

Beep-And-Sleep: Message and Energy Efficient Set Cover

Thorsten Götte ✉

Paderborn University, Germany

Christina Kolb ✉

University of Twente, The Netherlands

Christian Scheideler ✉

Paderborn University, Germany

Julian Werthmann ✉

Paderborn University, Germany

Abstract

We observe message-efficient distributed algorithms for the SETCOVER Problem. Given a ground set U of n elements and m subsets of U , we aim to find the minimal number of these subsets that contain all elements. In the default distributed distributed setup of this problem, each set has a bidirected communication link with each element it contains. Our first result is a $\tilde{O}(\log^2(\Delta))$ -time and $O(\sqrt{\Delta}(n+m))$ -message algorithm with expected approximation ratio of $O(\log(\Delta))$ in the KT_0 model. The value Δ denotes maximal cardinality of each subset. Our algorithm is *almost* optimal with regard to time and message complexity. Further, we present SETCOVER algorithm in the BEEPING model that only relies on carrier-sensing and can trade runtime for approximation ratio similar to the celebrated algorithm by Kuhn and Wattenhofer [PODC '03].

2012 ACM Subject Classification Theory of computation \rightarrow Distributed algorithms; Mathematics of computing \rightarrow Probabilistic algorithms

Keywords and phrases Set Cover, Approximation Algorithms, Beeping Model, KT_0 Model

1 Introduction

SETCOVER is a well-understood problem in both the centralized and distributed setting. Given a collection of elements $\mathcal{U} := \{e_1, \dots, e_n\}$ and sets $\mathcal{S} := \{s_1, \dots, s_m\}$ with $s_i \subseteq \mathcal{U}$ the goal is to cover all elements with as few sets as possible. SETCOVER has a wide variety of applications in many areas of computer science. On the one hand, it plays an essential role in the analysis of large data sets, which is often needed in fields like operations research, machine learning, information retrieval, and data mining (see [5] and the references therein). On the other hand, it is also used in purely distributed domains like ad-hoc sensor networks. An essential task in these networks is to determine a minimum set of nodes, a so-called DOMINATINGSET, such that all nodes are sensor range of this set. This set can then fulfill particular tasks like routing, collecting sensor data from neighbors, and various other tasks. Note that DOMINATINGSET is a special case of SETCOVER where all sets are also elements.

In the centralized setting, a simple greedy algorithm that always picks the set that covers most elements has a logarithmic approximation factor, which is the best we can hope for a polynomial-time algorithm unless $P = NP$. There are several randomized algorithms in the distributed CONGEST model that match the optimal approximation ratio, w.h.p., and have a near-optimal runtime of $O(\log(\Delta)^2)$ where Δ is the maximum degree of a set [6, 8]. For distributed algorithms, an instance of SETCOVER is usually modeled as a so-called *problem graph* $G_P := \{V_{\mathcal{U}} \cup V_{\mathcal{S}}, E\}$. Each set and each element corresponds to a node in this graph and for each set $s_i \in \mathcal{S}$ there is a bidirected edge $\{e_j, s_i\} \in E$ to each element $e_j \in \mathcal{U}$ it contains. Each of these edges models a bidirected communication channel between the nodes representing the set and element in the default distributed setup. Each set and element in each round of communication can send a distinct

message of size $O(\log(n))$ over each communication edge. Although there are numerous results concerning the runtime of algorithms in the CONGEST model, there are only a few that consider the message complexity, i.e., how many messages are needed to compute a distributed solution to SETCOVER. To the best of our knowledge, in all popular solutions to set cover, all sets/elements extensively communicate with their neighborhood and send (possibly distinct) messages to all of their neighboring set and elements. This setup implies a message complexity of $O(|E|)$ per round and $\tilde{O}(|E|)$ overall. From a practical standpoint, a high message complexity makes these algorithms less suitable for dense networks where sending a single message is considered costly. For example, in ad-hoc networks, one needs to take special care of the limited battery life of the nodes. In particular, sending messages should be reduced to a minimum due to the high energy requirements of the radio module. Thus, a distributed message-efficient algorithm for SETCOVER can therefore greatly reduce this energy consumption. Therefore, our first question is:

(Q.1) *Can we find a SETCOVER algorithm that sends $o(|E|)$ messages?*

However, even if we find message-efficient algorithms in the CONGEST model, they do not necessarily translate well to ad-hoc networks where they are needed. In practical ad-hoc networks, however, it is much more energy-efficient to only send a single signal, a so-called BEEP, to all neighboring nodes. Further, nodes can only distinguish if at least one or none of their neighbors beeped, i.e., they only rely on carrier sensing. This has recently been formalized in the so-called BEEPING model[1]. To the best of our knowledge there no SETCOVER or DOMINATINGSET algorithm for general graphs in this model. Thus, our second question is:

(Q.2) *Can we find a SETCOVER algorithm in the BEEPING model?*

We answer both questions affirmative and provide fast and efficient algorithms for SETCOVER in the KT_0 -model (a variant of the CONGEST model for analyzing message-efficient algorithms) and the BEEPING model. Our KT_0 algorithm has polylogarithmic runtime and approximation ratio while sending only $\tilde{O}(\sqrt{\Delta}(n+m))$ ¹ messages, w.h.p.². Although we cannot prove that this message complexity is optimal for this particular approximation ratio, we give evidence that one cannot hope for far better results as we show that there are instances that require $O(\sqrt{\Delta}n)$ messages for a constant approximation. Second, our BEEPING algorithm runs in $O(k^3)$ rounds and has an approximation ratio of $O(\log(\Delta)^{2/k}\sqrt{\Delta})$ and thus – similar to [8] – can trade runtime for approximation ratio. This tradeoff makes it particularly practical as the topology of sensor networks is often rapidly changing, and our algorithm can quickly react to these changes by computing a new solution in constant time. Existing solutions ([12, 13]) have a far better approximation ratio (as they are designed for UDGs), but have at least logarithmic runtime, which cannot trivially be reduced. Note that the focus of this work is to prove feasibility and not optimize logarithmic factors in message complexity and approximation ratios, i.e., we explicitly do not claim our factors to be optimal.

Our main technical contribution is an adapted version of Jia et al.’s DOMINATINGSET algorithm from [6] that will be the basis for both our contributions. We adapt this algorithm in two major ways: First, we massively reduce the messages that need to be exchanged by replacing all instances where nodes are counted through randomized approximation that only use a fraction of the messages *and* – perhaps more importantly – completely replacing the mechanism that lets nodes decide to join the solution. For the latter, we use geometrically distributed starting times where nodes add

¹ $\tilde{O}(\cdot)$ hides polylogarithmic factors.

² As usual, an event holds with high probability, if it holds with prob. $1 - o(n^{-c})$ for some $c > 1$

themselves when their time has come, and they have a certain threshold of uncovered neighbors. This approach does not require any additional messages. Note that the independent work of [4] which only recently came to our attention, uses a very similar technique³, but their analysis does not provide all the properties we need for our problems, at least not without non-trivial extensions.

This paper is structured as follows: First, we present some related work in the remainder of this section. Then, we present the algorithm for the BEEPING model in Section 3 and in Section 4 the message-efficient KT_0 algorithm and an almost matching lower bound. We provide more details on the models in the respective sections.

1.1 Related Work

There are only a few works on the message complexity of SETCOVER, however the space complexity of SETCOVER in streaming models is an active research topic with exciting results for several variants of the problem. In the streaming model, the edges of the problem graph, i.e., the information on which element belongs to which set, arrive sequentially. An algorithm can store each edge for later use and iterate over the whole input several times. The goal is to solve SETCOVER using as few passes and as few space as possible. Note that a trivial algorithm can just iterate over the input once, store all edges, and then solve the problem using an *offline* algorithm. Demaine et al. further showed that randomization is crucially needed for a space complexity of $o(mn)$ as it is impossible for a deterministic algorithm. In addition to this distributed model, Indyk et al. also successfully used the ideas and primitives from streaming algorithms in the area of Sublinear Algorithms [5]. The goal is to find an algorithm that makes as few queries as possible and computes a good approximate solution. As we will see, this model has a lot in common with our distributed model. In this case, the number of queries should be $o(mn)$, i.e., sublinear in the number of edges. Indyk et al. present a polynomial-time algorithm that makes $\tilde{O}(m + \sqrt{mn})$ queries and computes a $O(\log^2(n))$ -approximate solution. With exponential runtime, the approximation ratio even reduces to $O(\log^2(n))$, but the algorithm still needs to make $\tilde{O}(m + \sqrt{mn})$ queries. Furthermore, Indyk et al. gave strong evidence that $\Omega(nm^\epsilon)$ queries are indeed necessary to achieve good approximation ratios. Finally, we review some fully distributed approaches for SETCOVER and related Optimization Problems. As mentioned in the introduction Jia et al.[6] as well as Kuhn and Wattenhofer[8] have presented fast distributed algorithms to solve the DOMINATINGSET problem in time in $O(\log(n)^2)$. The goal is to find the minimal set of nodes adjacent to all nodes in the graph $G := (V, E)$. DOMINATINGSET is a special case of SETCOVER where each node is both an element and a set. Kuhn et al. further presented an algorithm to solve SetCover (and any covering and packing LP) in $O(\log(n)^2)$ rounds in the CONGEST model[7]. This is close to the optimal runtime of $O(\log(n))$, which is also proven in that paper. Roughgarden et al. presented a distributed algorithm for convex optimization problems (which includes SETCOVER) that works on any eventually connected communication network [10]. In particular, it works on dynamic communication networks that can change from round to round. Even though this algorithm is fully distributed, it heavily relies on sequential calculations to cope with these general communication networks. Therefore, its runtime and message complexity are polynomial in the number of nodes. Finally, many works consider the DOMINATINGSET-problem in models tailored to ad-hoc networks. First, there is a paper by Scheideler et al.[12] that observes the SINR model where the nodes are modeled as points in the euclidean plane. Finally, there already is a solution to the DOMINATINGSET-problem in the BEEPING-model for unit-disk graphs[13]. Their algorithm bears some similarities with ours but is not applicable to general graphs.

³ Instead of using geometric starting times they continuously increase a set's probability to join until it joins or does cover enough elements. From a probabilistic point of view this is (almost) equivalent to picking geometric starting times.

2 Preliminaries

In addition to some textbook probability distributions, in this work, we use the following version of the Chernoff Bound:

► **Lemma 1** (Chernoff Bound). *Let $X := \sum X_i$ be the sum of independent random variables with $X_i \in \{0, 1\}$. Then it holds that for any $0 < \delta$,*

$$\Pr[X \geq (1 + \delta)\mathbf{E}[X]] \leq e^{-\frac{\delta^2 \mathbf{E}[X]}{3}} \quad \text{and} \quad \Pr[X \leq (1 - \delta)\mathbf{E}[X]] \leq e^{-\frac{\min\{\delta, \delta^2\} \mathbf{E}[X]}{3}}. \quad (1)$$

3 An Efficient SETCOVER-Algorithm for the BEEPING-Model

We will now describe our first algorithm, which we dub the BEEP-AND-SLEEP algorithm, as most sets and elements will be idle during the execution. Before we go into the details of our result and the algorithm, let us first present the model. In this section, we use the following (standard) variant of the BEEPING model [1, 2]:

1. We observe a fixed communication graph $G := (V_S \cup V_U, E)$ with $V_S = n$ and $V_U = m$. Each set $s \in V_S$ has a bidirected edge $\{s, e\} \in E$ to each element $e \in V_U$ it contains. Each node can only communicate with its neighbors in G . Further, all nodes know Δ , the maximal degree of G . This assumption can be replaced by a polynomial upper bound, which would slow the algorithm down by a constant factor. Note that the nodes do not know their exact degree, and nodes have *no* identifiers.
2. Time proceeds in so-called *slots*. In each slot, a node can either *beep* or *listen*. If a node *listens* and any subsets of its neighbors *beeps*, the *listening* node receives a BEEP. It can neither distinguish which neighbors beeped nor how many neighbors beeped, i.e., it only relies on carrier sensing. Further, a node *cannot* simultaneously beep and listen but must choose one of the two options.
3. All nodes wake up in the same slot, i.e., we observe the BEEPING model with simultaneous wake-up. We believe that our algorithm can also be extended for arbitrary wake-up as each node only needs to be in sync with neighboring nodes. If nodes do not wake up in the same round, their internal counters only differ by 1 as each node wakes up one slot after its earliest neighbor. In this case, there are some standard tricks to simulate a single slot of a simultaneous wake-up algorithm within 3 slots [1, 2].

Given this model, we show the following:

► **Theorem 2.** *There is an algorithm in the BEEPING model that solves SETCOVER in time $O(k^3)$ with approximation ratio $O(\log^2(\Delta) \cdot \sqrt[k]{\Delta^3})$ where $k > 3$ is parameter known to all nodes.*

Thus, even for a constant k , our algorithm achieves a non-trivial approximation ratio, which is close to the optimal one as Kuhn et al. showed that any distributed algorithm with only local communication needs $O(k)$ for an approximation ratio of $O(\sqrt[k]{\Delta})$ [8]. We will now describe the algorithm promised by the theorem: The core idea is that all nodes that cover the *most* neighbors add themselves to the set similar to the sequential greedy algorithm. The main difficulty stems from the fact that we need to estimate the number of uncovered neighbors correctly and to avoid that too many sets that cover the same elements concurrently add themselves to the solution. The algorithm runs in k phases, where in phase i all sets that cover *approximately* $\Delta_i = \frac{\Delta}{\sqrt[k]{\Delta^i}}$ elements try to add themselves to the solution. Each phase is structured in $4k$ rounds. Further, each round again consists of $4k + 1$ slots, which brings the total runtime to $O(k^3)$. The pseudocode for the algorithm is given in Figure 1, on a high level, it works as follows:

1. At the beginning of *phase j*, i.e., before the first round of that phase, the sets and elements do the following: Each *set s* draws a geometric random variable X_s with parameter $1 - 1/\sqrt[k]{\Delta}$. Values bigger than $4k$ are rounded down to $4k$, so $X \in [0, 4k]$ always. Then, the set waits for $4k - X_s$ *rounds* and neither beeps or listens, it just stays idle. Each uncovered *element* marks itself as *active* with probability $\frac{4k}{\Delta_j} := \frac{4k \sqrt[k]{\Delta^j}}{\Delta}$. Thus, each set with Δ_j (or more) uncovered elements has at least $4k$ active elements in expectation. Each active element further picks a slot number $a_u \in [0, 4k]$ uniformly at random.
2. In each *round i* of phase *j*, the sets and elements do the following: Each *set s* that wakes up in this round listens to BEEPS for the first $4k$ slots of this round. Further, it counts all slots in which it received a BEEP. Otherwise, it remains idle for this round. Each active *element* that has not yet been covered, beeps *only* in the slot a_u it has drawn in the beginning and remains idle otherwise. Note that, by our choice of active elements, this implies that all sets that have at least Δ_j uncovered elements, therefore, hear one BEEP per slot *in expectation*.
3. In the *last* slot of a round, i.e., in slot $4k + 1$, the sets and elements do the following: Each *set* that received a BEEP in at least $3k$ slots adds itself to the solution and beeps. Each uncovered *element* (active or not) listens and considers itself covered if a neighbor joined the solution, i.e., if it hears a BEEP. If an element is covered, it does not need to send or receive messages for the remainder of the algorithm.

Note that the last step of each round where the sets add themselves is the main difference to the classical algorithm by Jia et al. [6]. In their work, each candidate locally computed a probability to join the solution. This used information from its 2-neighborhood. This approach seems not to be trivially doable with only BEEPS. Finally, note that this algorithm can also be used to compute a DOMINATINGSET, we provide the details at the end of this section.

3.1 Analysis

We will now prove Theorem 1 and show that our algorithm indeed fulfills the promised bounds. Since the runtime is deterministic, we will only prove the expected approximation ratio.

► **Lemma 3.** *The algorithm outputs a $O\left(\log^2(\Delta) \cdot \left(\sqrt[k]{\Delta}\right)^3\right)$ -approximate solution in expectation.*

Since the algorithm, for the most part, follows the LRG algorithm by Jia et al.[6], a generalized version of their core lemma also holds in this case. However, we need to "parameterize" it further to account that we cannot precisely count the uncovered elements. For each element covered by our algorithm, we define the random variables $\eta(u)$ and $\mu(u)$. We define $\eta(u)$ to be ratio between the *best* set in u 's neighbor and the worst set picked by the algorithm, i.e., the ratio by which the choice of our algorithm differs from the greedy solution. Further, let $\mu(u)$ be the random variable that denotes the number of candidates covering $u \in V$ in the round where it is first covered. Finally, we define η and μ the corresponding variables of the elements that maximize their expected values, i.e.,:

$$E[\mu] = \max_{u \in U} E[\mu(u)] \quad \text{and} \quad E[\eta] = \max_{u \in U} E[\eta(u)] \quad (2)$$

Given these definitions, the following modified version of Jia et al.'s algorithm[6] holds:

► **Lemma 4.** *Let S be the size of the set output by our algorithm. Then, it holds.*

$$E[S] \leq E[\eta] \cdot E[\mu] \cdot \log(\Delta) \cdot |S_{OPT}| \quad (3)$$

■ **Algorithm 1** BEEP-AND-SLEEP (BASED ON [6])

```

procedure SETS( $s, N_s, k, \Delta$ )                                ▷ Executed by the sets  $s \in V_S$ 
  for  $i := \Delta, \Delta/\sqrt[k]{\Delta}, \dots, 0$  do                ▷  $k$  phases.
    Pick  $X_s \sim Geo(1 - \frac{1}{\sqrt[k]{\Delta}})$ . ▷  $Geo(p)$  is the geometric distribution with success prob.  $p$ .
    Round  $X_s$  down to  $\Phi := 4k$  if necessary.
    Wait  $\Phi - X_s$  rounds.
    for  $\ell = 1, \dots, 4k$  do                                    ▷  $4k$  slots.
      Listen for BEEPS
    end for
    if  $s$  receives more than  $3k$  BEEPS then
      Add  $s$  to  $S$ .                                              ▷ Last slot of the round
      Announce to neighbors via BEEP.                            ▷ Each round takes  $4k + 1$  slots.
    end if
    Wait  $X_s$  rounds until end of phase.                       ▷ Note that  $\Phi = (\Phi - X_s) + X_s$ .
  end for                                                       ▷ Each phase takes  $k$  rounds.
end procedure

procedure ELEMENTS( $e, N_e, k, \Delta$ )                            ▷ Executed by the elements  $e \in V_U$ 
  for  $i := \Delta, \Delta/\sqrt[k]{\Delta}, \dots, 0$  do            ▷  $k$  phases.
     $Y_e \sim B(\frac{4k}{\Delta_i})$                                     ▷  $B(p)$  returns 1 with prob.  $p$ 
     $a_e \sim Uni(1, \dots, 4k)$     ▷  $Uni(a, \dots, b)$  picks an integer in  $[a, b]$  uniformly at random
    for  $j := 0, \dots, 4k$  do                                    ▷  $4k$  rounds.
      for  $\ell = 1, \dots, 4k$  do                                    ▷  $4k$  slots.
        if ( $Y_e = 1$ ) and ( $a_e = \ell$ ) and ( $e$  is uncovered) then
          Send BEEP to all neighboring sets.
        end if
      end for
      Listen if a neighbor added itself.                          ▷ Last slot of the round
      Set  $e$  as covered, if it received a BEEP                    ▷ Each round takes  $4k + 1$  slots.
    end for
  end for                                                       ▷ Each phase takes  $4k$  rounds.
end procedure

```

Proof. Before we go into the details of the proof, we define the functions $c_{min}(u)$ and $c_{max}(u)$ for each covered element $u \in V_U$. Suppose u is covered by sets s_1, \dots, s_m , i.e., these sets add themselves simultaneously. Further, let $d(s_1), \dots, d(s_m)$ be *spans* of s_1, \dots, s_m , i.e., the number of uncovered elements neighboring s_1, \dots, s_m . Based on this, we define c_{max} based on the set with *fewest* uncovered elements, i.e., the set that deviates the most from whatever set the greedy solution would have picked. The value c_{min} on the other hand is determined by the set in u 's neighborhood with biggest possible span, i.e., the set that the greedy solution would have picked. Note that this set may not be part of s_1, \dots, s_m . Formally, we have:

$$c_{max}(u) = \max_{v_i \in \{s_1, \dots, s_m\}} \frac{1}{d(s_i)} \quad \text{and} \quad c_{min}(u) = \min_{s \in N_u} \frac{1}{d(s)} \quad (4)$$

Further, by the analysis of that greedy algorithm, it holds that:

$$\sum_{u \in V_U} c_{min}(u) \leq H_\Delta |S_{OPT}| \quad (5)$$

On the other hand, by the definition of $\eta(u)$, it holds:

$$c_{max}(u) = \eta(u) \cdot c_{min}(u) \quad (6)$$

Given these definitions, we now consider a single round i . In the following, let S_i be set of candidates that add themselves to S in round i . Let V_i denote the set of elements that are uncovered at the start of round i . We have:

$$|S_i| \leq \sum_{v \in S_i} \frac{d(v)}{d(v)} \leq \sum_{v \in S} |C(v)| \frac{1}{d(v)} \quad (7)$$

$$\leq \sum_{v \in S} \sum_{u \in C(v)} c_{max}(u) \quad (8)$$

$$= \sum_{u \in V_i} c_{max}(u) \mu(u) \leq \sum_{u \in V_i} c_{min}(u) \cdot \eta(u) \cdot \mu(u) \quad (9)$$

For the expected value of S_i , this implies:

$$E[|S_i|] \leq \sum_{u \in V_i} c_{min}(u) E[\eta(u) \cdot \mu(u)] \quad (10)$$

$$\leq \sum_{u \in V_i} c_{min}(u) \cdot E[\eta(u)] \cdot \Pr[\mu(u) > 0] E[\mu(u) \mid \mu(u) > 0] \quad (11)$$

$$\leq \sum_{u \in V_i} c_{min}(u) \cdot E[\eta(u)] \cdot E[\mu(u) \mid \mu(u) > 0] \quad (12)$$

The second line follows from the fact that the random events that determine $\eta(u)$ and $\mu(u)$ are independent. Finally, we can use the linearity of expectation to sum over all k^2 rounds and get:

$$E[|S|] = \sum_{i=1}^{k^2} E[|S_i|] \leq \sum_{u \in V} E[\eta(u)] \cdot E[\mu(u) \mid t(u) > 0] c_{min}(u) \leq E[\eta] \cdot E[\mu] \cdot H_\Delta |S_{OPT}| \quad (13)$$

This proves the lemma. ◀

Given this lemma, it remains to bound $E[\eta]$ and $E[\mu]$. We begin with the latter. Our goal is to use a result by Miller et al. [9] (which needs to be adapted for geometric values). They showed that the number of candidates that pick the earliest possible wake-up time is $\sqrt[k]{\Delta}$ in expectation. Similarly, we can show the following result⁴.

► **Lemma 5.** *For any value $t > 1$ and node $u \in V$, it holds $\Pr[\mu_u > t] \leq (1 - 1/\sqrt[k]{\Delta})^t$*

Proof. For each node $u \in V$ we define the random variable \mathcal{P} that denotes the *phase* in which u is covered. Recall that every element is *always* covered because in the last round of the last phase, all remaining sets that cover at least one element simply add themselves to the solution. Thus, the variable \mathcal{P} takes values in $0, \dots, 4k$. In the remainder, we will focus only on a single phase. Throughout this proof, we divide the sets into two subsets. First, let N_u be the set of sets in u 's neighborhood, i.e., the sets that can cover u . We call these the *neighboring candidates*. Second, let $NN := V_S \setminus N_u$ be all other sets. We call these the *non-neighboring candidates*. Further, we define the wakeup time Φ_s of each $s \in V_S$ as $\Phi_s := 4k - X_s$. Now we observe the round in which u is covered and see:

⁴ Note that that the analysis in [4] only bounds this term in expectation (and not exact probability) and does not parameterize it on the number of phases as we do. **We need both these aspects for our problem.** Adding these two aspects to their analysis does not seem to be straightforward as they use a rather complex term to bound the expectation, whereas we can just exploit the fundamental properties of the geometric distribution.

► **Lemma 6.** *Suppose u gets covered in the first round of phase j , then*

$$\Pr[\mu_j \geq t \mid u \text{ is covered in round } 0] \leq \left(1 - \frac{1}{\sqrt[k]{\Delta}}\right)^t \quad (14)$$

Proof. In this case, the values X_s of *all* candidates that cover u are bigger than or equal to $4k$. Otherwise, they would have added themselves later. We call these the *early* candidates. For a single candidate $s \in N_u$, the probability to be *early* is at most

$$\Pr[s \text{ is early}] = \sum_{i=0}^{\infty} \frac{1}{\sqrt[k]{\Delta}}^{4k+i} = \frac{1}{\Delta^4} \sum_{i=0}^{\infty} \frac{1}{\sqrt[k]{\Delta}}^i \leq \frac{2}{\Delta^4} \quad (15)$$

This follows directly from the definition of the geometric distribution. As there are *at most* Δ neighboring candidates that cover u , the expected number of early candidates is at most $\frac{2}{\Delta^3}$ as it holds that:

$$E[\text{Early sets that cover } u] = \sum_{s \in N_u} \Pr[s \text{ is early}] = \sum_{s \in N_u} \frac{2}{\Delta^4} \leq \frac{2\Delta}{\Delta^4} = \frac{2}{\Delta^3} \quad (16)$$

As $\Delta \geq 2$ this further simplifies to $\frac{1}{\Delta^2}$. Since all candidates pick their wakeup time independently and μ_u is the exactly the number of early candidates, we can use Chernoff to show that:

$$\Pr[\mu_j \geq t \mid u \text{ is covered in round } 0] = \Pr\left[\mu_u \geq t\Delta^2 \frac{1}{\Delta^2}\right] \leq \Pr[\mu_u \geq t\Delta^2 E[\mu_u]] \quad (17)$$

$$\leq e^{-\frac{t\Delta^2}{3}} \leq e^{-t} \leq \left(1 - \frac{1}{\sqrt[k]{\Delta}}\right)^t \quad (18)$$

This was to be shown. ◀

The interesting stuff happens, if u gets covered in any other round of phase i . This part is more complex than the previous one as it requires a more careful analysis of the starting times. First, we will fix all of the random decisions made the algorithm **except** the decision of the neighboring sets. In particular, we condition on the following three variables:

1. **The wakeup times of the non-neighboring sets** Z_{NN} . It holds $Z_{NN} := (\phi_1, \dots, \phi_m)$. For each $l \in [0, m']$, the non-neighboring set s_l wakes up in round ϕ_l .
2. **The BEEPS by all uncovered elements** Z_U .

This random variable contains all random choices made by the uncovered elements, i.e. $Z_U := \{(Y_e, a_e) \mid e \in V_U\}$. Here, the variable Y_e denotes whether e is active and beeps and a_e denotes the slot in which e beeps. Based on this, we can define the *number* of slots in which a set s receives a BEEP from an element in subset $U \subseteq V_U$ as follows:

$$\text{SLOTS}(s, U) := |\{i \in [0, 4k] \mid \exists e \in U \cap N_s : Y_e = 1 \wedge a_e = i\}| \quad (19)$$

3. **The initial set of uncovered elements** \mathcal{U}_0 .

This is the set of uncovered elements in round 0 of phase j .

If we fix all these choices, the decision if a neighboring candidate of u adds itself to the solution that depends on *solely* its wake-up time. More precisely, the random choices of the other nodes define the last possible round in which a candidate may add itself to the solution. If it wakes up before this round, the concrete round is *only* determined by the geometric distribution. If it wakes up after this round, it will not add itself.

Now we claim the following

▷ Claim 7. Given $(Z_{NN}, Z_U, \mathcal{U}_0)$ for each neighboring set $s \in N_u$ there is a value $\rho_s \in [0, 4k]$ that denotes the last round in which s can cover u , i.e., $\Pr[s \text{ covers } u \text{ first} \mid \Phi_s \geq \rho_s] = 0$

The idea behind the construction is that the *fixed* values \mathcal{U}_0 , Z_{NCC} , and Z_U can clearly define the sets $\mathcal{A}_0 \subset V_S$ that add themselves to the solution in round 0. Given Z_{NN} , we see which sets wake up and count the BEEPS of their uncovered elements. These BEEPS are based on \mathcal{U}_0 and Z_U , so we see if enough elements beep in distinct slots. In particular, the set must receive a BEEP in at least $3k$ slots, so it must hold that

$$\mathcal{A}_0 := \{s \notin N_u \mid \phi_i = 0 \wedge (|\text{SLOTS}(s, \mathcal{U}_0)| \geq 3k)\} \quad (20)$$

Given \mathcal{A}_0 , i.e., the sets that add themselves in round 0, we can then compute $\mathcal{U}_1 \subseteq \mathcal{U}_0$, i.e., all uncovered elements in round 1. Then, by repeating the construction, we can use \mathcal{U}_1 , Z_{NCC} , and Z_U to determine \mathcal{A}_1 and therefore \mathcal{U}_2 and so on. This can be continued until round τ by the following recursive formulas:

$$\mathcal{A}_t = \{s \notin N_u \mid \phi_i = 0 \wedge (|\text{SLOTS}(s, \mathcal{U}_t)| \geq 3k)\} \quad (21)$$

and

$$U_t = \left\{ e \in \mathcal{U} \setminus \{u\} \mid \exists s \in \bigcup_{i=0}^{t-1} \mathcal{A}_i \right\} \quad (22)$$

This yields the uncovered elements $\mathcal{U}_0, \dots, \mathcal{U}_\tau$. For each candidate $s \in N_u$, we can then clearly identify the first round ρ_s where s does not receive enough BEEPS to add itself:

$$\rho_s := \arg \max_{0 \leq \tau \leq 4k} \text{SLOTS}(s, \mathcal{U}_\tau) \geq 3k \quad (23)$$

Given this observation, we can map each outcome of Z_{NCC} , Z_U , and \mathcal{U}_0 to a collection of thresholds $\bar{\rho} := (\rho_1, \dots, \rho_{\Delta_u})$ with the properties above. If we now condition on $\bar{\rho}$ and $\mathcal{P} = j$, we get – similar to Miller et al. in [9] – that:

► **Lemma 8.** *For any possible realization of thresholds $\bar{\rho}$ it holds: $\Pr[\mu_j \leq t \mid \bar{\rho}] \leq \left(1 - 1/\sqrt[k]{\Delta}\right)^t$*

Proof. For each neighboring set $s_\ell \in N_u$ define the adapted wakeup time as:

$$\Phi'_\ell := \begin{cases} (\Phi - X_\ell) & \text{if } (\Phi - X_\ell) \geq \rho_\ell \\ \infty & \text{else} \end{cases} \quad (24)$$

Here X_ℓ is the geometric random variable drawn to determine the wake-up time.

We can now order these adapted wakeup times $\Phi'_{(1)}, \dots, \Phi'_{(m)}$ such that $\Phi'_{(1)}$ is the earliest wakeup and $\Phi'_{(\Delta_u)}$ is the last. For each of these ordered wakeup times $\Phi_{(i)}$ we define $c_{(1)}$ to be the candidate that achieves this time, $X_{(i)}$ is the variables drawn by this set, and $\rho_{(i)}$ is the threshold. Given that $\Phi_{(i)} \leq \infty$, it holds:

$$\Phi'_{(i)} := \Phi - X_{(i)} \quad (25)$$

So the variable $X_{(1)}$ is *not* be the smallest variable that any neighboring candidate drew, but instead the smallest variable by any neighboring candidate *with finite wake-up time*.

Given this definition, we claim the following:

▷ Claim 9. The candidate $c_{(1)}$ archiving $\Phi'_{(1)}$ covers u if and only if $\Phi'_{(1)} \leq \infty$.

Proof. This claim can easily be verified, by observing the two possibilities. If $\Phi'_{(1)} \leq \infty$, then $\Phi'_{(1)}$ is earliest round where some neighborhood candidate tries to itself and is still a candidate. This follows directly from the definition of $\Phi'_{(1)}$. Thus, $c_{(1)}$ must cover u as no candidate could have done it before. Otherwise, if $\Phi'_{(1)} = \infty$, then it must hold that $\Phi - X_{(1)} \geq \rho_{(1)}$. Moreover, this implies that $\Phi - X_{(i)} \geq \rho_{(i)}$ for all other candidates as well. This follows from the fact that $\Phi'_{(1)}$ is the smallest wake-up time by definition, so all others must be infinity, too. In this case, *all* potential candidates do not cover enough elements when they wake up. This follows directly from the definition of each $\rho_{(i)}$. Thus, in this case, u is not covered by any candidate in this phase. \blacktriangleleft

Thus, for $\mu_u = t$ the t smallest values must all be equal and non-infinity. Formally:

$$\Pr[\mu_j = t] = \Pr[\Phi'_{(1)} = \dots = \Phi'_{(t)} \mid \Phi_{(1)} \leq \infty] \quad (26)$$

What follows are some fundamental calculations based on Miller et al.'s proof for exponential random variables. However, we need to adapt them to geometric variables. First, we observe that it holds:

$$\Pr[(\Phi - X_\ell) \leq \rho_\ell - x] = \Pr[-X_\ell \leq \rho_\ell - \Phi - x] = \Pr[X_\ell \geq (\Phi - \rho_\ell) + x] \quad (27)$$

And therefore, for any $x \leq \rho_\ell$ we see that the wakeup time is only determined by the parameter of the geometric distribution, namely

$$\Pr[\Phi'_\ell < x \mid \Phi'_\ell \leq x] = \Pr[(\Phi - X_\ell) \leq x] \mid \Phi'_\ell \leq x] = \Pr[X_\ell > \Phi - x \mid \Phi'_\ell \leq x] \quad (28)$$

$$= \Pr[X_\ell > \Phi - x \mid X_k \geq \Phi - x] \quad (29)$$

Now we can use the fact of the geometric distribution is memoryless and see:

$$\Pr[\Phi'_\ell < x \mid \Phi'_\ell \leq x] = \Pr[X_\ell > y \mid X_\ell \geq y] \quad \triangleright \text{Substituting } y := \Phi - x \quad (30)$$

$$= \Pr[X_\ell > 0] = \left(\frac{1}{\frac{k}{\sqrt{\Delta}}} \right) \quad (31)$$

Therefore, we can conclude the following for two consecutive $\Phi_{(i)}$ and $\Phi_{(i+1)}$ that it holds:

$$\Pr[\Phi'_{(i)} < x \mid \Phi'_{(i+1)} = x] = \Pr[\Phi_{(i)} \leq x + 1 \mid \Phi_{(i+1)} = x] \quad (32)$$

$$= \Pr[X_{(i)} \geq \Phi - x + 1 \mid X_{(i)} \geq \Phi - x] \quad (33)$$

$$= \Pr[X_{(i)} \geq 1] = \left(\frac{1}{\frac{k}{\sqrt{\Delta}}} \right) \quad (34)$$

And thus, for the opposite event it holds:

$$\Pr[\Phi'_{(i)} = x \mid \Phi'_{(i+1)} = x] = \left(1 - \frac{1}{\frac{k}{\sqrt{\Delta}}} \right) \quad (35)$$

Finally, we condition on $\Phi'_{(t)} = \tau$ for a round $0 \leq \tau \leq 4 \log(\Delta)$. Using the chain rule of conditional probability, we see that:

$$\Pr[\Phi'_{(1)}, \dots, \Phi'_{(t)} = \tau] = \prod_{i=1}^t \Pr[X_{(i)} = \tau \mid X_{(i+1)} = \tau] \leq \prod_{i=1}^t \left(1 - \frac{1}{\frac{k}{\sqrt{\Delta}}} \right) = \left(1 - \frac{1}{\frac{k}{\sqrt{\Delta}}} \right)^t \quad (36)$$

Note that this is independent of the actual round, the law of total probability yields the the result. It holds:

$$\Pr[\mu_u \geq t] := \sum_{\tau=1}^{\Delta} \Pr[\Phi'_{(t)} = \tau] \Pr[\Phi'_{(1)}, \dots, \Phi'_{(t)} = \tau] \quad (37)$$

$$= \sum_{\tau=1}^{\Delta} \Pr[\Phi'_{(t)} = \tau] \left(1 - \frac{1}{\sqrt[k]{\Delta}}\right)^t = \left(1 - \frac{1}{\sqrt[k]{\Delta}}\right)^t \quad (38)$$

◀

As the concrete value of the ρ_i 's and the phase is immaterial, by the law of total probability and the lemma follows for a single phase. ▶

Thus, the geometric series implies $E[\mu] \leq \sum_{t=1}^{\infty} (1 - 1/2)^t = \frac{1}{1 - 1/2} = 2$. Further, we show that:

► **Lemma 10.** For $k \geq 3$ it holds $E[\eta] \leq 25 \cdot \log(\Delta) \cdot \left(\sqrt[k]{\Delta}\right)^2$

Proof. In the following, we fix an element $u \in \mathcal{U}$. To proof the lemma, we observe the event that there is any set with span bigger than $6 \cdot \log(\Delta) \cdot \sqrt[k]{\Delta} \cdot \Delta_i$ and any set smaller than $\frac{\Delta_i}{4 \cdot \sqrt[k]{\Delta}}$ joins the solution. Denote this event as \mathcal{B} . Given that $\Pr[\mathcal{B}] \leq 1/\Delta$, the law of total expectation implies:

$$E[\eta(u)] = \Pr[\mathcal{B}] \cdot E[\eta(u) \mid \mathcal{B}] + \Pr[\neg \mathcal{B}] \cdot E[\eta(u) \mid \neg \mathcal{B}] \quad (39)$$

$$\leq \left(1 - \frac{1}{\Delta}\right) 24 \cdot \log(\Delta) \cdot \left(\sqrt[k]{\Delta}\right)^2 + \frac{2}{\Delta} \cdot \Delta \quad (40)$$

$$\leq 25 \cdot \log(\Delta) \cdot \left(\sqrt[k]{\Delta}\right)^2 \quad (41)$$

Thus, in the following, we show that $\Pr[\mathcal{B}]$ is at most $\frac{1}{\Delta}$. Let $H_u^i \subseteq N_u$ denote all neighbors that span $20 \cdot \log(\Delta) \cdot \sqrt[k]{\Delta} \cdot \Delta_i$ uncovered elements at any point in phase i and likewise let $L_u^i \subseteq N_u$ denote all neighbors that span less than $\frac{\Delta_i}{4 \cdot \sqrt[k]{\Delta}}$ at any point in the phase. Then, it holds via union bound that:

$$\Pr[\mathcal{B}] \leq \Pr[\exists s \in H_u] + \Pr[\exists s \in L_u : s \text{ adds itself to the solution}] \quad (42)$$

First, we observe that H_u^i is empty with prob. $1/\Delta$. The proof is straightforward: Suppose there is a set with $c \cdot \log(\Delta) \cdot \sqrt[k]{\Delta} \cdot \Delta_i$ uncovered neighbors in phase i . Then it had at least $c \cdot \log(\Delta) \cdot \Delta_{i-1}$ uncovered neighbors when it woke up in phase $i - 1$. The probability that in a fixed slot (of the $4k$ slots of that round), no elements beeps is:

$$\Pr[\text{No BEEP in slot } j] \leq \left(1 - \frac{1}{4k} \cdot \frac{4k}{\Delta_{i-1}}\right)^{c \cdot \log(\Delta) \cdot \Delta_{i-1}} \leq e^{-c \log(\Delta)} = \frac{1}{\Delta^c} \quad (43)$$

If $c \geq 6$, a union bound over all $4k$ slots then yields that all that with probability higher than $1 - \frac{1}{\Delta^2}$, there was at least one beeping element in every slot when s woke up. In particular, there was beeping element in more than $3k$ slots. Thus, the set in question must have added itself to the solution in phase $i - 1$ and it holds:

$$\Pr[s \in H_u^i : s \text{ was not added in phase } i - 1] \leq \sum_{j=1}^{4k} \Pr[\text{No BEEP in slot } j] \leq \frac{1}{\Delta^{c-4}} \leq \frac{1}{\Delta^2} \quad (44)$$

Now consider a set $s \in L_u^i$. The probability that s has $3k$ beeping neighbors on wake-up is bounded as follows:

$$\Pr[s \text{ has } \geq 3k \text{ beeping neighbors}] \leq \sum_{i=3k}^{\Delta} \binom{\frac{\Delta_i}{4^{\frac{k}{\sqrt{\Delta}}}}}{i} \left(\frac{4k}{\Delta_i}\right)^{3k} \leq \Delta \binom{\frac{\Delta_i}{4^{\frac{k}{\sqrt{\Delta}}}}}{3k} \left(\frac{4k}{\Delta_i}\right)^{3k} \quad (45)$$

$$\leq \Delta \left(\frac{e \cdot \Delta_i}{4^{\frac{k}{\sqrt{\Delta}}} \cdot 3k}\right)^{3k} \frac{4k^{3k}}{\Delta_i^{3k}} \leq \frac{1}{\Delta^2} \quad (46)$$

Thus, the probability that s has beeping neighbors in $3k$ *distinct* slots (and therefore adds itself to the solution) can only be smaller. Therefore, it holds

$$\Pr[\mathcal{B}] \leq \Pr[\exists s \in H_u] + \Pr[\exists s \in L_u : s \text{ adds itself to the solution}] \quad (47)$$

$$\leq \Pr[\exists s \in H_u] + \Pr[\exists s \in L_u : s \text{ has } 3k \text{ beeping neighbors}] \quad (48)$$

$$\leq \sum_{s \in H_s} \Pr[s \text{ was not added in phase } i-1] + \sum_{s \in L_s} \Pr[s \text{ has } 3k \text{ beeping neighbors}] \quad (49)$$

$$\leq 2\Delta \frac{1}{\Delta^2} = \frac{2}{\Delta} \quad (50)$$

This is what we wanted to show. ◀

This proves the theorem as $E[\eta \cdot \mu] \in O(\log(\Delta)^2 \sqrt[k]{\Delta^3})$.

3.2 Extension to DOMINATINGSET

Note that the algorithm can also be extended to the DOMINATINGSET problem, where each set is also an element. At first glance, all elements could simply also execute the code for the sets. Note, however, that a node cannot simultaneously beep and listen, which causes some problems with a direct simulation. In particular, an *active* uncovered elements cannot also listen for BEEPS in all slots. To solve this, we simply add all active elements to the solution. In phase i , each set that is added to the solution covers $\Omega(\log(n)\Delta_{i-1})$ elements, w.h.p.⁵ Suppose there are U_i uncovered elements, then optimal solution must at least be of size $\Theta(\log(n) \frac{U_i}{\Delta_{i-1}})$. As in phase i there are – by Chernoff – only $O(\log(n) \frac{U_i}{\Delta_i})$ active elements, w.h.p. we can just add to the solution without (asymptotically) affecting the approximation ratio. Note that this also implies that for $k = \log(\Delta)$ the total number of BEEPS is $\tilde{O}(|S_{OPT}|)$.

4 A Low-Message KT_0 Algorithm

We now move away from the BEEPING model and present our low-message KT_0 -CONGEST algorithm. In this section, we make the following less restrictive assumptions about the model:

1. Again, we observe a fixed communication graph $G := (V_S \cup V_U, E)$ with $V_S = n$ and $V_U = m$ with bidirected communication edges $\{s, e\} \in E$ between a set and its elements. Sets and elements can locally distinguish between their edges through port numbers, but no global identifiers uniquely identify a node. To simplify the presentation, all nodes know Δ and $\log(n+m)$.
2. Time proceeds in synchronous *rounds*. Each round, an element or set can send a distinct message of size $O(\log(n))$ to any subset of its neighbors. Messages are received in the next round.
3. All nodes wake up in the same round.

⁵ If we simply choose c big enough in Equation 43, we see that all sets with $\log(n)\Delta_i$ elements are gone, w.h.p)

In other literature, this model is sometimes referred to as the *clean network model*[11]. Note that this model does not intend to represent ad-hoc networks faithfully but instead is used to analyze the message efficiency of distributed algorithms (see .e.g., [3] for an overview). Similar to the BEEPING model, it starts with limited knowledge of its neighborhood and must *learn* everything it needs to solve the given problem. Given this model, we will show the following theorem:

► **Theorem 11.** *There is an algorithm in the KT_0 -CONGEST model that solves SETCOVER in time $O(\log^2(\Delta))$, expected approximation ratio of $O(\log(\Delta))$, and sends only $\tilde{O}(\sqrt{\Delta}(n+m))$ messages, w.h.p., given that all nodes know Δ and an approximation of $\log(n)$.*

As it turns out, we will only need a few changes to our already established algorithm to prove this theorem. Every BEEPING-Algorithm also works in the KT_0 -CONGEST-Model mentioned above, as the model is less restricted. To obtain a message efficient algorithm, we only need to make a few minor changes to our algorithm: First, to simplify the presentation, we do not parameterize the algorithm with k . Instead, we fix $k = \log(\Delta)$ and only consider this case. Second, we do not require the notion of slots anymore as a node can simply count how many of its neighbors beeped in a single round as the messages arrive via distinct channels. Each *round* now only consists of precisely one slot. In particular, an active element does *not* pick a slot number anymore, but directly sends its BEEP. Further, instead of beeping *all* neighbors with probability $\frac{2^i}{\Delta}$, each element picks $\frac{\log(n)2^i}{\Delta}$ neighbors uniformly at random and sends a BEEP. Finally, the most significant change to the algorithm is the following: Instead of executing all phases of the algorithm, we only execute it until phase $\frac{\log(\Delta)}{2}$, i.e., until the active degree of all sets is around $O(\sqrt{\Delta})$. We call this the first *stage* of the algorithm. Then, all uncovered elements notify their respective set that they are uncovered. Denote these elements as \mathcal{U}' . Finally, the algorithm continues (almost) as usual for the remaining $\frac{\log(\Delta)}{2}$ phases, but each set that joins the solution notifies *only* the elements in \mathcal{U}' that they are covered. We call this the second *stage* of the algorithm.

We will now prove Theorem 2. One can easily verify that most lemmas from our previous analysis are still correct. The first stage does not differ from our previous algorithm at all. Further, all nodes that do not receive any message in the second stage of the algorithm (but would have in the original algorithm) are already covered. Thus, they would be idle regardless. Thus, we only need to show that $E[\eta]$ is still small to prove the approximation ratio.

Before we start, we need the following auxiliary lemma that tightly bound the sets that add themselves to the solution in a phase i .

► **Lemma 12.** *Suppose each elements picks $c \cdot 8 \cdot \log(n) \frac{2^i}{\Delta}$ active edges uniformly and independently at random. Further, each set with at least $c \cdot 4 \log(n)$ active edges adds itself on wake-up. Then the following two statements hold w.h.p:*

1. *At the end of phase i there is no set with $\frac{\Delta}{2^i}$ uncovered elements.*
2. *Any set that adds itself in round i covers at least $\frac{\Delta}{8 \cdot 2^i}$ elements.*

Both statements follow through an elementary application of the Chernoff Bound. Note that this lemma directly implies that $E[\eta] \in O(1)$ as it bounds the *worst* and *best* set in each rounds, w.h.p. Therefore expected approximation ratio is $O(\log(\Delta))$, which is as good as the sequential greedy solution.

Proof. Both statement follow from Chernoff, we prove them both separately.

1. Let s be a set with $\frac{\Delta}{2^i}$ uncovered element on wake-up, then the expected number on active edges is

$$E[A_s^i] = \frac{\Delta}{2^i} \cdot c8 \log(n) \cdot \frac{2^i}{\Delta} = c8 \log(n) \quad (51)$$

Thus, the probability that only $c4 \log(n)$ edges are active is

$$\Pr[A_s^i \leq c4 \log(n)] \leq \Pr[A_s^i \leq (1 - \frac{1}{2})E[A_s^i]] \leq e^{-\frac{c8 \log(n)}{2 \cdot 2^i}} = \frac{1}{n^c} \quad (52)$$

Since with more uncovered elements, the probability can only be smaller, the statement follows.

2. Let s be a set with $\frac{\Delta}{8 \cdot 2^i}$ uncovered element on wake-up, then the expected number on active edges is

$$E[A_s^i] = \frac{\Delta}{8 \cdot 2^i} \cdot c8 \log(n) \cdot \frac{2^i}{\Delta} = c \log(n) \quad (53)$$

Thus, the probability that at least $c4 \log(n)$ edges are active is

$$\Pr[A_s^i \leq c4 \log(n)] \leq \Pr[A_s^i \leq (1 + 3)E[A_s^i]] \leq e^{-\frac{3^2 c \log(n)}{3+2}} \leq \frac{1}{n^c} \quad (54)$$

Since with fewer uncovered elements, the probability can only be smaller, the statement follows. \blacktriangleleft

Thus, it only remains to analyze the message complexity. We prove the message bound for each stage of the algorithm. We begin with the first stage and show that for it holds:

► **Lemma 13.** *Until phase $\frac{\log(\Delta)}{2}$, the nodes send at most $\tilde{O}(n\sqrt{\Delta})$ messages.*

Proof. As each element picks (up to) $c8 \log(n)2^i/\Delta$ edges for some constant $c > 0$ in phase i , the lemma follows immediately. The corresponding bound for the sets is less trivial as it does not directly follow from the algorithm. Here, we need to consider that only sets that add themselves to the solution send messages. In particular, each set that adds itself sends at most Δ messages and remains silent otherwise. Thus, we show that at most $\tilde{O}(\frac{n}{\sqrt{\Delta}})$ sets add themselves w.h.p. Note that every set that adds itself covers (at least) $\sqrt{\Delta}/8$ uncovered elements, w.h.p, otherwise it would *not* have added itself. This follows from the second statement in Lemma 12. On the other hand, each uncovered element is covered by *at most* $O(\log(n))$ sets, w.h.p. This follows from choosing $t \geq c \log(n)$ for some $c > 0$ in Lemma 5. Let $S_i \subset V_S$ be the solution in phase i and C_i be the covered elements, then it must hold:

$$|S_i| \cdot \sqrt{\Delta}/8 \leq c \log(n) |C_i| \Leftrightarrow |S_i| \leq \frac{8c \log(n) |C_i|}{\sqrt{\Delta}} \Leftrightarrow |S_i| \leq \frac{8c \log(n) n}{\sqrt{\Delta}} \in \tilde{O}(\frac{n}{\sqrt{\Delta}}) \quad (55)$$

In other words, if there are more $\tilde{O}(\frac{n}{\sqrt{\Delta}})$ sets that added themselves, then there must exist an element covered by more than $c \log(n)$ sets. This is a contradiction, which implies the lemma \blacktriangleleft

This lemma concludes the analysis of the algorithm's first stage. Finally, we need to observe the second stage. In this stage, the algorithm only uses communication edges adjacent to the set of uncovered elements in phase $\log(\Delta)/2$. Thus, to determine the message complexity, we only need to count these edges. Formally, we show:

► **Lemma 14.** *In phase $\frac{\log(\Delta)}{2}$, each set has at most $O(\sqrt{\Delta})$ uncovered elements.*

Proof. The lemma follows directly from the first statement of Lemma 12 as all sets that have more than $O(\sqrt{\Delta})$ uncovered elements must have added themselves in an earlier round w.h.p. \blacktriangleleft

Thus, in the remaining $O(\log^2(\Delta))$ rounds of the algorithm, all communication will only take place via these $O(m\sqrt{\Delta})$ edges. Since at most one message passes each edge in every round, this implies that at most $O(m\sqrt{\Delta} \log^2(\Delta))$ messages are sent, which proves Theorem 2.

Lower Bound

In this section, we prove a lower bound on the number of messages needed to approximate a solution. The proof works via a reduction to the sequential case. Here, it is well known that a large portion of the input, i.e., the connections between the nodes, must be revealed to the algorithm. In particular, it is known that the following holds:

► **Lemma 15** (Lower bound from [5]). *Consider a sequential computation model that allows the following two queries*

- **EltOf(i,j)** - Returns the j^{th} element of Set S_i or \perp if there is no such element.
- **SetOf(i,j)** - Returns the j^{th} set which constrains e_i or \perp if there is no such set.

Then, every algorithm that yields $O(1)$ -approximation for SETCOVER needs at least $\tilde{\Omega}(m\sqrt{n})$ queries on certain graphs.

In particular, the lower bound graph in [5] has maximal degree n and it holds $m = n$. Therefore, the bound can be rewritten as $\tilde{\Omega}((m+n)\sqrt{\Delta})$, which is exactly the message complexity of our algorithm. Now we show that if there is a distributed algorithm with less than $\tilde{\Omega}(m\sqrt{n})$ messages, it can be turned into a sequential algorithm with less than $\tilde{\Omega}(m\sqrt{n})$ queries. This is, of course, a contradiction to the lemma above. The proof's main ingredient is the observation that every message that is sent from v along its j^{th} channel can be emulated as looking up $\text{SetOf}(v, j)$. Other than that, the proof is quite technical. The main result is as follows:

► **Lemma 16.** *Any algorithm that yields an $O(1)$ -approximation for SETCOVER needs at least $\tilde{\Omega}(m\sqrt{n})$ messages on certain graphs in KT_0 model.*

Proof. The lemma follows from the fact that any KT_0 algorithm that sends $O(x)$ messages can be simulated with $O(x)$ queries in the sequential model. First, we create $(m+n)$ objects that store the internal variables of each set and element. These objects are stored in two arrays A_S and A_E , s.t., the object for e_i can be accessed through $A_E[i]$. Analogously, each set s_j can be accessed through $A_S[j]$. Consider a single round of a CONGEST algorithm: First, we iterate over all objects and perform the local computations that the set or element would execute in the CONGEST algorithm. Now for each message m_{ij} that e_i sends to its j^{th} set, we use the query $\text{SetOf}(i, j)$ to obtain its index j' and then add the message to $A_S[j']$. The same is done vice versa for messages from sets to elements. After all messages have been handled, A_E and A_S contain the nodes' states and all received messages. With this information, each node's action in the distributed algorithm can flawlessly be simulated. Thus, the sequential algorithm again iterates over A_E and A_S to compute the next round's messages according to the algorithm. Therefore, any distributed algorithm sending $O(x)$ messages can be transformed into a sequential algorithm with $O(x)$ queries. Therefore, any lower bound on the queries is also a lower bound on the messages. ◀

5 Conclusion and Future Work

In this work, we presented two different message- and energy-efficient distributed algorithms for SETCOVER for the BEEPING model and the KT_0 model. In future work, it would be interesting to see whether the existence of unique identifiers known to all nodes can improve the message complexity or if similar bounds hold.

References

- 1 Alejandro Cornejo and Fabian Kuhn. Deploying wireless networks with beeps. In Nancy A. Lynch and Alexander A. Shvartsman, editors, *Distributed Computing, 24th International Symposium, DISC 2010*,

- Cambridge, MA, USA, September 13-15, 2010. *Proceedings*, volume 6343 of *Lecture Notes in Computer Science*, pages 148–162. Springer, 2010. doi : 10 . 1007 / 978 - 3 - 642 - 15763 - 9 _ 15.
- 2 Fabien Dufoulon, Janna Burman, and Joffroy Beauquier. Beeping a Deterministic Time-Optimal Leader Election. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing (DISC 2018)*, volume 121 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:17, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: <http://drops.dagstuhl.de/opus/volltexte/2018/9809>, doi : 10 . 4230 / LIPIcs . DISC . 2018 . 20.
 - 3 Robert Gmyr and Gopal Pandurangan. Time-message trade-offs in distributed algorithms. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, volume 121 of *LIPIcs*, pages 32:1–32:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi : 10 . 4230 / LIPIcs . DISC . 2018 . 32.
 - 4 Christoph Grunau, Slobodan Mitrovic, Ronitt Rubinfeld, and Ali Vakilian. Improved local computation algorithm for set cover via sparsification. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2993–3011. SIAM, 2020. doi : 10 . 1137 / 1 . 9781611975994 . 181.
 - 5 Piotr Indyk, Sepideh Mahabadi, Ronitt Rubinfeld, Ali Vakilian, and Anak Yodpinyanee. Set cover in sub-linear time. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2467–2486. SIAM, 2018.
 - 6 Lujun Jia, Rajmohan Rajaraman, and Torsten Suel. An efficient distributed algorithm for constructing small dominating sets. *Distributed Comput.*, 15(4):193–205, 2002.
 - 7 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. The price of being near-sighted. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 980–989. ACM Press, 2006. URL: <http://dl.acm.org/citation.cfm?id=1109557.1109666>.
 - 8 Fabian Kuhn and Roger Wattenhofer. Constant-time distributed dominating set approximation. In Elizabeth Borowsky and Sergio Rajsbaum, editors, *Proceedings of the Twenty-Second ACM Symposium on Principles of Distributed Computing, PODC 2003, Boston, Massachusetts, USA, July 13-16, 2003*, pages 25–32. ACM, 2003. doi : 10 . 1145 / 872035 . 872040.
 - 9 Gary L. Miller, Richard Peng, Adrian Vladu, and Shen Chen Xu. Improved parallel algorithms for spanners and hopsets. In Guy E. Blelloch and Kunal Agrawal, editors, *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, pages 192–201. ACM, 2015.
 - 10 Damon Mosk-Aoyama, Tim Roughgarden, and Devavrat Shah. Fully distributed algorithms for convex optimization problems. *SIAM J. Optim.*, 20(6):3260–3279, 2010.
 - 11 David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, USA, 2000.
 - 12 Christian Scheideler, Andréa W. Richa, and Paolo Santi. An $o(\log n)$ dominating set protocol for wireless ad-hoc networks under the physical interference model. In Xiaohua Jia, Ness B. Shroff, and Peng-Jun Wan, editors, *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2008, Hong Kong, China, May 26-30, 2008*, pages 91–100. ACM, 2008. doi : 10 . 1145 / 1374618 . 1374632.
 - 13 Jiguo Yu, Lili Jia, Dongxiao Yu, Guangshun Li, and Xiuzhen Cheng. Minimum connected dominating set construction in wireless networks under the beeping model. In *2015 IEEE Conference on Computer Communications, INFOCOM 2015, Kowloon, Hong Kong, April 26 - May 1, 2015*, pages 972–980. IEEE, 2015. doi : 10 . 1109 / INFOCOM . 2015 . 7218469.