Two-Phase Cardinality Estimation Protocols for Sensor Networks with Provable Precision

Jacek Cichoń*, Jakub Lemiesz*, Wojciech Szpankowski† and Marcin Zawada*
*Faculty of Fundamental Problems of Technology
Wroclaw University of Technology
Poland
†Department of Computer Sciences
Purdue University, U.S.A.

***Abstract*—Efficient cardinality estimation is a common requirement for many wireless sensor network (WSN) applications. The task must be accomplished at extremely low overhead due to severe sensor resource limitation. This poses an interesting challenge for large-scale WSNs. In this paper we present a two-phase probabilistic algorithm based on order statistics and Bernoulli scheme, which effectively estimates the cardinality of WSNs. We thoroughly examine properties of estimators used in each phase as well as the precision of the whole procedure. The algorithm discussed in this paper is a modification of a recently published idea - the modification enables us to obtain a provable precision.**

I. Introduction

Most important feature of WSNs is their distributed nature. In such an architecture, sensor knowledge is limited to its collection of neighbors. This architecture has good scalability properties and does not suffer from central station becoming a single point of failure. On the other hand, it makes it increasingly difficult to observe some characteristics that would be straightforward to monitor in a centralized system. One example of such characteristic, which is required by the most fundamental WSN algorithms, like initialization or data aggregation, is the network cardinality (i.e. the number of devices connected to the network).

The cardinality estimation task poses several major challenges for large-scale WSNs. For example, the most straightforward method would be counting of distinct sensor identifiers. However, this simple approach is not desirable due to limited memory available to a sensor. Notice that for widely used sensors (e.g., MICA2DOT) based on Atmel's ATmega128 with available 4kB memory for data, such method enables us to count up to a thousand of sensors with 32-bit sensor identifiers. Moreover, the counting of identifiers requires at least logarithmic time to insert or check a newly arrived sensor and entails high communication cost (see Section IV).

In this paper, we present a probabilistic algorithm for approximate counting that uses tiny fraction of sensor memory (100 bytes), requires constant time to insert and check a newly arrived sensor and has much lower communication cost.

This research was supported by Polish Ministry of Science and Higher Education grant N N206 369739

Related Work: The problem of probabilistic counting is well studied in the literature (e.g. [1], [2], [3], [4]). Unfortunately, almost all of these techniques are designed to count elements in huge data sets and our simulations have shown that their accuracy is highly unsatisfactory if they are used (unmodified and separately) for a typical WSN size.

Recently, a part of authors of this paper proposed a probabilistic algorithm for WSN size estimation [5] that combines techniques of order statistics ([4], [6]) with balls and bins model ([2], [3]) modified by preselection. However, in this algorithm a modified balls and bins model was used and it was hard to obtain analytical confirmation of the algorithm's accuracy. In this paper, **we replaced the balls and bins model with Bernoulli scheme** and thanks to the work of Flajolet [7] as well as Jacquet and Szpankowski [8], we are able to confirm the high precision of the estimation (see Subsection II-B) without sacrificing the efficiency of the algorithm. Moreover, we are able to establish confidence intervals for the estimation returned by our algorithm. This in turn enables us to set desired accuracy by adjusting the parameter determining memory consumption.

In Section VI we compare our solution to Extrema Propagation, a randomized counting algorithm that was previously proposed ([6], [9]) for distributed systems.

II. Two-phase Algorithm

In this section, we present our two-phase algorithm. To make the exposition clearer, we describe an intuitive way of looking at the algorithm. In the first phase (see Subsection II-A) the order statistic is used to quickly obtain a rough estimate of the network size. Next, in the second phase (see Subsection II-B) we obtain a more accurate approximation on the basis of the number of successes in a series of Bernoulli trials. The probability of the success in a trial depends on the result of the first phase. Finally, we investigate the precision of the estimation in each phase separately as well as two phases combined.

A. First Phase: Order Statistics

Detailed description of this phase can be found in [5]. Here we will only briefly remind the reader of its major

principles and emphasize the essential feature, which has not been mentioned earlier.

Counting in this phase is based on order statistics. Initially each of n sensors generates a number from the unit interval uniformly at random. Let $X_{k:n}$ denote the kth order statistic of these numbers and let

$$\mathcal{Z}_{k,n} = \frac{k-1}{X_{k:n}} \ .$$

The main idea is that at the end of the phase each sensor knows the value of $X_{k:n}$ and uses $\mathcal{Z}_{k,n}$ to estimate the network cardinality n. The value of the parameter k determines the precision of the estimation. Details are presented in pseudocode (see Algorithm 1). Note that if n is smaller than k, then each sensor will know the exact value of n.

It is worth of mentioning that most of counting algorithms have strong limitation, namely they require some knowledge of n in advance. For example, in counting algorithms based on balls and bins model ([2], [5]) it is essential to know some reasonable upper bound on the number of balls to set the number of bins. Algorithm 1 theoretically works regardless of the network size n. In practice, we should avoid the situation when two sensor draw the same initial value, which is possible because of the machine representation of real numbers. By using 5-bytes representation we ensure proper functioning of Algorithm 1 for $n \leq 10^6$ (due to the Birthday Paradox, see [5]). Then, for $k = 20$ the algorithm requires only 100 bytes of memory on each device. Presently, we recall some facts concerning the estimator $\mathcal{Z}_{k,n}$.

Theorem 1. Suppose that $3 \leq k \leq n$ and $0 < \delta_1 < \frac{k-1}{k} < \delta_2$. Let $f(x) = -xe^{-x+1}$ and

$$F_k(\delta_1, \delta_2) = f\left(\frac{k-1}{k\delta_1}\right)^k + f\left(\frac{k-1}{k\delta_2}\right)^k$$

then

$$\Pr[\delta_1 n < \mathcal{Z}_{k,n} < \delta_2 n] > 1 - F_k(\delta_1, \delta_2) \ .$$

The proof of Theorem 1 can be found in [5]. The main idea is to reduce properties of order statistics to the Bernoulli distribution and then employ the Chernoff inequalities. The only change we introduce in this paper is the use of the stronger version of the Chernoff's theorem (see Lemma 1).

Corollary 1. Suppose that $k = 20$ and $n \geq 20$. Then by setting $\delta_1 = 0.5$ and $\delta_2 = 2.21$ from Theorem 1 we get $\Pr[0.5n < \mathcal{Z}_{20,n} < 2.21n] \geq 0.99$.

Remark 1. Numerical calculations with the incomplete regularized Beta functions show that for all $n \leq 10^7$ we have $\Pr[0.5n < \mathcal{Z}_{20,n} < 2n] \geq 0.99975$.

B. Second Phase: Bernoulli Trials

The second phase of our algorithm is based on Bernoulli trials (see Algorithm 2). At first, each sensor allocates a bitmap T of size m, which represents m trials, and initializes all entries to "0"s. For each trial i, a sensors decides with probability p whether it will participate. If so, it sets ith bit of T to "1" and broadcasts i to all neighbors. Upon receiving i, the sensor checks whether $T[i] = 0$; if so, it changes the value of $T[i]$ to "1" and forwards the number to its neighbors. Otherwise, it does nothing and goes to sleep. Notice that if a sensor receives the same number twice then the proper bit is already set so it does not forward the number further. As it may happen that two sensors decide to participate in the same trial, this feature allows to decrease the amount of transmitted packets. At any time, each sensor can get the estimated number of sensors by calculating $\log(\hat{x}/m)/\log(1-p)$, where $\hat{x}$ is the number of trials that result in "0" (trials when none of the sensors decided to participate in).

Further, we present the derivation and analysis of the estimator used in Algorithm 2. Let us assume that we have n sensors and m trials. Let us fix a probability $p \in (0,1)$. Then, we consider the following process: each sensor decides to participate in each trial with probability p. If none of the sensors decides to participate in a trial, the trial is considered a success. Let random variable $Y_{m,n}$ denote the number of successful trials at the end of the process. We easily obtain the expectation of random variable $Y_{m,n}$:

$$\mathbf{E}\left[Y_{m,n}\right] = m\left(1-p\right)^n \ .$$

To derive an estimator of the number n, we use the method of moments. We solve the equation $Y_{m,n} = m(1-p)^n$ for n and we define a random variable $\mathcal{W}_{m,n}$ to be the estimator:

$$\mathcal{W}_{m,n} = \frac{\log(\frac{Y_{m,n}}{m})}{\log(1-p)} \ .$$

Algorithm 1 ORDERSTATISTICS(k)

Initialization

set $T[i] \leftarrow 1$ for $i = 1, \ldots, k$
$x \leftarrow$ generate uniformly at random a value from $(0,1)$
$T[1] \leftarrow x$
broadcast $\langle x \rangle$ to neighbours

Upon receiving a message

receive $\langle x \rangle$
if $\forall_{1 \leq i \leq k} T[i] \neq x$ **then**
  if $\exists_{1 \leq i \leq k} T[i] = 1$ **then**
    $T[i_0] \leftarrow x$ for i_0 such that $T[i_0] = 1$ and sort T
    broadcast $\langle x \rangle$ to neighbours
  else
    if $x < T[k]$ **then**
      $T[k] \leftarrow x$ and sort T
      broadcast $\langle x \rangle$ to neighbours
    end if
  end if
end if

Return the estimated number of sensors

if $\exists_{1 \leq i \leq k} T[i] = 1$ **then**
  return $|\{i : T[i] \neq 1\}|$
else
  return $(k-1)/T[k]$
end if

Algorithm 2 BERNOULLITRIALS(m, p)

Initialization

set bitmap $T[i] \leftarrow 0$ for $i = 0, \ldots, m-1$
for $i \leftarrow 0, \ldots, m-1$ **do**
    set $T[i] \leftarrow 1$ with probability p
end for
set $M \leftarrow \{i : T[i] = 1\}$
broadcast $\langle M \rangle$ to neighbours

Upon receiving a message

receive $\langle M \rangle$
for all $i \in M$ **do**
    if $T[i] = 0$ **then**
        $T[i] \leftarrow 1$
    else
        $M \leftarrow M \setminus \{i\}$
    end if
end for
if $M \neq \emptyset$ **then**
    broadcast $\langle M \rangle$ to neighbours
end if

Return the estimated number of sensors

$\hat{x} = m - \sum_{i=0}^{m-1} T[i]$
if $x = 0$ **then**
    return ∞
else
    return $\log(\hat{x}/m)/\log(1-p)$
end if

We shall use the above estimator for $m = 800$. Note that since one bit represents one trial we need for this purpose 100 bytes.

In the derivation of the two following theorems which determine the bias and the relative error of the estimator $\mathcal{W}_{m,n}$ we apply a very useful result from [7]. For example, this result allows us to obtain a precise asymptotic of the following binomial sum

$$\sum_{k=1}^{n} \log(k) \binom{n}{k} p^k (1-p)^{n-k} = \log(pn) + \frac{p-1}{2pn} + O\left(\frac{1}{n^2}\right).$$

Theorem 2. Let $n \geq 2$, $p \in (0,1)$ and $Q = (1-p)^n$. Then the random variable $\mathcal{W}_{m,n}$ is an asymptotically unbiased estimator of n, namely

$$\mathbf{E}\left[\mathcal{W}_{m,n}\right] = n\left(1 + \frac{\varphi}{m} + O\left(\frac{1}{m^2}\right)\right) \ ,$$

where

$$\varphi = \frac{Q-1}{2Q\log Q} \ .$$

Proof: Notice that $\log(1-p) = \frac{\log Q}{n}$ and

$$\mathbf{E}\left[\mathcal{W}_{m,n}\right] = \frac{n}{\log Q}\left(\mathbf{E}\left[\log(Y_{m,n})\right] - \log m\right) \ .$$

Hence, we need to find the asymptotic of the sum

$$\mathbf{E}\left[\log(Y_{m,n})\right] = \sum_{k=1}^{m} \log k \Pr[Y_{n,m} = k] \ .$$

Since $Y_{n,m}$ follows the binomial distribution, we can use the method presented in [7] to obtain

$$\mathbf{E}\left[\log(Y_{m,n})\right] = \log(Qm) + \frac{Q-1}{2Qm} + O\left(\frac{1}{m^2}\right).$$

□

Theorem 3. Let the conditions for n, p and Q be the same as in Theorem 2. Then

$$\frac{\sigma(\mathcal{W}_{m,n})}{n} = \frac{\psi}{\sqrt{m}} + O\left(\frac{1}{m}\right).$$

where

$$\psi = \sqrt{\frac{1-Q}{Q\log^2 Q}} \ .$$

Proof: To obtain the standard deviation $\sigma(\mathcal{W}_{m,n})$ we note that

$$\mathbf{Var}\left[\mathcal{W}_{m,n}\right] = \frac{n^2}{\log^2 Q}\mathbf{Var}\left[\log(Y_{m,n})\right] \ .$$

Then, by method presented in [7] we get

$$\mathbf{Var}\left[\log(Y_{m,n})\right] = \frac{1-Q}{Qm} + O\left(\frac{1}{m^2}\right) \ .$$

A simple substituting and normalization finishes the proof. □

Remark 2. Note that ψ in Theorem 3 depends on p and n. Assume $p = c_n/n$ and let c_n be a number that minimizes the value of ψ for a given n. It can be shown that

$$c_n \underset{n\to\infty}{\longrightarrow} 2 + W\left(-2/e^2\right) \approx 1.59 \ ,$$

where $W(x)$ denotes the Lambert function. In the following, we use $c = 1.59$ (see Algorithm 3) .

Remark 3. For $n < k$ Algorithm 1 gives precise answer. If $n \geq k$ it returns an estimation $\mathcal{Z}_{k,n} = n\alpha$. For $k = 20$, $n \geq 20$ and $0.5 < \alpha < 2$ (see Remark 1) by setting $p = c/\mathcal{Z}_{k,n}$ we have $\varphi \leq 4.47$ and $\psi \leq 1.61$.

III. THE ALGORITHM'S PRECISION

Putting two phases together we get Algorithm 3 (let us recall that we set $c = 1.59$).

Algorithm 3 TWOPHASEALGORITHM(k, m)

compute $\widehat{n}_1$ using Algorithm 1 with parameter k
if $\widehat{n}_1 < k$ **then**
    return $\widehat{n}_1$
else
    set $p = c/\widehat{n}_1$
    compute $\widehat{n}_2$ using Algorithm 2 with parameters m, p
    return $\widehat{n}_2$
end if

The precision of Algorithm 3 can be determined by the theorem presented in this section. In the proof we shall use the following well known Chernoff bounds for binomial distribution (see e.g. [10]):

Lemma 1. Let $B_{p,n}$ denote a random variable with binomial distribution with parameters p and n, i.e., $\Pr[B_{p,n} = k] = \binom{n}{k}p^k(1-p)^{n-k}$. Suppose that $\delta > 0$ and $0 < \rho \leq 1$. Then

$$\Pr[B_{p,n} \geq (1+\delta)np] \leq \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{np}$$

and

$$\Pr[B_{p,n} \leq (1-\rho)np] \leq \left(\frac{e^{-\rho}}{(1-\rho)^{(1-\rho)}}\right)^{np} .$$

Let S_μ denote the event that in Algorithm 3 we obtain the cardinality estimation with a relative error not greater than $\mu \in (0,1)$. We will formulate now a technical theorem from which we will derive practical corollaries.

Theorem 4. Suppose that conditions of Theorem 1 hold, $\delta_1 > c/n$ and $m \geq 1$. Let $Q_n(\alpha) = (1-\frac{c}{\alpha n})^n, \quad Q(\alpha) = e^{-c/\alpha}$, $g(x) = e^{x-1}x^{-x}$ and

$$G^{\mu}_{k,m}(\alpha) = g(Q(\alpha)^{-\mu})^{mQ_k(\alpha)} + g(Q(\alpha)^{\mu})^{mQ_k(\alpha)} .$$

Then

$$\Pr[\neg S_\mu] \leq F_k(\delta_1,\delta_2) + \max_{\alpha\in(\delta_1,\delta_2)} G^{\mu}_{k,m}(\alpha) .$$

Proof: Let F_{δ_1,δ_2} denote the event that in the first phase we obtain an estimation $\mathcal{Z}_{k,n}$ such that

$$\delta_1 n \leq \mathcal{Z}_{k,n} \leq \delta_2 n .$$

By the law of total probability and simple upper-bounds we get

$$\Pr[\neg S_\mu] \leq \Pr[\neg S_\mu | F_{\delta_1,\delta_2}] + \Pr[\neg F_{\delta_1,\delta_2}] .$$

The probability of the event $\neg F_{\delta_1,\delta_2}$ can be upper-bounded as shown in Theorem 1. Observe that if F_{δ_1,δ_2} occurs then in the Algorithm 3 we get $p = \frac{c}{\alpha n}$ for some α satisfying $\delta_1 \leq \alpha \leq \delta_2$. Consequently, the probability of success in one trial of the second phase is $Q_n(\alpha)$ and

$$\begin{aligned}&\Pr[\neg S_\mu | F_{\delta_1,\delta_2}] = \\ &\Pr[\mathcal{W}_{m,n} \leq (1-\mu)n \ \vee \ \mathcal{W}_{m,n} \geq (1+\mu)n \mid F_{\delta_1,\delta_2}] = \\ &\qquad \Pr[Y_{m,n} \geq mQ_n(\alpha)Q_n(\alpha)^{-\mu} \mid F_{\delta_1,\delta_2}] + \\ &\qquad\qquad \Pr[Y_{m,n} \leq mQ_n(\alpha)Q_n(\alpha)^{\mu} \mid F_{\delta_1,\delta_2}] .\end{aligned}$$

Since random variable $Y_{m,n}$ follows the binomial distribution we can apply the Chernoff bounds

$$\Pr[Y_{m,n} \geq mQ_n(\alpha)Q_n(\alpha)^{-\mu}] \leq g(Q_n(\alpha)^{-\mu})^{mQ_n(\alpha)} ,$$

$$\Pr[Y_{m,n} \leq mQ_n(\alpha)Q_n(\alpha)^{\mu}]) \leq g(Q_n(\alpha)^{\mu})^{mQ_n(\alpha)} .$$

Observe that the conditions of Lemma 1 are satisfied as $Q_n(\alpha)^{-\mu} > 1$, $0 \leq Q_n(\alpha)^{\mu} < 1$ and $\mathbf{E}[Y_{m,n}] = mQ_n(\alpha)$. Note that $Q_n(\alpha) \leq Q(\alpha)$ for $\alpha > c/n$. Using facts that function $g(x)$ is increasing for $0 < x \leq 1$, decreasing for $x \geq 1$ and maximizing over $\alpha \in (\delta_1,\delta_2)$ proves the theorem. □

Let us now consider the case of memory size restricted to 100 bytes i.e. $k = 20$ and $m = 800$. Let us set an accuracy goal to 20% which in most of practical applications should be a sufficient precision.

Corollary 2. By setting $\delta_1 = 0.5062$ and $\delta_2 = 2.1199$ from Theorem 4 we get that $\Pr[S_{0.2}] \geq 0.954$.

In fact, we are able to show that within 100 bytes of available memory we have $\Pr[S_{0.2}] \geq 0.984$ and $\Pr[S_{0.25}] \geq 0.997$. The idea is to cleverly divide the interval (δ_1,δ_2) into disjoint subintervals. Then we can examine each subinterval separately by the technique similar to the one presented in the proof of Theorem 4.

Corollary 3. For 1 kB of available memory and $\delta_1 = 0.55$, $\delta_2 = 1.5$ Theorem 4 gives $\Pr[S_{0.1}] \geq 1 - 10^{-6}$.

IV. Communication Complexity

Let n denote the network size and let k, m be parameters used in the first and in the second phase respectively. Note that the energy consumed by transmissions is usually much higher than energy consumed by listening. Hence, we measure the communication complexity as the number of broadcasts B_n that a sensor carries out during the algorithm's run-time.

The number of broadcasts in the first phase is equal to the number of changes C_n in the table T (see Algorithm 1). Let $I_1, I_2, \ldots I_n$ be random variables such that $I_i = 1$ if there is a change when the ith number occurs and $I_i = 0$ otherwise. Observe that $\Pr[I_i = 1] = 1$ for $i = 1, \ldots k$. Let $X_{k:i}$ be the kth order statistic obtained from a sequence of $i \geq k$ independent random variables uniformly distributed in $(0,1)$. Recall that the density of $X_{k:i}$ is given by the formula [11]

$$f_{k:i}(t) = \mathrm{B}(k, i-k+1)^{-1}t^{k-1}(1-t)^{i-k} ,$$

where $\mathrm{B}(a,b)$ is the Beta function. Notice that for a random variable X uniformly distributed in $(0,1)$ and $i = k+1, \ldots n$ we have

$$\begin{aligned}&\mathbf{E}[I_i] = \Pr[X < X_{k:i-1}] = \\ &\qquad \int_0^1 \Pr[X < X_{k:i-1} | X_{k:i-1} = t] f_{k:i-1}(t)dt = \frac{k}{i} .\end{aligned}$$

Hence $\mathbf{E}[C_n] = k + k\sum_{i=k+1}^{n}\frac{1}{i} = k(1 + \mathrm{H}_n - \mathrm{H}_k)$ where H_n denotes the nth harmonic number. Let us recall that $\mathrm{H}_n = \ln n + \mathrm{O}(1) = \mathrm{O}(\ln n)$.

It is easy to see, that in the second phase (Algorithm 2) the number of broadcasts is upper bounded by m. Finally we obtain that the expected communication complexity of our algorithm is $\mathbf{E}[B_n] = O(\ln n)$. Note that in the algorithm that counts all distinct identifiers each sensor has to inevitably broadcast each identifier. Thus, for such algorithm $B_n = n$.

V. Time Complexity

Let D denote the diameter of the graph of the network. One can easily check that phase (1) as well as phase (2) requires D rounds. Hence, the whole algorithm requires $2D$ rounds for the proper estimation of the network size. Note, that it may happen that our algorithm will return ∞. However the probability of such event is very low, namely for 100 bytes

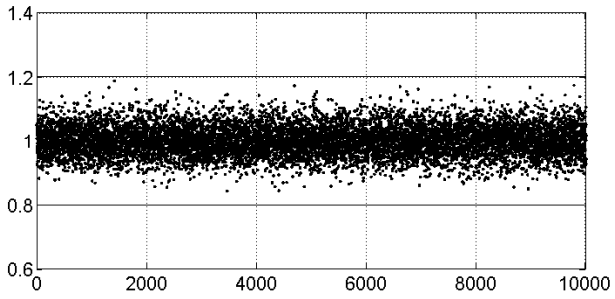

Fig. 1. Plot of $\hat{n}/n$ for $n = 1, \ldots, 10^4$: results given by Algorithm 3 for k=20 and m=800

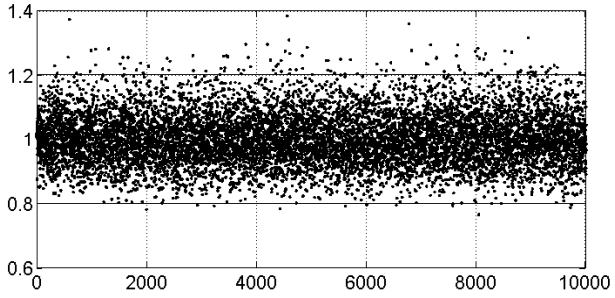

Fig. 2. Plot of $\hat{n}/n$ for $n = 1, \ldots, 10^4$: results given by HyperLogLog with 160 streams (5 bits per stream)

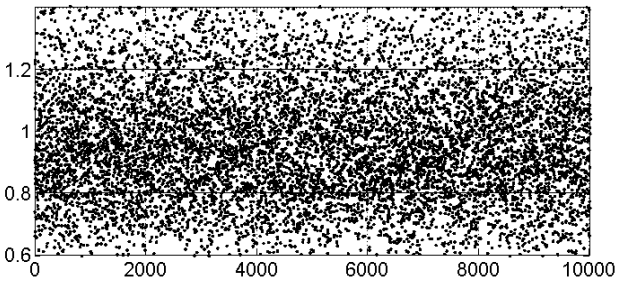

Fig. 3. Plot of $\hat{n}/n$ for $n = 1, \ldots, 10^4$: results given by Extrema Propagation with 20 floating-point numbers (5 bytes per number)

of available memory it is less than 10^{-11}. In such a case the algorithm can be simply re-executed.

VI. Experimental Results

In this section, we present simulation of Algorithm 3 with parameters $k = 20$ and $m = 800$. Note, that in such a setting our algorithm will use at most 100 bytes of memory. As a matter of fact, that was our storage limitation in the practical problem we considered.

In Fig. 1 we show a typical result of simulations. For each $n \in \{1, \ldots, 10^4\}$ we made a simulation of a network with n nodes. The dots represent values of $\frac{\hat{n}}{n}$, where $\hat{n}$ is the estimation returned by the algorithm. One can see that except for a few points we have $0.85 < \frac{\hat{n}}{n} < 1.15$. Hence, for most experiments we have $0.85 \cdot n < \hat{n} < 1.15 \cdot n$. This fact corresponds to the result from Section III, that our algorithm estimates the total number of nodes in the network with relative error of order 0.2 with high probability.

We compared our algorithm with two probabilistic counting techniques. As far as we know, HyperLogLog, is currently one of the most memory-efficient size approximation algorithms. Originally it was dedicated to estimating the number of distinct elements in massive data sets, however the adaptation to WSN requirements is quite straightforward and does not interfere with estimator properties, except that for small cardinalities the balls and bins model is used (see [3]). Extrema Propagation was introduced in the context of counting and data aggregation in distributed systems. In Fig. 1, Fig. 2 and Fig. 3 we compare the precision of Algorithm 3, HyperLogLog and Extrema Propagation within 100 bytes of available memory.

Note that by executing algorithms several times and averaging the results one can improve the estimation accuracy. However, based on a standard deviation of estimators, we can expect that in order to achieve at least the same accuracy as Algorithm 3, HyperLogLog and Extrema Propagation should be executed 3 and 18 times, respectively. According to our knowledge the proposed algorithm is the most accurate of all currently known algorithms estimating the number of nodes using only 100 bytes of memory.

VII. Conclusions

In this paper we presented a two-phase algorithm that efficiently solves the problem of estimating the size of a connected multi-hop network. We examined the estimators used in each phase as well as the precision of the whole algorithm. The precision is connected to the memory consumption and can be freely adjusted. The algorithm is fully distributed and the estimation is produced at each node. The algorithm's run-time is controlled by the diameter of the network.

References

[1] P. Flajolet and G. N. Martin, "Probabilistic counting algorithms for data base applications," *J. Comput. Syst. Sci.*, vol. 31, no. 2, pp. 182–209, 1985.

[2] K.-Y. Whang, B. T. V. Zanden, and H. M. Taylor, "A linear-time probabilistic counting algorithm for database applications," *ACM Trans. Database Syst.*, vol. 15, no. 2, pp. 208–229, 1990.

[3] P. Flajolet, E. Fusy, O. Gandouet, and F. Meunier, "Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm," in *Conference on Analysis of Algorithms, AofA 2007*, 2007.

[4] J. Lumbroso, "An optimal cardinality estimation algorithm based on order statistics and its full analysis," in *AofA'10*, ser. Discrete Mathematics and Theoretical Computer Science, no. 5333, 2010, pp. 491–506.

[5] J. Cichon, J. Lemiesz, and M. Zawada, "On cardinality estimation protocols for wireless sensor networks," in *ADHOC-NOW*, ser. Lecture Notes in Computer Science, vol. 6288. Springer, 2011.

[6] C. Baquero, P. S. Almeida, and R. Menezes, "Fast estimation of aggregates in unstructured networks," in *Proceedings of the 2009 Fifth International Conference on Autonomic and Autonomous Systems*, 2009, pp. 88–93.

[7] P. Flajolet, "Singularity analysis and asymptotics of Bernoulli sums," *Theoretical Computer Science*, vol. 215, no. 1-2, pp. 371–381, 1999.

[8] P. Jacquet, W. Szpankowski, and L. N, "Entropy computations via analytic depoissonization," *IEEE Trans. Information Theory*, vol. 45, pp. 1072–1081, 1998.

[9] D. Mosk-Aoyama and D. Shah, "Computing separable functions via gossip," in *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, ser. PODC '06, 2006, pp. 113–122.

[10] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. New York, NY, USA: Cambridge University Press, 2005.

[11] B. Arnold, N. Balakrishnan, and H. Nagaraja, *A First Course in Order Statistics*. New York: John Wiley & Sons, 1992.