hat{d}} + \mu^2 - 2\mu^2 + \mu^2\right) \\
&\leq \frac{2\mu|S_x|^2}{c^2\hat{d}} + \frac{\mu|S_x|^2}{(c-1)\hat{d}} \\
&\leq \frac{2\mu|S_x|^2}{c\hat{d}} \ .
\end{aligned}$$

Here the first inequality used Lemma 5, and the second used that we are assuming $|S_x| \geq c\hat{d}$.

Now that we have a bound on the variance of $\sum_{j\in S_x}\mathbf{1}_j$, we simply apply Chebyshev's inequality to obtain

$$\mathbf{Pr}\left[\left|\sum_{j\in S_x}\mathbf{1}_j - \mu|S_x|\right| \geq \epsilon\right] \leq \frac{\mathbf{Var}\left[\sum_{j\in S_x}\mathbf{1}_j\right]}{\epsilon^2} \leq \frac{2\mu|S_x|^2}{c\hat{d}\epsilon^2} \ .$$

Setting $\epsilon = \frac{\mu|S_x|}{2}$ yields:

$$\mathbf{Pr}\left[\sum_{j\in S_x}\mathbf{1}_j \leq \frac{\mu|S_x|}{2}\right] \leq \frac{8}{c\mu\hat{d}} \leq \frac{8}{(c-2)\hat{d}} \leq \frac{9}{c\hat{d}} \ .$$

So, with probability at least $1-\frac{9}{c\hat{d}}$, we have $|S_{x+1}| \leq |S_x| - \frac{\mu|S_x|}{2} = \frac{\mu|S_x|}{2}$ as required.

We will call any x with $|S_{x+1}| > \frac{\mu|S_x|}{2}$ *bad*. The random strings used to construct each matrix M_x are $\frac{c\log\hat{I}}{\log\hat{d}}$-wise independent, hence so are the events that each x is bad. Therefore,

$$\begin{aligned}
\mathbf{Pr}\left[\text{at least } \frac{4\log \hat{I}}{\log \hat{d}} \text{ values } x \text{ are bad}\right] &\leq \binom{c\log \frac{\hat{I}}{\hat{d}}}{\frac{4\log \hat{I}}{\log \hat{d}}} \left(\frac{9}{c\hat{d}}\right)^{\frac{4\log \hat{I}}{\log \hat{d}}} \\
&\leq \left(\frac{ce\log \frac{\hat{I}}{\hat{d}}}{\frac{4\log \hat{I}}{\log \hat{d}}}\right)^{\frac{4\log \hat{I}}{\log \hat{d}}} \left(\frac{9}{c\hat{d}}\right)^{\frac{4\log \hat{I}}{\log \hat{d}}} \\
&\leq \left(\frac{9e\log \hat{d}}{4\hat{d}}\right)^{\frac{4\log \hat{I}}{\log \hat{d}}} \\
&\leq 2^{-\frac{4\log \hat{I}}{\log \hat{d}}\log\sqrt{\hat{d}}} \\
&= 2^{-2\log \hat{I}} \\
&= \hat{I}^{-2} \ .
\end{aligned}$$

So, with high probability, at most $\frac{4\log \hat{I}}{\log \hat{d}}$ values x are bad, i.e. at least $\frac{c}{2}\log\frac{\hat{I}}{\hat{d}}$ are not. For any x which is not bad, M_x eliminates at least a half of the remaining non-defective items. Then, the number of items which are not eliminated by the concatenated matrix M is at most

$$\left(\frac{1}{2}\right)^{\frac{c}{2}\log\frac{\hat{I}}{\hat{d}}} |S_1| \leq \hat{I}\cdot 2^{-\frac{c}{2}\log\frac{\hat{I}}{\hat{d}}} \leq \hat{I}\cdot 2^{-\log\frac{\hat{I}}{\hat{d}}} \leq \hat{d} \ .$$

That is, at most $\hat{d}$ non-defective items remain.

We can now easily describe our two-stage testing strategy:

Proof (Proof of Theorem 9). In stage 1, use the construction from Theorem 11 to rule out all but $O(\hat{d})$ non-defective items, using $O(\hat{d}\log\frac{\hat{I}}{\hat{d}})$ tests. In stage 2, test all of the remaining items individually, using $O(\hat{d})$ tests. The probability that both stages succeed is at least $1-\hat{I}^{-2}$.

Finally, we describe how to implement this strategy for source identification in $\mathcal{BEEP}$.

Proof (Proof of Theorem 10). To perform source identification, the leader node generates $O(\log L\log\log L)$ independent uniformly random bits, and broadcasts them to all nodes in $O(D+\log L\log\log L)$ rounds. This is sufficient randomness to perform the group testing strategy of Theorem 9 with any $\hat{d}$ (and $\hat{I}=L$). Then, we initially set $\hat{k}$ such that $\hat{k}\log\frac{L}{\hat{k}} = D$ (or $\hat{k}=1$ if $D<\log L$). We repeatedly perform the group testing strategy of Theorem 9, doubling $\hat{k}$ until it successfully identifies all sources. By the argument of Theorem 3, this takes only $O(D+k\log\frac{L}{k}+\log L\log\log L)$ total rounds. Furthermore, since we perform

at most $\log k$ iterations of the group testing strategy, the probability that they all execute correctly (and therefore our overall probability of success) is at least $1 - \frac{\log k}{L^2} \geq 1 - \frac{1}{L}$ by a union bound.