Approximating the Size of a Radio Network in Beeping Model

Philipp Brandes[1], Marcin Kardas[2], Marek Klonowski[2], Dominik Pająk[2], and Roger Wattenhofer[1] ⋆

[1] ETH Zürich, Switzerland
[2] Department of Computer Science at the Faculty of Fundamental Problems of Technology
Wrocław University of Technology, Poland

Abstract. In a single-hop radio network, nodes can communicate with each other by broadcasting to a shared wireless channel. In each time slot, all nodes receive feedback from the channel depending on the number of transmitters. In the Beeping Model, each node learns whether zero or at least one node have transmitted. In such a model, a procedure estimating the size of the network can be used for efficiently solving the problems of leader election or conflict resolution. We introduce a time-efficient uniform algorithm for size estimation of single-hop networks. With probability at least $1-1/f$ our solution returns $(1+\varepsilon)$-approximation of the network size n within $\mathcal{O}\left(\log\log n+\log f/\varepsilon^2\right)$ time slots. We prove that the algorithm is asymptotically time-optimal for any constant $\varepsilon>0$.

1 Introduction

The number of nodes in the network is a parameter that is necessary to effectively perform many fundamental protocols and is useful for network analysis, gathering statistics etc. However, in modern applications of communication networks we often cannot assume that the size of the network or even its constant-factor approximation in known. Hence, the problem of designing an algorithm to precisely and efficiently estimate the number of nodes in radio networks is an important challenge. This is particularly clear in the context of networks with strictly limited communication channel, wherein one needs a precise estimation of the number of nodes in order to avoid collisions of transmissions caused by several nodes broadcasting at the same time. As a consequence, the most efficient algorithms for classic problems in radio networks, like leader election, use the size approximation as a subroutine.

In our paper we consider the problem of size estimation in a communication model that is weaker than the classic Multiple Access Channel, namely in the Beeping Model.

We consider a wireless network of n devices (nodes). The size n of the network is unknown to the nodes. The nodes have no identifiers or serial numbers that could be used to distinguish them. The aim is to estimate the network's size n by performing random transmissions and using the feedback of the communication channel. The main result of this paper is an asymptotically optimal (with respect to the time of execution) algorithm that returns a $(1+\varepsilon)$-approximation of the number of nodes in the network with controllable error probability. As the second result we show the matching lower bound.

1.1 Model

We study a single-hop radio network of n nodes with the Beeping Model as a communication model ([1,9]). The transmission of each node reaches all other nodes. That is, the network can be represented as a complete graph. We assume that the nodes are identical and indistinguishable and perform the same protocol. However, each node can independently sample any number of random bits. Randomization can be used freely, but the final result of the protocol needs to be deterministically computed based on the knowledge available to all the nodes. We ensure in this way that all the nodes upon completing the procedure obtain the same result, which could also be determined by a passive observer listening to the communication channel.

⋆ This paper is supported by Polish National Science Center – decision number 2013/09/N/ST6/03440 (the second author)

We assume that the time is discrete, i.e., it is divided into slots. We also assume that the nodes are synchronized as if they had access to a global clock. In every slot, each node independently decides whether to transmit to the channel or not. The nodes share a common communication channel and in every slot the channel can be in one of the two following states: NULL, when no node is transmitting and BEEP, if at least one node is transmitting (i.e., the channel is busy). All nodes receive the state of the channel immediately after each communication round.

The Beeping Model can be contrasted with the classical model of Radio Networks with Collision Detection where the channel can be in three states depending on whether zero, one, or more than one, node is transmitting. The third state is called "collision".

The result of any size estimation protocol is a random variable, an estimator $\hat{n}$ of true number of nodes n. We are interested in the probability of getting an approximation that differs from the true value by at most a constant multiplicative factor.

Definition 1. *For any $\varepsilon > 0$, we say that protocol $\mathcal{P}$ $(1+\varepsilon)$-approximates the number of nodes with probability at least $1 - 1/f$, if for any n it returns $\hat{n}$ such that*

$$\mathbb{P}\left(\frac{\hat{n}}{1+\varepsilon} \leq n < (1+\varepsilon)\hat{n}\right) \geq 1 - \frac{1}{f}.$$

The time complexity of protocol is expressed as a function of three variables n, f and ε.

1.2 Related Work

There are many papers devoted to size approximation in radio networks. Most of them work in the model of Radio Networks with Collision Detection. In [2] Bordim et al. presented a size approximation protocol for the network of the (unknown) size n with execution time $\mathcal{O}\left((\log n)^2\right)$ that finds an approximation $\hat{n}$ of the real number of nodes such that

$$\frac{n}{16 \log n} < \hat{n} < \frac{2n}{\log n}$$

with probability at least $1 - \mathcal{O}\left(n^{-1.83}\right)$. The authors assume communication model with collision detection and aim at saving energy of the network. Greenberg et al. [13] proposed a size approximation algorithm working in time $\log n + \mathcal{O}(1)$ producing an estimate of n with mean approximately $0.914n$ and standard deviation of $0.630n$. Greenberg et al. [13] also showed that a size approximation algorithm can be used to efficiently schedule transmissions such that each node succeeds to transmit.

Some papers presented other, more complex protocols that use elaborated size-approximation algorithms as a sub-procedure (e.g. [20]). In paper [19] Nakano and Olariu presented an energy-efficient initialization algorithm which needs to know the number of nodes n or its fair approximation to work properly. In [24] Willard showed an algorithm for a selection problem that needs $O(\log \log n)$ steps on average with a respective lower bound. This result has been used extensively for many other papers about fast leader election and size approximation in the context of radio networks.

An energy-efficient size estimation algorithm is proposed in Jurdziński et al. [15] for a model without collision detection. The algorithm requires $\mathcal{O}\left(\log^{2+\alpha} n\right)$ time slots with nodes being awake for at most $\mathcal{O}\left((\log \log n)^{\alpha}\right)$ slots for any $\alpha > 0$. The algorithm is a c-approximation for some constant c (with respect to n). In [3] authors present approximation of the size of the network in a similar model. Their protocol designed for collision detection model works in $O(\log n \log \log n)$ steps and returns a 2-approximation. The second protocol for no-collision detection settings needs $O(\log^2 n)$ steps for a 3-approximation. Moreover, the authors of [3] take into account energy of nodes necessary for completing the protocol. All the results aforementioned in this paragraph hold with high probability.

The problem of size estimation has been extensively studied in the context of computer databases ([10,11,23,12,5]). In that case, one is interested in estimating the cardinality (the number of distinct elements) of some multiset. Many protocols for size estimation have been proposed for radio networks ([16,7,8]). In many cases (including [13]) the proposed solutions provide asymptotically unbiased estimator $\mathbb{E}(\hat{n}) = n(1 + o(1))$ that is **not** well concentrated, i.e., $(\mathbb{V}\mathrm{ar}(\hat{n}) = \Omega(n^2))$. In such case one can have $\mathbb{P}(|\hat{n} - n| \geq c \cdot n) = \Theta(1)$. Thus one cannot expect obtaining c-approximation with high probability. Moreover, in contrast to most of the previous work, we use a controllable parameter of algorithm's success f. This can be particularly important for small n.

Independently, the problem of estimation of cardinality of a set emerged in the research devoted to RFID (Radio Frequency IDentification) technologies. There are many significant papers including [14,17,18,21,22,25] presenting different methods for various settings offering also some extra features. The result closest to our contribution is included in [6] where authors present a protocol for the model wherein both RFID and a single distinguished device called *the reader* in each round can transmit $O(1)$ bits. Using recent communication complexity result ([4]) they prove that every Monte Carlo counting protocol with relative error $\epsilon \in [1/\sqrt{n}, 0.5]$ and probability of failure smaller than 0.2 needs $\Omega(\frac{1}{\epsilon^2 \log 1/\epsilon} + \log\log n)$ execution time. For the same range of ϵ they demonstrated how to construct a protocol with $O(\frac{1}{\epsilon^2} + \log\log n)$ running-time. The model of a single-hop radio network considered in our paper and models of RFID systems are seemingly completely different. It turns, however, that the results from [6] can be almost instantly applied to the settings investigated in our paper at least for some ranges of parameters. On the other hand their results holds with constant probability while we demand probability of failure limited by $1/f$. As authors of [6] suggested repeating the basic algorithm and choosing the median to obtain arbitrary small probability of failure. Nevertheless, such approach leads to $\Theta(\log f)$ multiplicative factor overhead.

1.3 Our Results

In Section 1.1 we recall our model and introduce some new definitions. In Section 2 we present a time-efficient uniform algorithm for computing a $(1+\varepsilon)$-approximation of the size of the network with probability $1 - 1/f$ (where f is a parameter of the protocol) and provide its analysis. Our protocol requires $\mathcal{O}(\log\log n + \log f/\varepsilon^2)$ time slots.

In Section 3 we give a lower bound for the number of slots that are necessary to get a linear size estimation. For n nodes and any $f \geq 2$ we show that $\Omega(\log\log n + \log f/\varepsilon)$ slots are required to get a $(1+\varepsilon)$-approximation with probability greater than $1 - 1/f$ in the beeping model.

2 Size Estimation Algorithm

In this section we present an algorithm for $(1+\varepsilon)$-approximation of network size working in time $\mathcal{O}(\log\log n + \log f/\varepsilon^2)$ with probability at least $1 - 1/f$. With probability at most $1/f$ the algorithm may return a wrong estimate or work for a larger number of steps (or both). First in Subsection 2.1 we present a procedure for 64-approximation and later in Subsection 2.2 we show how to improve it to $(1+\varepsilon)$-approximation, for any $\varepsilon > 0$. An important feature of our algorithm is its uniformity:

Definition 2. *A randomized distributed algorithm $\mathcal{A}$ is called **uniform** if, and only if, in round i every node that has not yet transmitted successfully, transmits independently with probability p_i (the same for all nodes).*

For k active nodes the probability that exactly j nodes transmit in the i-th round is $\binom{k}{j}(p_i)^j(1-p_i)^{k-j}$. Note that p_i may depend on the state of the communication channel in previous rounds. In general, p_i can be even chosen randomly from some distribution during the

execution of the protocol (finally, all nodes have to use, however, the same value p_i). Due to their simplicity and robustness, uniform algorithms are commonly used.

2.1 64-approximation

Function 1 `Broadcast`$(\hat{n})$

transmit with probability $1/\hat{n}$
return the status of the channel

Function 2 `Phase1`$()$

$l \leftarrow 0$
$u \leftarrow 1$
while `Broadcast`$(2^u) \neq$ NULL **do**
  $u \leftarrow 2u$
while $l + 1 < u$ **do**
  $m \leftarrow \lceil (l + u)/2 \rceil$
  if `Broadcast`$(2^m) =$ NULL **then**
    $u \leftarrow m$
  else
    $l \leftarrow m$
return u

Function 3 `Phase2`(u, L)

$\mathcal{M} \leftarrow [\,]$
for $k = 1$ **to** L **do**
  append u to $\mathcal{M}$
  status $\leftarrow$ `Broadcast`(2^u)
  if *status* $=$ NULL **then**
    $u \leftarrow \max(u - 3, 0)$
  else if *status* $=$ BEEP **then**
    $u \leftarrow u + 3$
return the most frequent value in $\mathcal{M}$

Algorithm 1 `SizeApprox1`(f)

$u \leftarrow$ `Phase1`$()$
$d \leftarrow \lceil (\log f + \log\log f + \log\log u + 5)/3 \rceil$
$L \leftarrow 100 \log(2f) + \lceil 125d/4 \rceil + 13$
$u \leftarrow$ `Phase2`(u, L)
return 2^u

Fig. 1. The pseudocode of a 64-approximation algorithm.

Phase 1 in the Algorithm is based on Leader Election Protocol by Nakano and Olariu [20]. Similarly, Phase 2 is a modification of a subprocedure used in [20]. Both phases make use of Broadcast function to determine (with a certain probability) if the current estimation of the network size is too high or too low. Intuitively, in Phase 1 nodes try to bound from above the network size by doubling the estimate until the status of the channel suggests that it is too high. In each round of Phase 2 nodes adjust the estimate by factor 8 according to the status of the channel. We should note here that the closer the estimate is to the real network size, the more probable it is that the decision based on an output of call to Broadcast is incorrect. Because of this, after Phase 2 we return the most common estimate. The following lemmas provide bounds on time complexity and accuracy of the returned estimator.

Lemma 1 (Nakano, Olariu [20]). *With probability exceeding* $1 - \frac{1}{2f}$ *,* `Phase1` *takes at most* $\mathcal{O}(\log\log n + \log f)$ *rounds after which the returned value,* u*, satisfies the double inequality*

$$\frac{n}{\ln(4(\lceil \log\log(4nf) \rceil + 1)f)} \leq 2^u \leq 4(\lceil \log\log(4nf) \rceil + 1)fn. \tag{1}$$

Let us introduce the following notation (we assume that $n \geq 2$). Parameters $p_\alpha^{(N)}$, $p_\alpha^{(B)}$ will denote probabilities of NULL and BEEP conditioned that the broadcast probability in the current round is $\min\{\frac{1}{\alpha n}, 1\}$. If $\alpha \cdot n > 1$, then

$$p_\alpha^{(N)} = \mathbb{P}\left(\text{NULL} \,|\, 2^u = \alpha \cdot n\right) = \left(1 - \frac{1}{\alpha \cdot n}\right)^n,$$
$$p_\alpha^{(B)} = \mathbb{P}\left(\text{BEEP} \,|\, 2^u = \alpha \cdot n\right) = 1 - \left(1 - \frac{1}{\alpha \cdot n}\right)^n,$$

where $1/2^u$ is the probability of transmission for each node and n is the real number of nodes. Otherwise, with $\alpha n \leq 1$ we set $p_\alpha^{(N)} = 0$ and $p_\alpha^{(B)} = 1$. For any fixed α we can bound the values of $p_\alpha^{(N)}, p_\alpha^{(B)}$ using basic inequalities. The following Proposition can be easily verified.

Proposition 1. *For $n \geq 25$ we have:*

1. $p_{1/8}^{(N)} \leq 0.06$,
2. $p_{8}^{(B)} \leq 0.12$,
3. $p_{1/64}^{(B)} \geq 0.99$,
4. $p_{64}^{(N)} \geq 0.98$.

In the following Lemma we analyze `Phase2` and show that Algorithm 1 is a 64-approximation.

Lemma 2. *If $n \geq 25$, then Algorithm 1 with probability at least $1 - 1/f$ returns value $\hat{n} = 2^u$ such that $n/64 \leq \hat{n} \leq 64 \cdot n$ in time $\mathcal{O}(\log \log n + \log f)$.*

Proof. Assume that u, after `Phase1` satisfies the double inequality from Lemma 1. We want to show that, conditioned on such an event, the approximation returned by Algorithm 1 is a 64-approximation with probability at least $1 - \frac{1}{2f}$. Thus we need to analyze `Phase2`. The phase can be seen as a biased random walk of length L on a line, where points on the line correspond to the values of the estimator 2^u and transition probabilities equal $p_\alpha^{(N)}$ and $p_\alpha^{(B)}$ (see Figure 2). Consider a sequence $\mathcal{U} = \{\dots, u_{-2}, u_{-1}, u_0, u_1, u_2, \dots\}$, such that $2^{u_0} \leq n < 2^{u_1}$

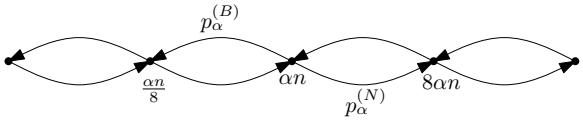

Fig. 2. An illustration of transition probabilities in `Phase2`.

and $u_{i+1} = u_i + 3$ for all $i \in \mathbb{Z}$. Let $\mathcal{P} = \{u_{-1}, u_0, u_1, u_2\}$. Let us call a

- *good step* – a step that starts and ends inside $\mathcal{P}$,
- *improving step* – a step that start outside $\mathcal{P}$ moving towards $\mathcal{P}$ (a NULL or BEEP such that the estimator after the step is better),
- *bad step* – a step that is leaving $\mathcal{P}$ or the one that starts outside set $\mathcal{P}$ moving further from $\mathcal{P}$.

We want to show that the state with the maximum number of visits will be a state from set $\mathcal{P}$, and thus the returned estimator will be a 64-approximation. Observe that during a good step an estimate from set $\mathcal{P}$ is added to set $\mathcal{M}$.

Denote by G, B, I the number of good, bad and improving steps during L steps of `Phase2`. By Lemma 1 the probability of a bad step is at most 0.12. Clearly, steps are dependent, however all the bounds for each step hold independently from other steps. Thus we can limit B by the sum of stochastically independent $0-1$ random variables and apply a Chernoff bound to get:

$$\mathbb{P}\left(B \geq 1.5 \cdot 0.12 \cdot L\right) \leq e^{1/12 \cdot 0.12 L} \leq \frac{1}{2f} .$$

Assume that $B < 0.12L$. Recall that d is the initial distance to set $\mathcal{P}$. Thus $G \leq B + d$. Since in a step (either good, bad or improving), the walk traverses an edge between two different states, the maximum number of visits to one state outside set $\mathcal{P}$ is at most

$$\left\lceil \frac{B}{2} \right\rceil + \left\lceil \frac{I}{2} \right\rceil \leq \frac{B+d}{2} + \frac{B}{2} + 2 = B + \frac{d}{2} + 2 \leq 0.18L + \frac{d}{2} + 2.$$

The total number of steps inside $\mathcal{P}$ is at least $L - I - B \leq L - (0.18L + d/2 + 2) = 0.82L - d/2 - 2$. Since $\mathcal{P}$ contains exactly four steps, there exists a step with at least $0.2L - d/6 - 2/3$

visits. Since the maximum number of visits to a state outside $\mathcal{P}$ is at most $0.18L + \frac{d}{2} + 2$, we need to show that

$$0.2L - d/8 - 1/4 \geq 0.18L + \frac{d}{2} + 2,$$

which is equivalent to

$$4L \geq 125d + 450\ .$$

We know from the definition of the algorithm that

$$L = 100\log(2f) + 125d/4 + 13 = 100\log f + 125d/4 + 113 > 125d/4 + 450/4.$$

Thus the state with the maximum number of visits is a state from set $\mathcal{P}$ which corresponds to a 64-approximation of the correct value of n. Now, by Lemma 1 with probability at least $1 - \frac{1}{2f}$, the total time of `Phase1` is $\mathcal{O}(\log\log n)$ and the value of u after the phase satisfies the double inequality (1). Conditioned on this event, with probability at least $1 - \frac{1}{2f}$ `Phase2` returns a 64-approximation. The time of `Phase2` is always $\mathcal{O}(\log f + \log\log\log n)$. Thus overall our algorithm returns u such that 2^u is a 64-approximation of n in time $\mathcal{O}(\log\log n + \log f)$ with probability at least $1 - \frac{1}{f}$.

2.2 A $(1+\varepsilon)$-approximation.

We now describe how to enhance the algorithm from the previous section with an additional phase to obtain a $(1+\varepsilon)$-factor approximation for any $\varepsilon > 0$. Intuitively, the procedure `Vote` checks whether the current estimate $\hat{n}$ is too big or too small. We let the nodes transmit with probability $1/\hat{n}$ for a fixed number of rounds. If our estimate is too small, a lot of nodes will transmit and there will not be enough silent rounds and thus we increase our estimate by a factor of $(1+\varepsilon)$. Similarly, if our estimate is too large, too many rounds will be silent and thus we decrease our estimate by a factor of $(1+\varepsilon)$.

Let $c = 1 + \varepsilon$, and denote $p_l = e^{-c}$ and $p_h = e^{-1/c}$.

Function 4 `Vote`$(\hat{n}, c, f)$

$p_l \leftarrow e^{-c}, \quad p_h \leftarrow e^{-1/c}$
$\delta_c \leftarrow (p_h - p_l)/(p_h + p_l)$
$\ell \leftarrow \lceil 3 \cdot e^3 \cdot \log f/\delta_c^2 \rceil$
$nulls \leftarrow 0$
for $i = 1$ **to** ℓ **do**
  if `Broadcast`$(\hat{n})$ = NULL **then**
    $nulls \leftarrow nulls + 1$
if $nulls < (1+\delta_c) \cdot p_l \cdot \ell$ **then**
  return UNDERSTIMATED
else
  return OVERESTIMATED

Function 5 `Refine`$(\hat{n}, c, f)$

if `Vote`$(\hat{n}, c^{1/2}, f)$ = UNDERSTIMATED **then**
  return $c^{1/4} \cdot \hat{n}$
else
  return $c^{-1/4} \cdot \hat{n}$

Function 6 `Phase3`$(\hat{n}, f)$

$f' \leftarrow 14f$
$initial \leftarrow$ `Vote`$(\hat{n}, \sqrt{2}, f')$
if *initial*=UNDERSTIMATED **then**
  $\phi \leftarrow \sqrt{2}$
else
  $\phi \leftarrow 1/\sqrt{2}$
for $i = 1$ **to** 13 **do**
  $\hat{n} \leftarrow \phi \cdot \hat{n}$
  if $initial \neq$ `Vote`$(\hat{n}, \sqrt{2}, f')$ **then**
    return $\hat{n}$
return $\hat{n}$

Algorithm 2 `SizeApprox2`(f, c)

$\hat{n} \leftarrow$ `SizeApprox1`$(4f)$
$\hat{n} \leftarrow$ `Phase3`$(\hat{n}, 4f)$
$t \leftarrow \lceil \log_{4/3} \log_c 2 \rceil$
for $i = t$ **downto** 1 **do**
  $\hat{n} \leftarrow$ `Refine`$(\hat{n}, c^{(4/3)^i}, 2^{i+1}f)$
return $\hat{n}$

Fig. 3. The pseudocode of c-approximation algorithm.

We have:

$$\mathbb{P}\left(\text{NULL}|\hat{n} \geq cn\right) \geq \left(1 - \frac{1}{cn}\right)^n \geq e^{-1/c}\left(1 - \frac{1}{cn}\right) \geq p_h/2. \tag{2}$$

$$\mathbb{P}\left(\text{NULL}|\hat{n} \leq n/c\right) \leq \left(1 - \frac{c}{n}\right)^n \leq \leq e^{-c} = p_l \tag{3}$$

Thus $p_h/2$ upper bounds the probability of NULL in a round under the condition that approximation $\hat{n}$ is c times too high. On the other hand p_l lowerbounds the probability of NULL in a round conditioned that $\hat{n}$ is c times too low.

Denote $\delta = \frac{p_h - p_l}{p_h + p_l}$, and observe that for such δ we have

$$p_h/2\left(1-\delta\right) = p_l\left(1+\delta\right). \tag{4}$$

Moreover since $p_h - p_l = e^{-1/c} - e^{-c} > 0$, then $\delta > 0$. Observe also that $\delta < 1/2$.

In the following lemmas we bound the probability that procedure `Vote` returns OVERESTIMATED and UNDERSTIMATED, assuming that estimator $\hat{n}$ deviates from n by factor c. We note that in all calls to `Vote` in the algorithm the inequality $c < 3$ holds.

Lemma 3. *If $\hat{n} < n/c$, then procedure `Vote`$(\hat{n}, c, f)$ returns UNDERSTIMATED with probability at least $1 - \frac{1}{f}$.*

Proof. By (3), the probability that no node transmits is upperbounded by p_l. Let X_i denote the random variable that is 0 if at least one node transmits and 1 otherwise. Thus, if we let the nodes transmit ℓ times, we obtain as expected value for $X = \sum_{i=1}^{\ell} X_i$, $E\left[X\right] \leq \ell \cdot p_l$. Chernoff bound yields:

$$\mathbb{P}\left(X \geq (1+\delta)\, p_l l\right) = \mathbb{P}\left(X \geq (1+\delta)\left(1 + \frac{p_l l - E[X]}{E[X]}\right) E\left[X\right]\right) \leq e^{-\frac{((1+\delta)p_l l - E[X])^2}{E[X]}}.$$

We know that $E[X] \leq p_l l$ hence $((1+\delta)p_l l - E[X])^2 \geq (\delta p_l l)^2$. Since $\ell \geq \frac{3}{\delta^2} e^3 \log f$, then $\delta^2 p_l l \geq \log f$ hence $((1+\delta)p_l l - E[X])^2 \geq E[X] \log f$ and $\mathbb{P}\left(X \geq (1+\delta)\, p_l l\right) \leq \frac{1}{f}$. Thus, with probability at least $1 - 1/f$, variable *nulls* in procedure `Vote` satisfies $nulls < (1+\delta) \cdot p_l \cdot \ell$. Thus `Vote` returns UNDERSTIMATED with probability at least $1 - 1/f$.

Lemma 4. *If $\hat{n} > cn$, then procedure `Vote`$(\hat{n}, c, f)$ returns OVERESTIMATED with probability at least $1 - \frac{1}{f}$.*

Proof. By (2), the probability that no node transmits is lowerbounded by $p_h/2$. Let X_i denote the random variable that is 0 if at least one node transmits and 1 otherwise. Thus, if we let the nodes transmit ℓ times, we obtain as expected value for $X = \sum_{i=1}^{\ell} X_i$, $E\left[X\right] \geq \ell \cdot p_h/2$. Chernoff bound yields:

$$\mathbb{P}\left(X \leq (1+\delta)\, p_l l\right) = \mathbb{P}\left(X \leq (1-\delta)\, p_h l/2\right) \leq \mathbb{P}\left(X \leq (1-\delta)\, E[X]\right) \leq e^{-\frac{\delta^2}{2} E[X]} \leq \frac{1}{f}.$$

This holds for $\ell \geq \frac{3}{\delta^2} e^3 \log f$, since $p_h > e^{-1}$. Thus with probability at least $1 - 1/f$, variable *nulls* does not satisfy the condition after **if**, thus `Vote` returns OVERESTIMATED with probability at least $1 - 1/f$.

Lemma 5. *If $\hat{n}$ is a 64-approximation of the number of nodes n, then procedure `Phase3`$(\hat{n}, f)$ returns a 2-approximation of n with probability at least $1 - 1/f$ using $\mathcal{O}\left(\log f\right)$ slots.*

Proof. We call an execution of `Vote`$(\hat{n}, \sqrt{2}, 14f)$ successful if it:

- returns OVERESTIMATED when $\hat{n} \geq \sqrt{2}n$,
- returns UNDERSTIMATED when $\hat{n} \leq n/\sqrt{2}$.

Procedure `Phase3` makes at most 14 calls to `Vote` and by Lemmas 3 and 4 each call is successful with probability at least $1 - 1/(14f)$. Therefore the probability that all calls are successful is at least $1 - 1/f$.

We want to argue that if all calls to procedure `Vote` are successful, then we obtain 2-approximation. If $\hat{n} \geq \sqrt{2}n$, then the first call to `Vote` returns OVERESTIMATED and we start decreasing the estimate. After at most $\log_{\sqrt{2}} 64 + 1 = 13$ iterations, the value $\hat{n}$ satisfies $\hat{n} \leq n/\sqrt{2}$ and `Vote` returns UNDERSTIMATED. The returned estimator is a 2-approximation of n because we divide the estimator by $\sqrt{2}$ until it is at most $n/\sqrt{2}$ for the first time. We make similar argument if the initial estimate is too small, i.e., $\hat{n} \leq n/\sqrt{2}$. If the initial estimate is correct, then after making at most 2 increases we will obtain an estimate that is at least $\sqrt{2}$ times too big, thus the third call to `Vote` returns OVERESTIMATED and we finish the procedure. Using the same argument as above we can show that the returned estimator is a 2-approximation. Similarly, if the initial value is correct, we make at most 2 decreases.

Each call to `Vote`$(\hat{n}, \sqrt{2}, 14f)$ requires $\mathcal{O}(\log f)$ slots.

Lemma 6. *If $\hat{n}$ is a c-approximation of the number of nodes then procedure* ***`Refine`***$(\hat{n}, c, f)$ *returns $c^{3/4}$-approximation with probability at least $1 - 1/f$ using $\mathcal{O}(\log f/\varepsilon^2)$ slots.*

Proof. Observe that if $\hat{n}$ is already a $c^{1/2}$-approximation, then regardless of the output of `Vote` we obtain a $c^{3/4}$-approximation.

On the other hand if $cn \geq \hat{n} \geq c^{1/2}n$, then by Lemma 4, with probability at least $1 - 1/f$, procedure `Vote` returns OVERESTIMATED and we decrease the estimate by factor of $c^{1/4}$. Finally if $n/c \leq \hat{n} \leq c^{-1/2}n$, then with probability at least $1 - 1/f$, by Lemma 3 `Vote` returns UNDERSTIMATED and we increase the estimate by factor of $c^{1/4}$.

To bound the time complexity of procedure `Refine` we need to bound the number of steps of procedure `Vote`. With $c = 1 + \varepsilon$ and $\varepsilon > 0$ we have

$$\delta_{1+\varepsilon} = \frac{e^{-\frac{1}{1+\varepsilon}} - e^{-(1+\varepsilon)}}{e^{-\frac{1}{1+\varepsilon}} + e^{-(1+\varepsilon)}} = \frac{e^{-1}}{e^{-1}} \frac{e^{\frac{\varepsilon}{1+\varepsilon}} - e^{-\varepsilon}}{e^{\frac{\varepsilon}{1+\varepsilon}} + e^{-\varepsilon}} \geq \frac{e^{\frac{\varepsilon}{1+\varepsilon}} - e^{-\frac{\varepsilon}{1+\varepsilon}}}{e^{\frac{\varepsilon}{1+\varepsilon}} + e^{-\frac{\varepsilon}{1+\varepsilon}}} = \tanh\left(\frac{\varepsilon}{1+\varepsilon}\right).$$

Therefore

$$\delta^{-2} \leq \coth^2\left(\frac{\varepsilon}{1+\varepsilon}\right) = 1 + \frac{1}{\sinh^2\left(\frac{\varepsilon}{1+\varepsilon}\right)} \leq \frac{1}{\varepsilon^2} + \frac{2}{\varepsilon} + 2,$$

where the last inequality is the result of $\sinh(x) \geq x$ for $x \geq 0$. Hence $\delta_\varepsilon^{-2} = O(\varepsilon^{-2})$ as $\varepsilon \to 0$. We call procedure `Vote` with $c^{1/2} = (1+\varepsilon)^{1/2} \geq 1 + \varepsilon/4$ for $\varepsilon < 1$. Hence the complexity of a single execution of procedure `Vote` is $O(\varepsilon^{-2} \log f)$.

Theorem 1. *For $\varepsilon > 0$ algorithm* ***`SizeApprox2`***$(f, 1+\varepsilon)$ *returns $(1+\varepsilon)$-approximation of number of nodes with probability at least $1 - 1/f$ using $\mathcal{O}\left(\frac{\log f}{\varepsilon^2} + \log\log n\right)$ slots.*

Proof. With probability at least $1 - 1/(4f)$ call to `SizeApprox1` returns 64-approximation, which we turn into 2-approximation with probability at least $1 - 1/(4f)$ by calling `Phase3`. Next, we refine the approximation using $t = \lceil \log_{4/3} \log_{1+\varepsilon} 2 \rceil$ iterations. The probability of failure of the i-th iteration is at most $1/(2^{i+1}f)$, for $1 \leq i \leq t$. Therefore, by union bound, the probability of failure of the `SizeApprox2` is at most

$$\frac{1}{4f} + \frac{1}{4f} + \frac{1}{2f} \cdot \sum_{i=1}^{t} 2^{-i} \leq \frac{1}{f}.$$

Assuming that none of the `Vote` calls failed we compute the quality of the resulting estimate. We can show by induction using Lemma 6 that after i-th iteration of the loop in algorithm `SizeApprox2`, the current estimate $\hat{n}$ is a $(1+\varepsilon)^{(4/3)^{i-1}}$-approximation. Hence after t iterations we get a $(1+\varepsilon)$-approximation.

By Lemma 6 the number of slots used by t iterations is

$$\sum_{i=1}^{t} \mathcal{O}\left(\frac{\log(2^{i+1}f)}{\varepsilon^2(4/3)^{2i}}\right) \leq \sum_{i=1}^{\infty} \mathcal{O}\left(\frac{\log(2^{i+1}f)}{\varepsilon^2(4/3)^{2i}}\right) = \mathcal{O}\left(\frac{\log f}{\varepsilon^2}\right),$$

where the last inequality is justified by the fact that the $\mathcal{O}(\cdot)$ notation from Lemma 6 holds uniformly (i.e., the hidden constant is independent from f, i and ε).

Adding the slots used by `SizeApprox1` and `Phase3` we get the final time complexity.

3 Lower Bound

In this section we show that any (not necessarily uniform) size estimation algorithm returning a $(1+\varepsilon)$-approximation of the number of nodes with probability at least $1-1/f$ works in time $\Omega(\log\log n + \frac{\log f}{\varepsilon})$.

We start the analysis of beeping model by showing how the execution by different number of nodes relates to each other. Namely, we prove that (in probability) history of the channel state observed in case of n and m nodes performing any randomized protocol are similar for n close to m. We subscript symbol $\mathbb{P}$ with n to denote probability conditioned on the number of nodes running some algorithm, $\mathbb{P}_n(A) = \mathbb{P}(A \,|\, |\mathcal{N}| = n)$ for any event A. For a vector $h \in \{\text{NULL}, \text{BEEP}\}^t$ we write $\mathbb{P}(h)$ to denote the probability that during the first t slots of the execution of algorithm the global history of channel is h.

Lemma 7. *Let $\mathcal{A}$ be any randomized algorithm for a single-hop radio network with beeping communication model. For a global history of channel state, $h \in \{NULL, BEEP\}^*$ and $m \geq n \geq 1$, there is $\mathbb{P}_m(h) \geq (\mathbb{P}_n(h))^{m/n}$.*

Proof. We proceed with a coupling argument. Let us consider a set $\mathcal{S} = \{s_1, \ldots, s_{nm}\}$ consisting of nm nodes. Even though the nodes are indistinguishable, for the purpose of analysis we can identify them by the random sources they use. That is, we assume that node s_i has access to an infinite sequence of random bits $\boldsymbol{X}_i = X_i^{(1)}, X_i^{(2)}, \ldots$. Clearly, if $\boldsymbol{X}_i = \boldsymbol{X}_j$, then nodes s_i and s_j behave identically during an execution of any algorithm (of course $\mathbb{P}(\boldsymbol{X}_i = \boldsymbol{X}_j) = 0$ for $i \neq j$). We partition $\mathcal{S}$ in two different ways – into n independent networks $\mathcal{N}_1, \ldots, \mathcal{N}_n$ with m nodes each (called big networks) and m independent networks $\mathcal{N}'_1, \ldots, \mathcal{N}'_m$ with n nodes each (called small networks). We require that for each big network $\mathcal{N}_i$ there exists at least one small network $\mathcal{N}'_j$ such that $\mathcal{N}'_j \subseteq \mathcal{N}_i$. We assume that all networks are independent from each other, i.e., there are no interferences of communication channels. In these two settings, however, each node from $\mathcal{S}$ belongs to exactly one big and one small network and in both cases uses the same random source for making its decisions. Our goal is to compare the execution of algorithm $\mathcal{A}$ performed by the same nodes grouped into big and small networks.

Let $\boldsymbol{H}_1, \ldots, \boldsymbol{H}_n$ and $\boldsymbol{H}'_1, \ldots, \boldsymbol{H}'_m$ denote global histories of channel states during the executions of algorithm $\mathcal{A}$ by big and small networks, respectively. We are going to show by induction on h's length that if h is a prefix of channel histories of all small networks, $\boldsymbol{H}'_1, \ldots, \boldsymbol{H}'_m$, then it is also a prefix of channels histories of all big networks, $\boldsymbol{H}_1, \ldots, \boldsymbol{H}_n$. The base case of empty string $h = \varepsilon$ holds trivially. Therefore, let us assume that the statement is true for all global histories of length $t \geq 0$ and that $h = h_1, h_2, \ldots, h_t, h_{t+1}$ is a prefix of channel histories of small networks. By induction, $h_1, h_2, \ldots, h_t$ is a prefix of each $\boldsymbol{H}_1, \ldots, \boldsymbol{H}_n$. At the beginning of the $(t+1)$-st slot each node decides whether to transmit or not based on its random source, local history and the global history $h_1, h_2, \ldots, h_t$. However, in this case the local

history is redundant as it can be reconstructed from $\boldsymbol{X}_i$ and the global history. Therefore, if in the $(t+1)$st slot the resulting channel states of each small network are $h_{t+1} = \text{NULL}$,

$$H_1'^{(t+1)} = \ldots = H_m'^{(t+1)} = \text{NULL},$$

then all nodes decided not to transmit and

$$H_1^{(t+1)} = \ldots = H_n^{(t+1)} = \text{NULL}.$$

Otherwise, if

$$H_1'^{(t+1)} = \ldots = H_m'^{(t+1)} = \text{BEEP},$$

then in every small network there is at least one node that decided to transmit during the $(t+1)st$ slot. For each big network $\mathcal{N}_i$ there is some small network $\mathcal{N}_j' \subseteq \mathcal{N}_i$, hence $H_j'^{(t+1)} = \text{BEEP}$ implies $H_i^{(t+1)} = \text{BEEP}$. Therefore, h is a prefix of $\boldsymbol{H}_1, \ldots, \boldsymbol{H}_n$. Finally, all networks are independent, thus

$$\begin{aligned}(\mathbb{P}_n(h))^m &= \mathbb{P}\left(\boldsymbol{H}_1' \text{ starts with } h \wedge \ldots \wedge \boldsymbol{H}_m' \text{ starts with } h\right)\\ &\leq \mathbb{P}\left(\boldsymbol{H}_1 \text{ starts with } h \wedge \ldots \wedge \boldsymbol{H}_n \text{ starts with } h\right)\\ &= (\mathbb{P}_m(h))^n.\end{aligned}$$

Lemma 8. *For any non-empty finite set of global histories of channel state $H \subseteq \{NULL, BEEP\}^*$ and $m > n \geq 1$ there is*

$$\mathbb{P}_m(H) \geq \frac{(\mathbb{P}_n(H))^{m/n}}{|H|^{m/n-1}}.$$

Proof. By Lemma 7 we get

$$\mathbb{P}_m(H) = \sum_{h\in H} \mathbb{P}_m(h) \geq \sum_{h\in H} (\mathbb{P}_n(h))^{m/n}.$$

Using Hölder inequality

$$\sum_{i=1}^{n} |x_i y_i| \leq \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p} \cdot \left(\sum_{i=1}^{n} |y_i|^q\right)^{1/q}$$

with $p = m/n$ and $q = m/(m-n)$ we obtain

$$\sum_{h\in H} (\mathbb{P}_n(h))^p = \frac{1}{|H|^{p/q}} \left(\sum_{h\in H} 1^q\right)^{p/q} \cdot \sum_{h\in H} (\mathbb{P}_n(h))^p \geq \frac{1}{|H|^{p/q}} \left(\sum_{h\in H} \mathbb{P}_n(h)\right)^p = \frac{(\mathbb{P}_n(H))^{m/n}}{|H|^{m/n-1}}.$$

As we stated in Section 1.1, in any algorithm $\mathcal{A}$ the decision whether to stop the execution after the current slot and what estimation to return is based only on the global history of channel state. For any history $h \in \{\text{NULL}, \text{BEEP}\}^*$ that causes nodes to finish the execution of $\mathcal{A}$ we denote by $\mathcal{A}(h)$ the estimated network size returned by $\mathcal{A}$.

Theorem 2. *Let $\mathcal{A}$ be a size estimation algorithm for a single-hop radio network assuming the beeping communication model. If for any network size n algorithm $\mathcal{A}$ returns $(1+\varepsilon)$-approximation with probability at least $1 - 1/f$ and within at most T_n time slots (T_n non-decreasing), then*

$$T_n \geq \max\left\{\frac{\log_2 f + (1+\varepsilon)^2 \log_2(1-1/f)}{(1+\varepsilon)^2 + 1/n - 1}, \log_2\log_2(1 + 2\varepsilon n + \varepsilon^2 n) - \log_2\log_2(1+\varepsilon) - 1\right\}.$$

Proof. For $k \in \mathbb{N}_+$ let

$$H_k = \{h \in \{\text{NULL}, \text{BEEP}\}^* : |h| \leq T_k, \frac{k}{1+\varepsilon} \leq \mathcal{A}(h) \leq (1+\varepsilon)k\}$$

be a set of all global histories of length at most T_k for which the value returned by algorithm $\mathcal{A}$ is a $(1+\varepsilon)$-approximation of k. Clearly, $\mathbb{P}_k(H_k) \geq 1 - 1/f$. Let $m = \lfloor (1+\varepsilon)^2 n + 1 \rfloor$, so that $m/(1+\varepsilon) > (1+\varepsilon)n$ and thus $H_n \cap H_m = \emptyset$. This way,

$$\mathbb{P}_m(H_n) \leq 1 - \mathbb{P}_m(H_m) \leq 1/f.$$

On the other hand by Lemma 8 there is

$$\mathbb{P}_m(H_n) \geq \frac{(\mathbb{P}_n(H_n))^{m/n}}{|H_n|^{m/n-1}} \geq \frac{(1-1/f)^{m/n}}{|H_n|^{m/n-1}}.$$

Therefore,

$$|H_n| \geq \left(f(1-1/f)^{m/n}\right)^{\frac{1}{m/n-1}}.$$

We know that set H_n contains words of length at most T_n and no word is a prefix of another, so $|H_n| \leq 2^{T_n}$. Finally, we get

$$T_n \geq \log_2 |H_n| \geq \frac{\log_2 f + \frac{m}{n}\log_2(1-1/f)}{m/n - 1} \geq \frac{\log_2 f + (1+\varepsilon)^2 \log_2(1-1/f)}{(1+\varepsilon)^2 + 1/n - 1}.$$

Now, let $a_1 = 1$ and

$$a_i = \lfloor (1+\varepsilon)^2 a_{i-1} + 1 \rfloor \leq (1+\varepsilon)^2 a_{i-1} + 1 \leq \frac{(1+\varepsilon)^{2i} - 1}{(1+\varepsilon)^2 - 1}.$$

All sets H_{a_i} must be non-empty and pairwise disjoint. Because T_n is non-decreasing, we have

$$\left| \bigcup_{i:\ a_i \leq n}^{\cdot} H_{a_i} \right| \leq 2^{T_n}.$$

For

$$i \leq \frac{\log_2(((1+\varepsilon)^2 - 1)n + 1)}{2\log_2(1+\varepsilon)}$$

there is $a_i \leq n$. Therefore,

$$T_n \geq \log_2 \log_2(1 + 2\varepsilon n + \varepsilon^2 n) - \log_2 \log_2(1+\varepsilon) - 1.$$

Remark 1. For $\varepsilon \to 0$ and $f \geq 2$ we get

$$T_n = \Omega\left(\frac{\log f}{2\varepsilon + 1/n} + \log\log n\right).$$

For a constant ε (independent of n and f) there is

$$T_n = \Omega(\log f + \log\log n).$$

4 Final Remarks

We presented an algorithm for $(1+\varepsilon)$-approximation of the size of a single-hop radio network with Beeping Model that needs $\mathcal{O}\left(\log\log n + \log f/\varepsilon^2\right)$ time slots, wherein n is the real number of nodes and $1/f$ is the probability of failure. We also proved the matching lower bound for a constant ε. In some subprocedures we used quite big constants for the sake of technical simplicity of the analysis. As a future work we leave improving all those parameters. We believe that they can be significantly lowered to make the protocol practical for real-life scenarios already for moderate n.

References

1. Y. Afek, N. Alon, Z. Bar-Joseph, A. Cornejo, B. Haeupler, and F. Kuhn. Beeping a maximal independent set. *Distributed Computing*, 26(4):195–208, 2013.
2. J. L. Bordim, J. Cui, T. Hayashi, K. Nakano, and S. Olariu. Energy-efficient initialization protocols for ad-hoc radio networks. In *Algorithms and Computation*, pages 215–224. Springer, 1999.
3. I. Caragiannis, C. Galdi, and C. Kaklamanis. Basic computations in wireless networks. In X. Deng and D. Du, editors, *Algorithms and Computation, 16th International Symposium, ISAAC 2005, Sanya, Hainan, China, December 19-21, 2005, Proceedings*, volume 3827 of *Lecture Notes in Computer Science*, pages 533–542. Springer, 2005.
4. A. Chakrabarti and O. Regev. An optimal lower bound on the communication complexity of gap-hamming-distance. In L. Fortnow and S. P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 51–60. ACM, 2011.
5. P. Chassaing and L. Gerin. Efficient estimation of the cardinality of large data sets. In *4th Colloquium on Mathematics and Computer Science*, pages 419–422. DMTCS Proceedings, 2006.
6. B. Chen, Z. Zhou, and H. Yu. Understanding RFID counting protocols. In S. Helal, R. Chandra, and R. Kravets, editors, *The 19th Annual International Conference on Mobile Computing and Networking, MobiCom'13, Miami, FL, USA, September 30 - October 04, 2013*, pages 291–302. ACM, 2013.
7. J. Cichoń, J. Lemiesz, W. Szpankowski, and M. Zawada. Two-phase cardinality estimation protocols for sensor networks with provable precision. In *Proceedings of WCNC'12*, Paris, France, 2012. IEEE.
8. J. Cichoń, J. Lemiesz, and M. Zawada. On size estimation protocols for sensor networks. In *Proceedings of the 51th IEEE Conference on Decision and Control, CDC 2012, December 10-13, 2012, Maui, HI, USA*, Proceedings of 51st Annual Conference on Decision and Control (CDC), pages 5234–5239. IEEE, 2012.
9. A. Cornejo and F. Kuhn. Deploying wireless networks with beeps. In *DISC*, pages 148–162, 2010.
10. P. Flajolet, E. Fusy, O. Gandouet, and F. Meunier. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In *Proceedings of the Conference on Analysis of Algorithms (AofA'07)*, pages 127–146, 2007.
11. P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31(2):182–209, 1985.
12. F. Giroire. Order statistics and estimating cardinalities of massive data sets. *Discrete Applied Mathematics*, 157(2):406–427, 2009.
13. A. G. Greenberg, P. Flajolet, and R. E. Ladner. Estimating the multiplicities of conflicts to speed their resolution in multiple access channels. *J. ACM*, 34(2):289–325, Apr. 1987.
14. H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu. Counting RFID tags efficiently and anonymously. In *INFOCOM 2010. 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 15-19 March 2010, San Diego, CA, USA*, pages 1028–1036. IEEE, 2010.
15. T. Jurdzinski, M. Kutyłowski, and J. Zatopianski. Energy-efficient size approximation of radio networks with no collision detection. In *Proceedings of COCOON '02*, pages 279–289. Springer-Verlag, 2002.
16. J. Kabarowski, M. Kutyłowski, and W. Rutkowski. Adversary immune size approximation of single-hop radio networks. In *Theory and Applications of Models of Computation*, volume 3959 of *LNCS*, pages 148–158. Springer, 2006.
17. M. S. Kodialam and T. Nandagopal. Fast and reliable estimation schemes in RFID systems. In M. Gerla, C. Petrioli, and R. Ramjee, editors, *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking, MOBICOM 2006, Los Angeles, CA, USA, September 23-29, 2006*, pages 322–333. ACM, 2006.
18. M. S. Kodialam, T. Nandagopal, and W. C. Lau. Anonymous tracking using RFID tags. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 6-12 May 2007, Anchorage, Alaska, USA*, pages 1217–1225. IEEE, 2007.
19. K. Nakano and S. Olariu. Energy-efficient initialization protocols for single-hop radio networks with no collision detection. *Parallel and Distributed Systems, IEEE Transactions on*, 11(8):851–863, 2000.
20. K. Nakano and S. Olariu. Uniform leader election protocols for radio networks. *IEEE Trans. Parallel Distrib. Syst.*, 13(5):516–526, 2002.
21. C. Qian, H. Ngan, Y. Liu, and L. M. Ni. Cardinality estimation for large-scale RFID systems. *IEEE Trans. Parallel Distrib. Syst.*, 22(9):1441–1454, 2011.
22. M. Shahzad and A. X. Liu. Every bit counts: fast and scalable RFID estimation. In Ö. B. Akan, E. Ekici, L. Qiu, and A. C. Snoeren, editors, *The 18th Annual International Conference on Mobile Computing and Networking, Mobicom'12, Istanbul, Turkey, August 22-26, 2012*, pages 365–376. ACM, 2012.
23. K.-Y. Whang, B. T. V. Zanden, and H. M. Taylor. A linear-time probabilistic counting algorithm for database applications. *ACM Trans. Database Syst.*, 15(2):208–229, 1990.
24. D. E. Willard. Log-logarithmic selection resolution protocols in a multiple access channel. *SIAM J. Comput.*, 15(2):468–477, 1986.
25. Y. Zheng and M. Li. ZOE: fast cardinality estimation for large-scale RFID systems. In *Proceedings of the IEEE INFOCOM 2013, Turin, Italy, April 14-19, 2013*, pages 908–916. IEEE, 2013.