Average Counting via Approximate Histograms

JACEK CICHOŃ and KAROL GOTFRYD, Faculty of Fundamental Problems of Technology, Wrocław University of Science and Technology, Poland

We propose a new algorithm for the classical averaging problem for distributed Wireless Sensors Networks. This subject is well studied and there are many clever algorithms in the literature. These algorithms are based on the idea of local exchange of information. They behave well in dense networks (for example in networks which connections form a complete graph), but their convergence to the real average is very slow in linear or cyclic graphs.

Our solution is different: in order to calculate the average, we first build an approximate histogram of observed data, and then from this histogram we estimate the average. In our solution we use the extreme propagation technique and probabilistic counters. It allows to find the approximation of the average of a set of measurements done by sensor network with arbitrary precision, controlled by two parameters. Our method requires O(D) rounds, where D is the diameter of the network. We study the message complexity of this algorithm and we show that it is of order O(log n) for each node, where n is the size of the network.

CCS Concepts: • **Networks** → **Mobile ad hoc networks**; • **Computer systems organization** → *Sensor networks*; • **General and reference** → *Performance*;

Additional Key Words and Phrases: average estimation, data aggregation, Delta method, Erlang distribution, exponential distribution, extrema propagation, message propagation, probabilistic counters

ACM Reference Format:
Jacek Cichoń and Karol Gotfryd. 2018. Average Counting via Approximate Histograms. *ACM Trans. Sensor Netw.* 14, 9, Article 1 (March 2018), 33 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

The problem of averaging in distributed Wireless Sensors Networks (WSN) has been widely studied in a series of papers (see e.g. [Boyd et al. 2006; Dimakis et al. 2008]). In [Aysal et al. 2009], at each clock tick one randomly chosen sensor broadcasts its information to all its neighbors, then each neighbor averages its own value with the received one. However, with such an algorithm the network's global sum is not preserved so this kind of algorithms does not converge to the true average. This drawback was eliminated in [Kempe et al. 2003] and later in [Iutzeler et al. 2012]. But the convergence speed to the average value of all these algorithms in networks with topology similar to line graph is slow and highly exceeds the diameter of the network, which is the obvious

This paper is supported by the National Science Centre Poland (NCN) under Grant No. 2013/09/B/ST6/02258 and by the Polish Ministry of Science and Higher Education under Research Project No. 0402/0206/16. A preliminary report with part of results from this paper was presented at the conference IEEE Seventh International Conference on Intelligent Systems, Modelling & Simulation (ISMS) 2016.
Authors' address: Jacek Cichoń, jacek.cichon@pwr.edu.pl; Karol Gotfryd, karol.gotfryd@pwr.edu.pl, Department of Computer Science, Faculty of Fundamental Problems of Technology, Wrocław University of Science and Technology, Wybrzeże Wyspiańskiego 27, 50-370, Wrocław, Poland.

1550-4859/2018/3-ART1 $15.00
https://doi.org/0000001.0000001

lower bound on the number of rounds needed to compute the exact average. We shall discuss these phenomena in Section 1.1.

There have been a lot of papers which study the problem of estimation of the size of the network and which estimate the sum $\sum_{v\in V} X_v$ of observed numeric values X_v by nodes $v \in V$. The direct use of any pair of such algorithms for estimation of the average value $\frac{1}{|V|}\sum_{v\in V} X_v$ does not give satisfactory results, since both nominator and denominator are biased and numerical experiments show that the cumulative error is relatively large.

In this paper we propose a novel method of estimation of average in the distributed environment, which can be summarized as follows: build an approximate histogram of observed data using probabilistic counters and finally calculate the average from this histogram. The proposed algorithm is particularly suited for large-scale sensor networks consisting of thousands of devices, where the distances between two nodes (counted in number of hops) may be very large.

Notice that a trivial algorithm for computing average consists in gossiping all pairs (id_v, T_v) containing nodes' identifier id_v and its measurements T_v. It requires uniqueness of nodes identifiers (which can be eliminated by some random mechanism). But more importantly, it has a large communication complexity and requires a large amount of memory for each node to store all data. In large networks storing the actual data gathered from other stations is not possible because of memory limitations. Also the lack of knowledge about network topology makes the task of data aggregation even more difficult. Thus one needs to find a tradeoff between the amount of information processed and stored by every node and the number of messages sent by all stations.

It is worth mentioning that in practice the knowledge of exact average of sensors' measurements is not necessary in most applications and approximate ones are also acceptable (a detailed analysis of this issue can be found in [Li et al. 2017]). A similar approach is used in [He et al. 2015] for the identification of other important aggregates such as quantiles and range countings in wireless sensor networks.

Let us note that we do not assume e.g. that the devices have some global identifiers known in advance (in [Cardoso et al. 2009] a similar setting is considered, where stations have some random unique identifiers, which are previously not known to other stations). The knowledge of nodes about the network topology is also very limited – initially each station knows only its direct neighbors and has even no information about the network size (each parameter required for running the algorithm is determined during the precomputation phase).

Another important feature of our algorithm is that after its execution the same estimate of average is produced at every node. Some of the existing efficient protocols result in obtaining the outcomes of data aggregation in some predetermined node by gathering all data from other stations. These algorithms, in order to achieve the same effect, would require an additional communication phase for spreading the information about calculated estimates. We also do not want to allow for such situations, where one station coordinates the whole process, expending much more energy than other devices. As it is important to have all devices still being capable of working for possibly long time, we focused also on preventing any particular station from running out of battery much faster than others.

Our method is based on the extreme propagation technique popularized by C. Baquero, P. S. Almeida, and R. Menezes in 2009 in [Baquero et al. 2009] (see also [Baquero et al. 2012; Blaskiewicz et al. 2012]) and on the notion of probabilistic counters. A discussion on some practical aspects of communication cost of extrema propagation technique based algorithms in real distributed settings can be found e.g. in [Baquero et al. 2012; Cardoso et al. 2009]

Probabilistic counters were invented in 1977 by Robert Morris in [Morris 1978]. The idea of using this kind of counting techniques for estimation of aggregates in networks was introduced in

[Considine et al. 2004; Nath et al. 2004]. The method of using collections of probabilistic counters is not new. It was used for example to control the use of flash memory blocks in [Cichoń and Macyna 2011].

Our algorithm runs in $O(D)$ steps, where D is the diameter of network. Its precision is controlled by two parameters. Let us recall that we are not assuming any knowledge on the size of the network. We assume only that we know some reasonable upper bound on the network diameter. The problem of finding lower and upper bounds on the diameter of a network is well studied (see e.g. [Ajwani et al. 2012; Roditty and Williams 2013]). Let us remark that it is folklore that a single BFS run rooted at an arbitrary source yields trivial upper bounds on the diameter with factor 2, which is sufficient for our purposes. Let us also remark that Stephan Holzer et al. [Holzer et al. 2014] presented an algorithm that computes a $\frac{3}{2}$-approximation of the diameter w.h.p. in $O(\sqrt{n \log n} + D)$ time (where n denotes the number of nodes).

It is worth to remark that the exact calculation of the network diameter in the settings where the number of devices in the network is not known appears to be time-consuming and the existing protocols have large time complexity (see e.g. [Peleg et al. 2012] and [Frischknecht et al. 2012] where the authors consider the model of the network that is similar to that studied in our paper). However, we do not need to know the exact value of the network diameter. It is sufficient for our protocol to have only some crude approximation.

One of the crucial factors determining the efficiency of protocols designed for wireless sensor networks is their energy complexity. As the energy is a critical resource in such distributed environments, we put a lot of attention on the energetic aspects of designed algorithm. We based our averaging method on a variant of extrema propagation technique, which appears to be a robust framework for building more complex energy efficient protocols (cf. the discussion in [Baquero et al. 2012; Cardoso et al. 2009]). As the dominating factor for energy expenditure during the execution of the proposed algorithm is the number and the size of messages sent by each station, we measure its energy efficiency in terms of message complexity, defined for a given node as the number of transmissions it performs. In Section 4.1 we present a thorough discussion on the message complexity of our solution, showing in Lemma 4.5 that the expected number of messages sent by each station is of order $O(\ln n)$. Hence, the overall message complexity (i.e. the total number of transmissions performed by all nodes in the network) is $O(n \ln n)$.

As we shall see, there is a kind of tradeoff between the communication cost and time and memory complexity of the averaging protocols in settings where the resources of nodes in very large networks are significantly limited. Some additional payoff in number of transmissions (but still only within a constant factor) allows for designing more efficient and robust solutions. It is also worth noting, that the size of messages sent is directly related to the precision of obtained estimates. If in a given application only some crude approximation is required, then the number of probabilistic counters can be reduced, thus resulting in smaller size of messages.

This paper is an extension of a preliminary report [Cichoń and Gotfryd 2016] presented at the conference ISMS 2016. This paper contains a more detailed discussion on related work, all proofs necessary for formal analysis of our algorithm, a section with experimental analysis of two previously known algorithms, a comprehensive evaluation of precision of the algorithm, analysis of its message complexity and some other new remarks.

1.1 Previous algorithms

In this section we show that there exist families of graphs for which the time complexity of averaging methods relying on gossip-based information exchange is of order $\Theta(n^2)$. This will serve as a motivation for posing the following question which arises when considering the worst-case

behavior of the abovementioned algorithms. Namely, if there exists a protocol for distributed average estimation having worst-case time complexity not exceeding the network size or diameter. Starting with these motivating examples we will then answer this question affirmatively, showing that with a small payoff in energy expenditure (in terms of message complexity) we are able to guarantee worst-case running time of our averaging algorithm of order $\Theta(D)$ (and hence $O(n)$) regardless of the network topology.

We shall present results of numerical experiments in which we investigated the number of steps needed to achieve the desired accuracy by these algorithms in line graphs. More precisely, for several values of n we organized $2n$ nodes in a line graph; we put initially values 1 into the first n nodes and values 0 into last n nodes; clearly the average of initial values is 0.5; we ran this algorithm until the maximal error is less than 0.1 and we recorded the number of rounds. We repeated this experiment 50 times for each n and we calculated the mean from it.

Before we skip into the summary of our experiments with these algorithms let us emphasize that the discussed procedures converge quickly to the average and are very efficient in terms of energy complexity in case of dense graphs (for example in complete graphs or Random Geometric Graphs). Line graphs are the worst case for them.

1.1.1 Push-Sum Algorithm. The Push-Sum algorithm from [Kempe et al. 2003] uses two variables (s, w). In the initial phase this pair is initialized at node u to $(x_u, 1)$ where x_u is the value observed by u, and then u sends this pair to itself. At the beginning of each round each node u gathers all pairs $(s_i, w_i)_{i=1,\ldots,k}$ sent to it and calculates $s = \sum_{i=1}^{k} s_i$, $w = \sum_{i=1}^{k} w_i$. Next, at the end of each round, it chooses uniformly at random one of its neighbors v and sends the pair $\frac{1}{2}(s, w)$ to v and to itself. The number s_u/w_u is the estimate of the average value of $(x_v)_{v\in V}$, where V is the set of all nodes.

It is easy to check that after each round the average of all values $(s_v)_{v\in V}$ is always the correct average and that $\sum_{v\in V} w_v = |V|$. But the proof of correctness of this algorithm (convergence to actual average of values s_u/w_u for each node u) is non-trivial. It uses the "potential" function $\Phi = \sum_{u,v\in V} \left(c_{u,v} - \frac{w_u}{|V|}\right)^2$, where $c_{u,v}$ are so called *contributions* at node u, related to s_u and w_u by the formulas $s_u = \sum_{v\in V} c_{u,v} \cdot x_v$ and $w_u = \sum_{v\in V} c_{u,v}$. Authors show that the function Φ decreases at least twice after each round and later they use classical results about the diffusion speed in graphs to show the rate of convergence.

The numerical experiments show that the expected number of rounds necessary to reach precision 0.1 (for each node) is close to an^2, where $a \approx 0.27$ (see Fig. 1). Therefore, in this situation (line graph) $O(n^2)$ steps are required to be sure that each node has an estimation of the average value 0.5 from the interval $[0.4, 0.6]$.

1.1.2 BWGossip Algorithm. The averaging algorithm from [Iutzeler et al. 2012] (BWGossip Algorithm) is fully distributed. Each node has two variables s and w, therefore the total state of this algorithm is described by two tables $(s_v)_{v\in V}$ and $(w_v)_{v\in V}$ and the state of a node v is described by a pair (s_v, w_v). Initially all s_v are set to be the values observed by the nodes (which we want to average) and $w_v = 1$.

At each step one random node $v \in V$ wakes up and sends to all its neighbors the vector $m_v = (\frac{s_v}{n_v+1}, \frac{w_v}{n_v+1})$, where n_v is the cardinality of its neighborhood $\mathcal{N}_v$. The node v updates its state to m_v and each node $u \in \mathcal{N}_v$ updates its state to $(s_u, w_u) = (s_u, w_u) + m_v$ (according to element-wise operations). The number s_{v^*}/w_{v^*} is then the estimate of the average value of $(x_v)_{v\in V}$ after this step at the node $v^* \in V$.

Figure 2 shows the average number of rounds (transmissions) of BWGossip algorithm required to reach the precision of order 10% with respect to the number n of nodes in the network. This

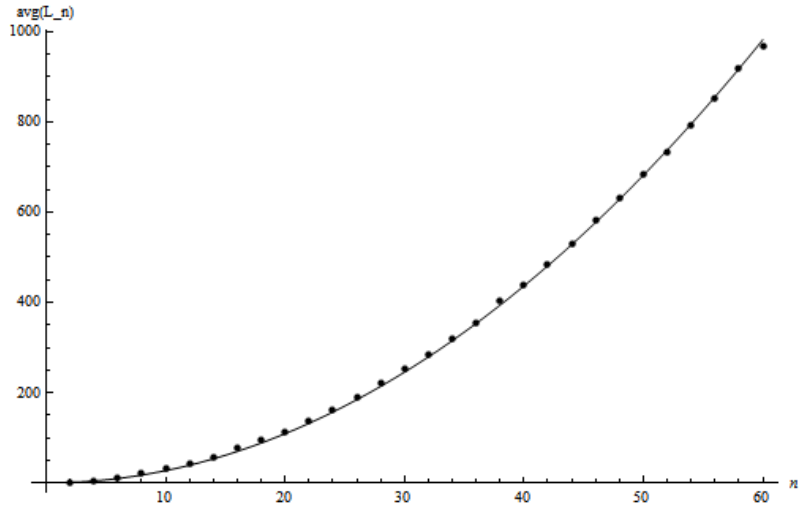

Fig. 1. Line graph with n nodes and the average number of rounds in Push-Sum algorithm needed to reach precision 0.1 starting from the vector $(1, \ldots, 1, 0, \ldots, 0)$. We made 1000 numerical experiments for each even n between 2 and 60.

figure contains experimental data (points) and its approximation by the function $L = 0.55 \cdot n^3$. We see that for obtaining the desired accuracy the algorithm requires on average the number of rounds which is of order n^3.

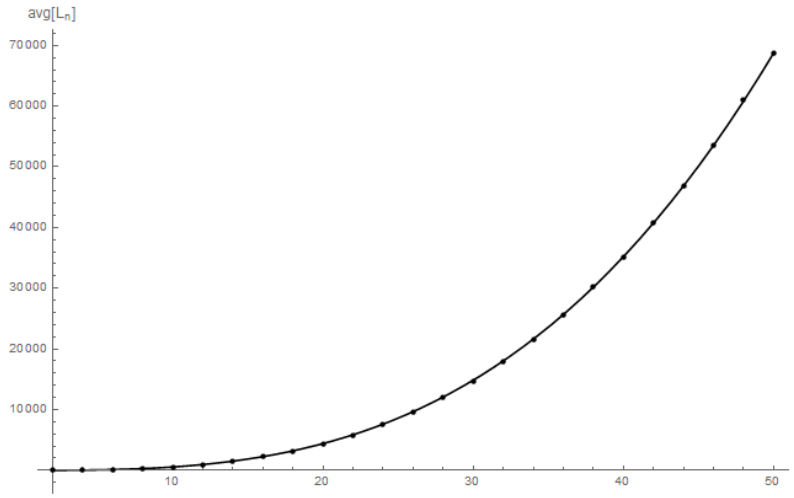

Fig. 2. Average number of transmissions required by BWGossip Algorithm to reach the accuracy of order 10% with respect to the network size (line graph, first half of nodes has value 1, the other half – 0).

Let us notice that the algorithm considered in this section is asynchronous, so at each round only one (randomly chosen) node is activated. In order to compare this algorithm with the previous one and with our method we slightly modify the original procedure from [Iutzeler et al. 2012]. Namely

we divide the time into rounds and in each round we wake up all nodes once but in a random order. This time (see Figure 3) we see that the expected number of rounds is close to $0.4 \cdot n^2$, hence is of order $O\left(n^2\right)$.

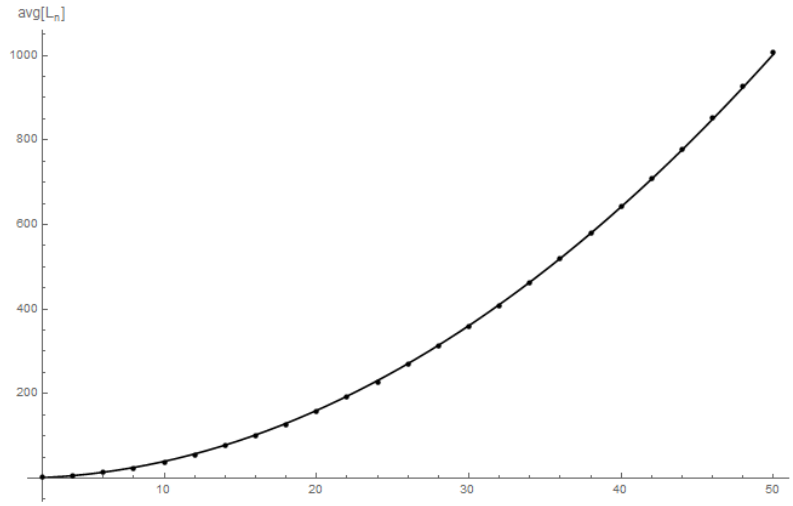

Fig. 3. Average number of rounds required by modified BWGossip Algorithm to reach the accuracy of order 10% with respect to the network size (line graph, first half of nodes has value 1, the other half – 0).

Slightly different and more general approach to the problem of distributed averaging in wireless ad hoc networks was presented in [Boyd et al. 2006]. However, the authors explicitly assume that the underlying graph is not a bipartite graph, so we may not test their solutions on line graphs. But simulations performed on non-bipartite graphs similar to line graphs show analogous behavior of algorithms based on this approach: the expected number of rounds required to obtain a given precision is of order $O\left(n^2\right)$.

1.1.3 Related papers. An important problem closely related to average approximation is the estimation of the distribution of some parameter's values detected by the nodes in a distributed environment. A lot of algorithms for that problem have been proposed in recent years (see e.g. [Borges et al. 2012; Haridasan and van Renesse 2008; Sacha et al. 2010] and the references therein). This class of protocols aims in determining some practical approximation of the actual distribution of some global characteristic of the network. This is clearly much more than simply estimating individual aggregates, as such estimates can usually be trivially obtained from the calculated distribution. In [Haridasan and van Renesse 2008], the authors introduced algorithms based on gossiping, which allow for computing an estimation of the distribution of nodes' measurements by each peer in the network. They considered a couple of techniques in order to decrease the size of messages transmitted between nodes and reduce the overall message complexity of the protocol. Among others, they considered two methods based on dynamically building histograms of values received so far from other nodes.

In the first one histograms with equal intervals are constructed, whereas the second one relies on histograms with unequal bins, such that each bin contains approximately the same number of points (*equi-depth histograms*). The latter approach, though leading to more precise results, requires however some additional work for dynamic splitting and merging histograms' intervals as new values are arriving. A series of experimental results were presented, showing that the

proposed protocols are able to efficiently (in terms of time and communication complexity) produce quite accurate estimations of the parameter's distribution for various classes of distributions. The quality of obtained approximations was evaluated with Kolmogorov-Smirnoff distance between two probability measures (maximum over all nodes was taken into account for analyzing the worst-case errors). Especially the approach where so called *equi-depth histograms* were used resulted in estimations with relatively high precision.

Another class of algorithms for estimation of cumulative density function (cdf) was presented in [Sacha et al. 2010]. It also employs gossip-based dissemination of information between nodes and provides a mechanism for self-assessment of the accuracy of calculated estimations. In general, their protocols consist of multiple consecutive instances, where in every such instance the set of thresholds is selected (based on the knowledge acquired so far) in order to construct some data structure effectively equivalent to cumulative histogram of values gathered from other nodes. The estimations of values of the actual cdf in the selected points are then combined into an approximate cdf using linear interpolation. From the approximation calculated in a given instance, the set of thresholds is then refined based on some heuristics in order to produce more accurate distribution estimations in successive steps. The algorithm was also extended with a technique allowing for evaluation of the approximation's accuracy. The experiments presented in the paper show the robustness and efficiency of the protocol for both smooth and skewed distributions.

However, protocols from [Haridasan and van Renesse 2008; Sacha et al. 2010] assume different communication model from that considered by us. The efficiency and robustness of these methods rely on the fact that sets of nodes' neighbors may vary in time, which in particular requires that any two nodes can communicate directly. In [Sacha et al. 2010] the authors assume, among other things, that stations are allowed to randomly select their neighbors and change them over time for assuring better connectivity properties. In contrast, the algorithm introduced by us is designed for multi-hop settings, where the network topology does not change. It is also worth mentioning that the analysis of these algorithms is restricted only to experimental evaluation based on simulations. Let us also note that these algorithms do not result in a consensus – each peer has its own approximation of the values' distribution and hence the obtained aggregates may vary across the nodes.

The problem of reaching a consensus on the aggregates like average of the measurements performed by the nodes has also been extensively studied in the literature in recent years. As the problem of obtaining precise estimations of parameters of interest with the additional requirement that each station should have the same value at the end of the protocol is not a trivial task, some relaxations have been considered. An example are the algorithms for *quantized consensus* discussed in [Kashyap et al. 2007], where a discrete version of the averaging problem in sensor networks was studied. The authors in [Kashyap et al. 2007] proposed solutions based on classical pairwise gossiping protocol, where the sensors quantize the values of their measurements and at the end of protocol's execution each station obtains some integer approximation $a\hat{v}g \in \{\mu, \mu + 1\}$ of the actual average avg such that $avg \in [\mu, \mu + 1]$. They also provide a comprehensive theoretical analysis of conditions under which the nodes converge to quantized consensus and the rate of that convergence. These algorithms, however, do not lead to true consensus, as each sensor can end up with one of two possible values for average approximation.

In [Benezit et al. 2009] a similar problem named *interval consensus*, which is some generalization of well known voting problem in distributed systems, was studied. The authors considered a network, where each node has a quantized value of some measurement and there is a given set of intervals known in advance to the stations. The goal is to determine in fully distributed manner which interval contains the average of their measurement. They proposed another modifications of pairwise gossip by appropriately adapting the updating rules to allow for reaching the consensus on the set of intervals and provide some theoretical and experimental analysis of their algorithms.

Both of the aforementioned techniques of quantized consensus, however, are relying on gossip-based information exchange. Thus, although they are also suited for the task of average estimation in the distributed systems, the performance issues pointed out in Section 1.1 also apply to them.

Besides the algebraic operations like sum and average, there are other important types of aggregate operations which can provide valuable characteristics of the large scale wireless sensor networks. An example are the holistic aggregations, extensively studied in [He et al. 2015] and [Li et al. 2017]. The authors propose energy efficient sampling-based algorithms for approximating with an arbitrary precision the quantiles, frequencies, ranks and the number of distinct elements in distributed datasets consisting of values gathered by the sensors. The discussion involves also the detailed, formal and experimental analysis of the precision of constructed estimators and the communication complexity of designed protocols. Moreover, in [He et al. 2015] an algorithm for approximate counting of the number of observations in a given range based on dynamic binary tree is discussed. However, all these protocols are designed for some kind of centralized environment in the sense that they require the existence of a predetermined station (called sink or coordinator) which gathers all the partial aggregation results from other nodes and calculates the final estimation, which is then disseminated across the whole network.

A lot of attention has also been recently given to constructing reliable and fault-tolerant data aggregation algorithms in wireless settings in the presence of message loss, links and nodes failures or in the scenarios where nodes can dynamically leave or join the network. The examples of novel, efficient protocols which have been proven to be resilient to such communication failures can be found, among others, in [Almeida et al. 2017; Gansterer et al. 2013; Jesus et al. 2015].

In the design and analysis of performance of the algorithms for distributed data aggregation one of the key factors is the time they require for spreading the data across the network and for convergence of the nodes' estimations to their final values. There is a vast body of literature dealing with the relations of convergence speed of the protocols based on randomized rumor spreading or gossiping (cf. [Boyd et al. 2006; Karp et al. 2000]) to the characteristics of graphs describing the underlying networks like conductance or vertex expansion. For the detailed discussion on that subjects see e.g. [Chierichetti et al. 2010; Giakkoupis and Sauerwald 2012] and references therein.

1.2 Mathematical Notation and Background

We denote by $|A|$ the cardinality of a set A. By $\Gamma(x)$ we denote the standard generalization of the factorial function. Notice that $\Gamma(n) = (n-1)!$ for any integer $n \geq 1$.

We denote by $\mathbf{E}[X]$ and $\mathbf{var}[X]$ the expected value and the variance of the random variable X, respectively. We denote by $\xrightarrow{d}$ the convergence in distribution of random variables.

Let us recall that a random variable X has the exponential distribution with parameter $\lambda > 0$ ($X \sim \mathrm{Exp}(\lambda)$) if its density function f_X is given by the formula $f_X(x) = \lambda \exp(-\lambda x)$. It is well known that if $X_1, \ldots, X_n \sim \mathrm{Exp}(\lambda)$ are independent and $Y = \min\{X_1, \ldots, X_n\}$, then $Y \sim \mathrm{Exp}(n\lambda)$. We denote by $\mathcal{N}(0, s^2)$ the normal distribution with mean 0 and variance s^2.

If $X_1, \ldots, X_L$ are independent random variables with a common $\mathrm{Exp}(\mu)$ distribution, then the sum $S = X_1 + \ldots + X_L$ has the Erlang distribution with parameters L and μ ($S \sim \mathrm{Erl}(L, \mu)$), i.e. its density function is given by the formula

$$f_{L,\mu}(x) = \frac{\mu^L x^{L-1} e^{-\mu x}}{(L-1)!} \,. \tag{1}$$

All of the above probabilistic concepts and distributions of random variables can be found in the book [Billingsley 2012].

For the sake of clarity, the main symbols used in this paper together with its brief descriptions are summarized in Table 1.

Table 1. Major symbols.

Symbol	Definition
n	the size of the network
$\vec{T} = (T_i)_{i=1,\ldots,n}$	the vector of the observable data stored by the sensors
$\mathrm{avg}(\vec{T})$	the arithmetic mean of the elements of vector $\vec{T}$ of real numbers
m, M	minimal and maximal value in $\vec{T}$
K	the number of intervals (bins) used for construction of a histogram
$\mathcal{I} = (I_1, \ldots, I_K)$	the partition of the interval $[m, M]$ into sub-intervals
$m(I)$	the middle point of the interval I
$\vec{H} = (H_i)_{i=1,\ldots,K}$	the histogram of the actual sensors' data $\vec{T}$
$\mathrm{wm}_{\mathcal{I}}(\vec{T})$	the weighted mean of the elements of $\vec{T}$ by partition $\mathcal{I}$
L	the number of probabilistic counters used for cardinality estimation
$\vec{C}_L = (C_{L,i})_{i=1,\ldots,K}$	the approximate histogram of the data $\vec{T}$
$\mathrm{err}(\mathrm{wm}(\vec{H}), \mathrm{wm}(\vec{C}_L))$	the approximation error of the average
$\mathrm{MC}_{G;x}^{K,L}$	the message complexity of the algorithm for node x in the network G
$\mathbf{E}[X]$	the expected value of a random variable X
$\mathbf{var}[X]$	the variance of a random variable X
$\mathrm{Exp}(\lambda)$	the exponential distribution with parameter λ
$\mathrm{Erl}(L, \lambda)$	the Erlang distribution with parameters L and λ
$\mathcal{N}(\mu, s^2)$	the normal distribution with mean μ and variance s^2
Γ	the Gamma function (an extension of the factorial function)

1.3 Organization of the Paper

In Section 2 we describe the idea of approximate histograms of data collected by nodes in a network. We show that an estimator of the average value based on this notion is asymptotically unbiased. In Section 3 a measure of precision of approximate histograms is introduced and the worst case of data is recognized. We present a pseudocode of proposed algorithm for average estimation, discuss its basic properties and show a detailed analysis of its energy efficiency defined in terms of message complexity in Section 4. Section 5 presents the results of a series of numerical experiments which illustrate the precision of estimates obtained using our procedure applied to various kinds of data. An example of utilizing additional information gathered during execution of the averaging algorithm for estimation of the network size is demonstrated in Section 6. Finally, we conclude in Section 7 summarizing fundamental properties of proposed algorithm.

2 HISTOGRAMS

We assume that the network is modeled by a connected graph with diameter D. The edges of this graph correspond to bidirectional communication links. Suppose that the network consists of n nodes numbered by $\{1, \ldots, n\}$ and that each node stores a value T_k. Let $\vec{T} = (T_i)_{i=1,\ldots,n}$. Our goal is to estimate the mean

$$\mathrm{avg}(\vec{T}) = \frac{1}{n} \sum_{i=1}^{n} T_i$$

in an efficient and easy way.

Using the extreme propagation technique in its basic form we may assume that each node knows the values $m = \min\{T_i : i = 1, \ldots, n\}$ and $M = \max\{T_i : i = 1, \ldots, n\}$. If $m = M$ then the average

value of the sequence (T_i) is known. Suppose hence that $m < M$ and let $\Delta = M - m$. We fix a parameter K and we split the interval $[m, M]$ into K disjoint intervals of equal length: we put $I_i = [m + \frac{\Delta}{K}(i-1), m + \frac{\Delta}{K} i)$ for $i = 1, \ldots K-1$ and $I_K = [m + \frac{\Delta}{K}(K-1), M]$, so

$$[m, M] = I_1 \cup I_2 \cup \ldots \cup I_K \,.$$

Let $\mathcal{I} = (I_1, \ldots, I_K)$. We denote by $m(I_i)$ the middle point of the interval I_i, i.e. we put $m(I_i) = m + \frac{\Delta}{K}(i - \frac{1}{2})$ and for an arbitrary vector $\vec{x} = (x_i)_{i=1,\ldots,K}$ of non-negative numbers such that $\sum_{i=1}^{K} x_i > 0$ we define the *weighted mean* of $\vec{x}$ by partition $\mathcal{I}$ as

$$\mathrm{wm}_{\mathcal{I}}(\vec{x}) = \frac{\sum\limits_{i=1}^{K} m(I_i) x_i}{\sum\limits_{i=1}^{K} x_i} \,.$$

We will drop the index $\mathcal{I}$ when this parameter is clear from the context.

Let $H_i = |\{k : T_k \in I_i\}|$ for $i \in \{1, \ldots, K\}$. We call the vector $\vec{H} = (H_i)_{i=1,\ldots,K}$ the *histogram* of the data $(T_i)_{i=1,\ldots,n}$. We are going to approximate the average value of observed data $\vec{T}$ by an *approximation* of the number

$$\mathrm{wm}(\vec{H}) = \frac{\sum\limits_{i=1}^{K} m(I_i) \cdot H_i}{\sum\limits_{i=1}^{K} H_i} \,. \tag{2}$$

In this approach each observed value is approximated by the nearest element from the set of middle points $(m(I_i))_{i=1,\ldots,K}$, so some error in this method is unavoidable. We call this error a *discretization error*. It is controlled by the number K of sub-intervals into which we divide the range of observed data and by the spread of observed data, as stated in Theorem 2.1.

Theorem 2.1 (Discretization error). *For an arbitrary vector $\vec{T}$ of observed data we have*

$$\frac{\left|\mathrm{wm}(\vec{H}) - \mathrm{avg}(\vec{T})\right|}{M - m} \le \frac{1}{2K} \,,$$

where $m = \min\{T_i : i = 1, \ldots, n\}$ and $M = \max\{T_i : i = 1, \ldots, n\}$.

Proof. Let $\Delta = M - m$. Notice that $H_1 + \ldots + H_K = n$. Therefore

$$|\mathrm{wm}(\vec{H}) - \mathrm{avg}(\vec{T})| = \left|\frac{1}{n}\sum_{i=1}^{n} T_i - \frac{\sum_{j=1}^{K} m(I_j) H_j}{\sum_{j=1}^{K} H_j}\right| = \frac{1}{n}\left|\sum_{j=1}^{K}\sum_{T_i \in I_j} T_i - \sum_{j=1}^{K} m(I_j) H_j\right| =$$

$$\frac{1}{n}\left|\sum_{j=1}^{K}\left(\sum_{T_i \in I_j} T_i - m(I_j) H_j\right)\right| = \frac{1}{n}\left|\sum_{j=1}^{K}\sum_{T_i \in I_j}\big(T_i - m(I_j)\big)\right| \le \frac{1}{n}\sum_{j=1}^{K}\sum_{T_i \in I_j}\big|T_i - m(I_j)\big| \le$$

$$\frac{1}{n}\sum_{j=1}^{K}\sum_{T_i \in I_j}\frac{\Delta}{2K} = \frac{1}{n}\frac{\Delta}{2K}\sum_{j=1}^{K}\sum_{T_i \in I_j} 1 = \frac{\Delta}{2Kn}\sum_{j=1}^{K} H_j = \frac{M-m}{2K} \,,$$

so the theorem is proved. □

Remark 2.2. We do not discuss in this paper the problem of proper setting the parameter K, i.e. the number of bins in histogram. There is extensive literature on this subject, see for example [Scott 1979] or [Chen et al. 2008].

2.1 Approximate Counters

Probabilistic counters have been intensively investigated in last years. They were invented in 1977 by Robert Morris (see [Morris 1978]). This original version was carefully analyzed in the early 1980s by Philippe Flajolet (cf. [Flajolet 1985]), who coined the name Approximate Counting. A popular realization of this kind of counters is Hyperloglog (see [Flajolet et al. 2007]), which is a popular tool in Big Data applications. In more recent investigations some other methods were proposed for estimation of the cardinality of a stream of data. Some of them are well suited for counting the size of distributed networks (see e.g. [Cichoń et al. 2011]).

One class of such counters is based on the following property of exponential distribution: if $X_1, \ldots, X_n$ are independent random variables with the common distribution Exp(1), then the random variable $Y = \min\{X_1, \ldots, X_n\}$ has the distribution Exp(n) (see Section 1.2).

In fact, one random variable $Y \sim \mathrm{Exp}(n)$ is not sufficient for estimation of the parameter n. However, if we have a sequence $Y_1, \ldots, Y_L$ of independent random variables with Exp(n) distribution where $L > 2$, then the random variable $Z = Y_1 + \ldots + Y_L$ has the Erlang distribution Erl(L, n). From Eq. 1 we can easily deduce that $\mathbf{E}\left[\frac{L-1}{Z}\right] = n$ and $\mathbf{var}\left[\frac{L-1}{Z}\right] = \frac{n^2}{L-2}$. Therefore, the random variable $C = \frac{L-1}{Z}$ is an unbiased estimator of the number n and its precision is controlled by the parameter L. We will use this approach in this paper.

Remark 2.3. If all nodes $v \in V$ from a considered connected network generate numbers $(X_v)_{v \in V}$, then the number $\min\{X_v : v \in V\}$ can be easily computed by the ordinary extrema propagation technique in $\mathrm{O}(D)$ rounds (cf. [Baquero et al. 2009]). We will use this observation in Section 4.

Remark 2.4. Another popular approach to construction of approximate counters is based on order statistics (see e.g. [Cichoń et al. 2012b]). They have an interesting properties: namely, this kind of counters gives precise results for small number of observations. However, we found that the theoretical analysis of solutions based on such counters is more difficult than that one based on Erlang distribution in the context of the present paper.

2.2 Approximate Histograms

Let $\vec{T} = (T_i)_{i=1,\ldots,n}$ be the sequence of observed values. We split the interval $[\min(\vec{T}), \max(\vec{T})]$ into K intervals $(I_i)_{i=1,\ldots,K}$ of equal lengths and we associate with each interval I_i an approximate counter $C_{L,i}$ counting the number $H_i = |\{k : T_k \in I_i\}|$ based on the Erlang distribution Erl(L, H_i). The method of transformation of numbers $(H_i)_{i=1,\ldots,K}$ into the numbers $(C_{L,i})_{i=1,\ldots,K}$ will be described in Section 4.

We call the vector $\vec{C}_L = (C_{L,i})_{i=1,\ldots,K}$ *an approximate histogram* of the data $\vec{T}$. Let $\vec{H}$ be the histogram obtained from $\vec{T}$. We will show in Theorem 2.6 that the number $\mathrm{wm}(\vec{C}_L)$ is an asymptotically unbiased estimator of the number $\mathrm{wm}(\vec{H})$ when $L \longrightarrow \infty$. Before we prove this convergence result we will need the following auxiliary lemma.

Lemma 2.5. *Suppose that $X_L \sim \mathrm{Erl}(L, m)$, where $L > 2$. Let $Y_L = \frac{L-1}{X}$. Then $\mathbf{E}[Y_L] = m$, $\mathbf{var}[Y_L] = \frac{m^2}{L-2}$ and the sequence $\sqrt{L}(Y_L - m)$ converges in distribution to the normal distribution $\mathcal{N}(0, m^2)$.*

Proof. Let f_{X_L} and f_{Y_L} denote the probability density functions for the random variables X_L and Y_L, respectively. Notice that $\Pr[\frac{L-1}{X} < t] = \Pr[X > \frac{L-1}{t}] = 1 - \Pr[X \leq \frac{L-1}{t}]$. Therefore $f_Y(t) = -f_X(\frac{L-1}{t})\frac{d}{dt}(\frac{L-1}{t})$. After some simplifications we obtain the following formula

$$f_{Y_L}(t) = \frac{e^{-\frac{(L-1)m}{t}}\left(\frac{(L-1)m}{t}\right)^L}{t \cdot \Gamma(L)}.$$

From this formula we easily deduce that $\mathbf{E}[Y_L] = m$ and $\mathbf{var}[Y_L] = \frac{m^2}{L-2}$. Observe that

$$f_{Y_{L+1}}(t) = \frac{me^{-\frac{Lm}{t}}\left(\frac{Lm}{t}\right)^L}{t^2 \cdot \Gamma(L)} .$$

It is worth to remark that this means that the random variable Y_{L+1} follows InvGamma$(L+1, Lm)$ distribution.

Let $Z_L = \frac{\sqrt{L}}{m}(Y_{L+1} - m)$. Then $\Pr[Z_L < t] = \Pr[Y_{L+1} < m + \frac{mt}{\sqrt{L}}]$. From this we deduce that the probability density function f_{Z_L} of the random variable Z_L is given by the formula

$$f_{Z_L}(t) = \frac{L^{\frac{3L}{2}+\frac{1}{2}}}{\Gamma(L)} e^{-\frac{L^{3/2}}{\sqrt{L}+t}} \left(\sqrt{L}+t\right)^{-L-2} .$$

Let us transform the first part of this formula:

$$\frac{L^{\frac{3L}{2}+\frac{1}{2}}}{\Gamma(L)} = \frac{L^{\frac{3L}{2}+\frac{1}{2}}}{(L-1)!} = \frac{L^L\sqrt{L}}{e^L L!} \cdot e^L \cdot L^{\frac{1}{2}L} \cdot L$$

and observe that $\lim_{L\to\infty} \frac{L^L\sqrt{L}}{e^L L!} = \frac{1}{\sqrt{2\pi}}$.

After some simple transformations we get

$$\ln\left(e^L \cdot L^{\frac{1}{2}L} \cdot L \cdot e^{-\frac{L^{3/2}}{\sqrt{L}+t}} \left(\sqrt{L}+t\right)^{-L-2}\right) = -\frac{1}{2}t^2 + \mathrm{O}\left(\frac{1}{\sqrt{L}}\right) ,$$

so for each fixed t we have $\lim_{L\to\infty} f_{Z_L}(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2}$. The classical Scheffé's lemma implies that the sequence (Z_L) converges in distribution to the normal distribution $\mathcal{N}(0,1)$.

Therefore the random variable $\frac{\sqrt{L}}{m}(Y_L - m)$ converges in distribution to the standard normal random variable $\mathcal{N}(0,1)$, or, equivalently, the random variable $\sqrt{L}(Y_L - m)$ converges in distribution to $\mathcal{N}(0, m^2)$ if L grows to infinity. □

Theorem 2.6. *Let $\vec{H} \in \mathbb{R}^K$ be a vector of non-negative numbers such that $C = \sum_{i=1}^K H_i > 0$. Then*

$$\sqrt{L}\big(\mathrm{wm}(\vec{C}_L) - \mathrm{wm}(\vec{H})\big) \xrightarrow{d} \mathcal{N}(0, s^2) ,$$

when $L \to \infty$, where $s^2 = \frac{1}{C^4} \sum_{i=1}^K \left(\sum_{j=1}^K (m(I_j) - m(I_i))H_iH_j\right)^2$.

Proof. Let us fix i such that $H_i \geq 1$. Then $C_{L,i} = \frac{L-1}{X}$ where $X \sim \mathrm{Erl}(L, H_i)$. From Lemma 2.5 we deduce that the sequence $\sqrt{L}(C_{L,i} - H_i)$ converges (if L grows to infinity) in distribution to the normal distribution $\mathcal{N}(0, H_i^2)$. Notice that if $H_i = 0$, then $C_{L+1,i} = 0$, so $\sqrt{L}(C_{L,i} - H_i) = 0$, hence also in this case we have the convergence to $\mathcal{N}(0,0)$ interpreted as the Dirac delta function. Observe also that random variables $C_{L,1}, \ldots, C_{L,K}$ are independent. Therefore

$$\sqrt{L}(C_{L,1} - H_1, \ldots, C_{L,K} - H_K) \xrightarrow{d} \mathcal{N}(0, \Sigma) ,$$

where $\Sigma = \mathrm{diag}(H_1^2, \ldots, H_K^2)$ is the square diagonal matrix with elements $(H_1^2, \ldots, H_K^2)$ on the main diagonal.

We are going to apply the Multivariate Delta Method (see e.g. [Small 2010]) to the function wm(). Notice that

$$\frac{d}{dx_i}\mathrm{wm}() = \frac{d}{dx_i}\frac{\sum_{j=1}^K m(I_j)x_j}{\sum_{j=1}^K x_j} = \frac{\sum_{j=1}^K (m(I_i) - m(I_j))x_j}{(\sum_{j=1}^K x_j)^2} .$$

Let $\nabla_{\vec{H}}$ be the gradient $(\frac{d}{dx_1}\mathrm{wm}(), \ldots, \frac{d}{dx_K}\mathrm{wm}())$ evaluated at the point $\vec{H} = (H_1, \ldots, H_K)$. From the Multivariate Delta Method we get

$$\sqrt{L}(\mathrm{wm}(\vec{C}_L) - \mathrm{wm}(\vec{H})) \xrightarrow{d} \mathcal{N}(0, \nabla^T_{\vec{H}} \Sigma \nabla_{\vec{H}}) \,,$$

hence

$$\sqrt{L}(\mathrm{wm}(\vec{C}_L) - \mathrm{wm}(\vec{H})) \xrightarrow{d} \mathcal{N}(0, s^2) \,,$$

where

$$s^2 = \sum_{i=1}^{K} \left(\frac{\sum_{j=1}^{K} (m(I_j) - m(I_i)) H_j}{(\sum_{j=1}^{K} H_j)^2} \right)^2 H_i^2 = \frac{\sum_{i=1}^{K} \left(\sum_{j=1}^{K} (m(I_j) - m(I_i)) H_i H_j \right)^2}{\left(\sum_{i=1}^{K} H_i \right)^4} \,.$$

Hence the theorem is proved. □

From the Theorem 2.6 the following corollaries follow.

Corollary 2.7. $\mathbf{E}\left[\mathrm{wm}(\vec{C}_L)\right] \approx \mathrm{wm}(\vec{H})$

Corollary 2.8. *Let $C = \sum_{i=1}^{K} H_i$. If $C > 0$ then*

$$\mathbf{var}\left[\mathrm{wm}(\vec{C}_L)\right] \approx \frac{1}{LC^4} \cdot \sum_{i=1}^{K} \left(\sum_{j=1}^{K} (m(I_j) - m(I_i)) H_i H_j \right)^2 \,.$$

3 PRECISION OF APPROXIMATE HISTOGRAMS

Our goal is to compare the number $\mathrm{wm}(\vec{H})$ (see Formula 2) with $\mathrm{wm}(\vec{C}_L)$. In Section 5 we shall discuss the outcomes of a series of performed experiments. For a proper interpretation of obtained results we will use the following measure of error of the estimate $\mathrm{wm}(\vec{C}_L)$

$$\mathrm{err}(\mathrm{wm}(\vec{H}), \mathrm{wm}(\vec{C}_L)) = \frac{|\mathrm{wm}(\vec{H}) - \mathrm{wm}(\vec{C}_L)|}{M - m} \,.$$

Notice that $0 \leq \mathrm{err}(\mathrm{wm}(\vec{H}), \mathrm{wm}(\vec{C}_L)) \leq 1$.

Let us observe that this notion of error is independent on linear transformations of $\vec{T}$. Namely, if $a > 0$ and $T'_i = aT_i + b$ for $i = 1, \ldots, n$, then $H'_i = H_i$. Therefore, approximate counters count the same cardinalities. and this means that this notion of error is independent on such changes of scale. For example if we change the units of measurement from Fahrenheit to Celsius or to Kelvins degrees, then the error of estimate will remain the same. The following Theorem 3.1 states this property more precisely.

Theorem 3.1. *Let $\mathcal{I} = (I_1, \ldots, I_K)$ be a partition of interval $[m, M]$ into intervals of equal lengths. Let $\alpha, \beta \in \mathbb{R}$ and $\alpha > 0$. Let $\mathcal{J} = (\alpha I_i + \beta)_{i=1,\ldots,K}$. Let $\vec{x}, \vec{y} \in \mathbb{R}^K$ be nonnegative vectors such that $\sum_{i=1}^{K} x_i > 0$ and $\sum_{i=1}^{K} y_i > 0$. Then*

$$\mathrm{err}(\mathrm{wm}_{\mathcal{I}}(\vec{x}), \mathrm{wm}_{\mathcal{I}}(\vec{y})) = \mathrm{err}(\mathrm{wm}_{\mathcal{J}}(\vec{x}), \mathrm{wm}_{\mathcal{J}}(\vec{y})) \,.$$

Proof. Let $J = \bigcup\{J_i : i = 1, \dots, K\}$. Then $J = \alpha[m, M] + \beta = [\alpha m + \beta, \alpha M + \beta]$. For an arbitrary $\vec{z} \in \mathbb{R}^K$ such that $\sum_{i=1}^K z_i > 0$ we have

$$\mathrm{wm}_{\mathcal{J}}(\vec{z}) = \frac{\sum_{i=1}^K m(J_i) z_i}{\sum_{i=1}^K z_i} = \frac{\sum_{i=1}^K (\alpha m(I_i) + \beta) z_i}{\sum_{i=1}^K z_i} =$$
$$\frac{\alpha \sum_{i=1}^K m(I_i) z_i + \beta \sum_{i=1}^K z_i}{\sum_{i=1}^K z_i} = \alpha \frac{\sum_{i=1}^K m(I_i) z_i}{\sum_{i=1}^K z_i} + \beta = \alpha \cdot \mathrm{wm}_{\mathcal{I}}(\vec{z}) + \beta \,.$$

Therefore

$$\mathrm{err}(\mathrm{wm}_{\mathcal{J}}(\vec{x}), \mathrm{wm}_{\mathcal{J}}(\vec{y})) = \frac{|\mathrm{wm}_{\mathcal{J}}(\vec{x}) - \mathrm{wm}_{\mathcal{J}}(\vec{y})|}{\alpha(M - m)} =$$
$$\frac{\alpha |\mathrm{wm}_{\mathcal{I}}(\vec{x}) - \mathrm{wm}_{\mathcal{I}}(\vec{y})|}{\alpha(v_K - v_1)} = \mathrm{err}(\mathrm{wm}_{\mathcal{I}}(\vec{x}), \mathrm{wm}_{\mathcal{I}}(\vec{y})) \,,$$

so the Theorem is proved. □

From Theorem 3.1 we deduce that investigating the errors of proposed estimator of average value based on probabilistic counters may be reduced to such data where the middle points $(m(I_i))_{i=1,\dots,K}$ are fixed and equal to $(1, 2, \dots, K)$. In this case we have

$$\mathrm{wm}_{\mathcal{I}}(\vec{x}) = \frac{\sum\limits_{i=1}^{K} i \cdot x_i}{\sum\limits_{i=1}^{K} x_i}$$

and (see Corollary 2.8) $\mathbf{var}\left[\mathrm{err}(\mathrm{wm}(\vec{H}), \mathrm{wm}(\vec{C}_L))\right] \approx h(H_1, \dots, H_K)$ where

$$h(x_1, \dots, x_K) = \frac{1}{L \cdot (K+1)^2} \frac{\left(\sum\limits_{j=1}^{K} \left(\sum\limits_{i=1}^{K} (j-i) x_i x_j\right)\right)^2}{\left(\sum\limits_{i=1}^{K} x_i\right)^4} \,.$$

Theorem 3.2 (formulated and proved below) implies that when $\sum_{i=1}^K H_i = C$ is fixed, then the function h attains its maximum value at the point $\vec{c} = (\frac{C}{2}, 0, \dots, 0, \frac{C}{2})$ and we have $h(\vec{c}) = \frac{1}{8L} \frac{(K-1)^2}{(K+1)^2}$. This case of highly concentrated data at two extreme values will be carefully discussed in Section 5, where we present the results of performed experiments.

In the case when the initial data stored by the stations are distributed uniformly among the bins, i.e. when $H_i = a$ for each $i = 1, \dots, K$, we have $h(a, a, \dots, a) = \frac{1}{12L} \frac{K^2-1}{K(K+1)^2} \leq \frac{1}{12 \cdot L \cdot K}$.

For $c > 0$ and $k \geq 2$ we put

$$\Sigma_{c,k} = \{\vec{x} \in \mathbb{R}^k : \sum_{i=1}^{k} x_i = c \wedge \bigwedge_{i=1}^{k} (x_i \geq 0)\} \,.$$

Notice that $\Sigma_{c,k}$ is a compact subset of $\mathbb{R}^k$.

Theorem 3.2. *Let $c > 0$, $k \geq 2$ and*

$$f(x_1, \dots, x_k) = \sum_{j=1}^{k} x_j^2 \left(\sum_{i=1}^{k} (j-i) x_i\right)^2 \,.$$

If $\vec{b} = (\frac{c}{2}, 0, \dots, 0, \frac{c}{2}) \in \Sigma_{c,k}$ then $f(\vec{b}) = \sup\{f(\vec{x}) : \vec{x} \in \Sigma_{c,k}\}$ and $f(\vec{b}) = \frac{(k-1)^2 c^4}{8}$.

The proof of this theorem is based on the following Lemma 3.3.

LEMMA 3.3. *Suppose that* $\vec{x} = (x_1, \ldots, x_k) \in \Sigma_{c,k}$, $1 < l < k$ *and* $x_l > 0$. *Let*

$$\vec{x'} = \left(x_1 + \frac{k-l}{k-1} x_l, x_2, \ldots, x_{l-1}, 0, x_{l+1}, \ldots, x_{k-1}, x_k + \frac{l-1}{k-1} x_l \right) .$$

Then $f(\vec{x}) \leq f(\vec{x'})$.

PROOF. Let $I_j(y_1, \ldots, y_k) = \sum_{i=1}^{k} (j-i) y_i$. Then

$$I_j(\vec{x'}) - I_j(\vec{x}) = (j-1)\frac{k-l}{k-1} x_l - (j-l) x_l + (j-k)\frac{l-1}{k-1} x_l =$$
$$\frac{x_l}{k-1}((j-1)(k-l) - (j-l)(k-1) + (j-k)(l-1)) = 0 ,$$

therefore $I_j(\vec{x'}) = I_j(\vec{x})$ for each $j = 1, \ldots, k$. Moreover, from the identity $(l-i)(k-1) = (k-l)(1-i) + (l-1)(k-i)$ we get

$$I_l(\vec{x}) = \frac{k-l}{k-1} I_1(\vec{x}) + \frac{l-1}{k-1} I_k(\vec{x}) .$$

From this observation and from the fact that $I_1(\vec{x}) \leq 0$ and $I_k(\vec{x}) \geq 0$ we deduce that

$$I_l(\vec{x})^2 \leq \left(\frac{k-l}{k-1} \right)^2 I_1(\vec{x})^2 + \left(\frac{l-1}{k-1} \right)^2 I_k(\vec{x})^2 .$$

Hence

$$f(\vec{x'}) - f(\vec{x}) = \sum_{j=1}^{k} {x'_j}^2 I_j(\vec{x'})^2 - \sum_{j=1}^{k} {x_j}^2 I_j(\vec{x})^2 =$$
$$\sum_{j=1}^{k} ({x'_j}^2 - x_j^2) I_j(\vec{x})^2 = ({x'_1}^2 - x_1^2) I_1(\vec{x})^2 - x_l^2 I_l(\vec{x})^2 + ({x'_k}^2 - x_k^2) I_k(\vec{x})^2 \geq$$
$$x_l^2 \left(\left(\frac{k-l}{k-1} \right)^2 I_1(\vec{x})^2 - I_l(\vec{x})^2 + \left(\frac{l-1}{k-1} \right)^2 I_k(\vec{x})^2 \right) \geq 0 ,$$

so the lemma is proved. □

PROOF OF THEOREM 3.2. The set $\Sigma_{c,k}$ is a compact subset of $\mathbb{R}^k$ and f is a continuous function on $\Sigma_{c,k}$. Therefore there exists a point $\vec{b} \in \Sigma_{c,k}$ such that

$$f(\vec{b}) = \sup\{ f(\vec{x}) : \vec{x} \in \Sigma_{c,k} \} .$$

From Lemma 3.3 we deduce that the maximal value of the function f on the set $\Sigma_{c,n}$ is attached on the subset $\{(a, 0, \ldots, 0, c-a) : 0 \leq a \leq c\}$. Let us observe that

$$f(\alpha, 0, \ldots, 0, c-\alpha) = 2(k-1)^2 \alpha^2 (c-\alpha)^2 .$$

Thus, the function $g(\alpha) = f(\alpha, 0, \ldots, 0, c-\alpha)$ reaches its maximum on the interval $[0, c]$ at point $\alpha = \frac{c}{2}$ and $g(\frac{c}{2}) = \frac{(k-1)^2 c^4}{8}$. Hence the theorem is proved. □

4 ALGORITHM

In this section we describe a pseudocode (Algorithm 1) of the proposed in this paper algorithm HistMean for distributed averaging. It is used by every node in the network. We assume that the communication in the network is divided into rounds and that in each round each pair of connected nodes can exchange information in both directions. Let us also recall, that the graph modeling the underlying network is undirected and the message complexity of the algorithm defined with respect to a given node is the number of transmission events that occur in this node. As we shall discuss later in this section, each such transmission comprises of sending L values. For practical purposes it suffices to use 5 bytes for each value in order to obtain satisfactory precision for most of the applications.

The input of this algorithm are:

(1) D^*: an upper approximation of the diameter D of a network
(2) m: minimal value of observed data
(3) M: maximal value of observed data
(4) K: the number of sub-intervals dividing the range $[m, M]$
(5) L: the number of exponential random variables connected with each sub-interval

We assume that in the initial phase, before running this algorithm, an another algorithm calculates the values m and M. As mentioned previously, this can be done by utilizing the ordinary deterministic version of extrema propagation procedure (cf. [Baquero et al. 2009; Cichoń et al. 2012a]). We also assume that the number D^* is known. In fact, it is sufficient to know some upper bound on the network diameter, because this algorithm stabilizes (no new messages is sent) after D rounds, where D is the actual network diameter. Both numbers K and L have influence on the precision of obtained estimates of the average of observed data.

The algorithm uses an array $X[1 \dots K][1 \dots L]$ of floating point numbers. Initially this array is filled with values ∞. At the beginning each node uses the observed value T to find the number a such that

$$T \in [m + \frac{M-m}{K}(a-1), m + \frac{M-n}{K}a] \,.$$

Notice that a is the index of sub-interval of the interval $[m, M]$ the observed value T falls into. Then the node fills the subarray $X[a][1 \dots L]$ with independently generated random numbers from exponential distribution with parameter 1.

Let X_v denotes the copy of the array X controlled by the node v. The idea of our algorithm is to transform *inside the network* arrays $(X_v)_{v \in V}$ into the array consisting of values

$$X[a][j] = \min\{X_v[a][j] : v \in V\}$$

and use this array to estimate the histogram of data. More precisely, we use the vector $X[a][1 \dots L]$ for estimation of the number of nodes which observed values are in the a-th interval. The calculation of the array X is done using a variant of extrema propagation method. Notice that at the end of the algorithm each node has precisely the same copy of the array X.

The algorithm HistMean uses $K \cdot L$ probabilistic counters. It was shown in [Cichoń et al. 2012a] that for a large class of natural graphs the expected message complexity of the classical extreme propagation procedure for one node is $O(\log D)$, where D is the diameter of the graph. But in our algorithm a message is transmitted by a node if any change in the array of probabilistic counters is observed. It can be easily checked that if $K \cdot L \gg D$ then the expected number of transmissions for one node in the proposed algorithm is close to $D \cdot K$. The detailed analysis of message complexity of the procedure HistMean is presented in section 4.1.

ALGORITHM 1: Distributed averaging algorithm HistMean

```
function HistMean(D*, m, M, K, L)
begin
    /* Initialization                                                    */
    T = observed value
    for a = 1 ... K do
        for j = 1 ... L do
            X[a][j] = +∞
        end
    end
    Δ = M − m
    find a such that T ∈ [m + (Δ/K)(a − 1), m + (Δ/K)a]
    for j = 1 ... L do
        /* generation of a pseudo-random real from Exp(1) distribution   */
        X[a][j] = RandomExp(1)
    end
    send pair (a, X[a]) to all neighbors
    /* broadcasting loop                                                 */
    for I = 1 ... D* do
        C = X
        foreach received (a, Y) do
            for j = 1 ... L do
                C[a][j] = min(C[a][j], Y[j])
            end
        end
        for a = 1 ... K do
            if C[a] ≠ X[a] then
                X[a] = C[a]
                send pair (a, X[a]) to all neighbors
            end
        end
    end
    /* final calculations                                                */
    for a = 1 ... K do
        S[a] = 0
        for j = 1 ... L do
            S[a] = S[a] + X[a][j]
        end
        if S[a] > 0 then
            H[a] = (L − 1)/S[a]
        end
        else
            H[a] = 0
        end
    end
    S1 = Σ_{i=1}^{K} (m + (Δ/K)(i − 1/2))H[i]
    S2 = Σ_{i=1}^{K} H[i]
    return S1/S2
end
```

The memory complexity of this algorithm is $\mathrm{O}(K \cdot L)$. In fact, if we know that $n \leq 10^{10}$ (where n denotes the total number of nodes), then each probabilistic counter can be implemented as 5 bytes float reals without losing precision.

Notice that after execution of this algorithm each node knows not only the approximate value of the average of observed data (the same estimation for all nodes), but also knows the approximate histogram, which may be used to other purposes. As an example, in Section 6 we discuss one of such possible applications, namely how to make use of the outcomes of our algorithm to the problem of network size estimation.

Let us also remark that we can easily recognize from the histogram such situations where the observed values are not concentrated near the mean value. This may happen, for example, when half of the stations observed value 0 and the second half observed value 1. The algorithms described in the Section 1.1 do not give such information.

4.1 Message complexity of the algorithm

In this section we will examine the average message complexity of our averaging algorithm HistMean for any station in the network represented by the graph G, defined as the number of messages sent by this station during the execution of the algorithm. Such definition of message complexity as the number of transmissions can be justified by the fact that the energy usage while sending a message is much higher than the cost of listening.

As already outlined above, the message complexity of the proposed averaging algorithm can be bounded above by $O(D \cdot K)$ for any node in any connected network with diameter D. This is due to the simple observation that in the worst case scenario stations can update all K vectors in all rounds. This in turn leads to K transmissions in each of D rounds of the algorithm. The simple worst case bound gives a good approximation of the average number of transmissions for graphs with small diameter, i.e. when $D = \mathrm{O}(1)$. In that case, if $n \gg K$, then in most rounds stations receive much more than K vectors of Exp(1) random variables and we may expect that on average there will be some changes in pretty large fraction of min-Exp vectors $X[i]$ related to the intervals of histogram. But this approximation appears to be very crude if the underlying graph describing network topology has large diameter, e.g. linear in number of nodes (i.e. $D = \mathrm{O}(n)$). Then, intuitively, the number of vectors received by a node from their neighbors in consecutive rounds will be usually close to K or even less than K. Thus, in many rounds no new minima will be gathered and hence nothing will be transmitted.

This preliminary discussion will be formalized later in this section. As we shall see, assuming the parameters K and L of the algorithm being constant, the average message complexity for any node in an arbitrary connected graph can be upper bounded by some function $f(n) = \mathrm{O}(\ln n)$ (in general $f(n)$ depends on both K and L). This is a significant improvement of the simple bound $O(D \cdot K)$ when network diameter D is large.

For the sake of clarity let us recall that in each round of our algorithm each station first gathers all vectors from their neighbors, and then for every $1 \leq i \leq K$ calculates the pointwise minimum of the stored vector $X[i]$ of current minimal values and received vectors corresponding to i^{th} interval. Finally, if some value of $X[i]$ has changed, then the station sends the updated $X[i]$ to their neighbors. In the analysis below sending a vector of L values is considered as a single transmission.

Before we give the formal statements of the results on message complexity, let us introduce some required notations and basic definitions.

Let $G = (V, E)$ be an arbitrary connected graph with $|V| = n$ and diameter D. For any fixed $x \in V$ and $0 \leq r \leq D$ we denote $S(x, r) = \{v \in V : d(x, v) = r\}$ and $B(x, r) = \{v \in V : d(x, v) \leq r\}$ with $S(x, 0) = B(x, 0) = \{x\}$. Moreover, assuming that our histogram consists of K intervals I_i, $1 \leq i \leq K$,

and the data observed by the nodes are $(T_v)_{v\in V}$, we put $S^{(i)}(x,r) = \{v \in V : d(x,v) = r \wedge T_v \in I_i\}$ and $B^{(i)}(x,r) = \{v \in V : d(x,v) \le r \wedge T_v \in I_i\}$. Observe that there is exactly one $i \in \{1,\ldots,K\}$ such that $S^{(i)}(x,0) = B^{(i)}(x,0) = \{x\}$ and for all $j \neq i$ we have $S^{(j)}(x,0) = B^{(j)}(x,0) = \emptyset$. Finally, by the $\mathrm{MC}^{K,L}_{G;x}$ we denote the random variable representing the message complexity of the algorithm HistMean for graph G and node x when running with parameters K and L.

Let us also recall that the n^{th} harmonic number is defined by $H_n = \sum_{i=1}^{n} \frac{1}{i}$ and the n^{th} harmonic number of order $r > 1$ is given by $H_n^{(r)} = \sum_{i=1}^{n} \frac{1}{i^r}$. It is a well known fact that $H_n = \ln n + \mathrm{O}(1)$ and for $r > 1$ we have $\lim_{n\to\infty} H_n^{(r)} = \zeta(r)$, where ζ is the Riemann Zeta function.

Some of the theorems proved in this section are generalizations of the results from [Cichoń et al. 2012a], hence we will omit part of details referring to the original paper when possible.

The main result of this section is the following Theorem 4.1.

Theorem 4.1. *Let $G = (V,E)$ be an arbitrary connected graph with $|V| = n$ nodes and let $x \in V$. For any fixed constant parameters $L \ge 1$ and $K \ge 1$ the expected message complexity of the algorithm HistMean for x is*

$$\mathbf{E}\left[\mathrm{MC}^{K,L}_{G;x}\right] = \sum_{i=1}^{K} \sum_{\substack{r\ge 0 \\ |B^{(i)}(x,r)|>0}} 1 - \left(1 - \frac{|S^{(i)}(x,r)|}{|B^{(i)}(x,r)|}\right)^{L} = \mathrm{O}(\ln n).$$

We will prove this theorem in the following steps. First, we derive the expression for expected message complexity for any fixed node x of an arbitrary graph G in the case $K = 1$ and $L \ge 1$. We will prove that it is maximized over all connected graphs with n nodes for the line graph L_n and one of the outermost vertices. Thus, we generalize the results obtained in [Cichoń et al. 2012a]. Next we will analyze the case $K > 1$ using the fact that for any fixed partition of n values (stored by the stations) into K bins it suffices to consider separately the number of transmissions related to particular bins and sum them up for obtaining the overall message complexity for a given station. We will also show that for the outermost vertex of L_n for any fixed n and K the most balanced partition of n values into K bins maximizes the expected message complexity. This will lead us to the upper bound, which is of logarithmic order in the network size n for constant K and L. The following lemmas formalize these results.

Lemma 4.2. *Let $G = (V,E)$ be an arbitrary connected graph representing the network with $|V| = n$ nodes and let $x \in V$. Fix some $L \ge 1$. Then*

$$\mathbf{E}\left[\mathrm{MC}^{1,L}_{G;x}\right] = \sum_{r\ge 0}\left(1 - \left(1 - \frac{|S(x,r)|}{|B(x,r)|}\right)^{L}\right)$$

and

$$\mathbf{var}\left[\mathrm{MC}^{1,L}_{G;x}\right] = \sum_{r\ge 1}\left(1 - \left(1 - \frac{|S(x,r)|}{|B(x,r)|}\right)^{L}\right)\left(1 - \frac{|S(x,r)|}{|B(x,r)|}\right)^{L}.$$

Proof. Proof of this lemma is similar to that of Theorem 1 in [Cichoń et al. 2012a], which is a particular case for $L = 1$. Denoting by $M^L_{x;r}$ the event that the station x transmits in round $r \ge 0$, our goal is to estimate the probability $\Pr[\mathbf{1}_{M^L_{x;r}}] = 1$, where $\mathbf{1}_{M^L_{x;r}}$ is an indicator random variable related to the event $M^L_{x;r}$.

Notice that x sends in round r a message with updated vector of current minima if and only if in this round there was some change on at least one of the coordinates of the min-Exp vector stored by x. From the proof of Theorem 1 in [Cichoń et al. 2012a] it follows that for any fixed $1 \le j \le L$ such a change on the j^{th} coordinate occurs with probability $\frac{|S(x,r)|}{|B(x,r)|}$. Because all Exp(1) random

variables the initial vectors consist of were generated independently by each station, the events that j^{th} coordinate changes for $1 \leq j \leq L$ are independent, hence the probability that there was some change leading to the transmission of updated vector is

$$\Pr[M^L_{x;r}] = 1 - \left(1 - \frac{|S(x,r)|}{|B(x,r)|}\right)^L.$$

Using the same kind of argument as in [Cichoń et al. 2012a] one can show that the events $\{M^L_{x;r}\}_{r\geq 0}$ are independent and from the definition of message complexity we obtain $\mathrm{MC}^{1,L}_{G;x} = \sum_{r\geq 0} \mathbf{1}_{M^L_{x;r}}$. Therefore

$$\mathbf{E}\left[\mathrm{MC}^{1,L}_{G;x}\right] = \sum_{r\geq 0}\left(1 - \left(1 - \frac{|S(x,r)|}{|B(x,r)|}\right)^L\right)$$

and

$$\mathbf{var}\left[\mathrm{MC}^{1,L}_{G;x}\right] = \sum_{r\geq 1}\left(1 - \left(1 - \frac{|S(x,r)|}{|B(x,r)|}\right)^L\right)\left(1 - \frac{|S(x,r)|}{|B(x,r)|}\right)^L.$$

□

Lemma 4.3. *For an arbitrary connected graph $G = (V, E)$ with $|V| = n$, any vertex $x \in V$ and any fixed $L \geq 1$*

$$\mathbf{E}\left[\mathrm{MC}^{1,L}_{G;x}\right] \leq \mathbf{E}\left[\mathrm{MC}^{1,L}_{L_n;x^*}\right] = n - \sum_{i=1}^{n}\left(1 - \frac{1}{i}\right)^L$$

where x^* is an outermost vertex of the line graph L_n.

Proof. The proof proceeds in the same vein as that of Theorem 2 in [Cichoń et al. 2012a]. We adopt the same set of simple transformations of the original graph G with the property that applying them to any graph G and vertex x for which there exists $r > 0$ such that $|S(x,r)| > 1$ will result in increasing the expected message complexity for x. Performing some finite number of such transformations on G results eventually in the line graph L_n with x being one of the outermost vertices.

The only difference is that for proving that these transformations increase the message complexity we apply the inequality (for $b \geq 1$, $c \geq 2$)

$$1 - \left(1 - \frac{c}{b+c}\right)^L < 1 - \left(1 - \frac{c-1}{b+c-1}\right)^L + 1 - \left(1 - \frac{1}{b+c}\right)^L,$$

which is equivalent to

$$\frac{(b+c-1)^L - b^L}{(b+c-1)^L} - \frac{(b+c-1)^L - b^L}{(b+c)^L} > 0.$$

Calculating the expected message complexity for the case when x is one of the outermost vertices of L_n completes the proof. □

We have already shown that the theorems on message complexity of the extreme propagation technique from [Cichoń et al. 2012a] can be easily generalized for the case of transmitting vectors of $L \geq 1$ random values. Let us now consider the case $K \geq 1$.

Lemma 4.4. *Let $G = (V, E)$ be an arbitrary connected graph with $|V| = n$ nodes and let $x \in V$. For any fixed $L \geq 1$ and $K > 1$*

$$\mathbf{E}\left[\mathrm{MC}^{K,L}_{G;x}\right] = \sum_{i=1}^{K}\sum_{\substack{r\geq 0\\ |B^{(i)}(x,r)|>0}} 1 - \left(1 - \frac{|S^{(i)}(x,r)|}{|B^{(i)}(x,r)|}\right)^L.$$

Moreover, for any fixed partition of n values into K bins such that n_i stations have initial values belonging to the interval I_i, $1 \le i \le K$, the expected message complexity of x is not greater than for the line graph L_n with x at one of its ends, i.e.

$$\mathbf{E}\left[\mathrm{MC}^{K,L}_{G;x}\right] \le \mathbf{E}\left[\mathrm{MC}^{1,L}_{L_n;x}\right] = n - \sum_{i=1}^{K}\sum_{j=1}^{n_i}\left(1-\frac{1}{j}\right)^L. \tag{3}$$

Proof. From construction of the algorithm HistMean and the fact that all random values generated during the initial phase of our algorithm are independent one can easily see that sending an updated vector $X[i]$ related to i^{th} bin of the histogram for any fixed i can be triggered only by receiving another vector with values generated initially by some other station u such that $T_u \in I_i$. It is clear that any received vector related to some other interval I_j, $j \ne i$, has no influence on occurrence of transmissions of $X[i]$. From this observation we may conclude that for any fixed partition of n values into K bins $(n_1, \ldots, n_K)$ it is sufficient to separately count the number of times the min-Exp vector $X[i]$ is updated (and thus transmitted) for $1 \le i \le K$ and sum up these counts over all bins to obtain the overall message complexity for given graph G and vertex x. The same type of argument as in the case $K = 1$ leads us to the conclusion that for any fixed G and x and any given dataset $\vec{T} = (T_v)_{v\in V}$ the expected message complexity $\mathbf{E}\left[\mathrm{MC}^{K,L}_{G;x}\right]$ is not greater than for L_n with x at one of its ends and the same initial data $\vec{T}$. □

Lemma 4.5. *For any fixed n, K and L the message complexity for the outermost vertex x of line graph L_n is maximized for the most balanced partitions of n values into K bins, i.e. when for any $1 \le i, j \le K$ the numbers of stations n_i and n_j with initial values from intervals I_i and I_j, respectively, fulfill $|n_i - n_j| \le 1$. Moreover, if K and L are constant, then $\mathbf{E}\left[\mathrm{MC}^{K,L}_{L_n;x}\right] = K \cdot L \cdot \ln n + O(1) = \Theta(\ln n)$.*

Proof. Fix n, K and L and let x be the outermost vertex of L_n. From the formula 3 from Lemma 4.4 it follows that for maximizing $\mathbf{E}\left[\mathrm{MC}^{K,L}_{L_n;x}\right]$ it suffices to find such partition $\eta = (n_1, \ldots, n_K)$ of n values into K bins (with additional condition that $n_1 > 0$ and $n_K > 0$) that minimizes the function

$$\varphi(K, L, \eta) = \sum_{i=1}^{K}\sum_{j=1}^{n_i}\left(1-\frac{1}{j}\right)^L.$$

Let us consider a partition $\eta = (n_1, \ldots, n_s, \ldots, n_t, \ldots n_K)$ and suppose that $|n_s - n_t| \ge 2$ for some $1 \le s, t \le K$, $s \ne t$. Without loss of generality we assume $n_s \ge n_t + 2$. Let us define $\eta' = (n'_1, \ldots, n'_K)$ such that $n'_s = n_s - 1$, $n'_t = n_t + 1$ and $n'_i = n_i$ for $i \notin \{s, t\}$. Then we have

$$\begin{aligned}\varphi(K, L, \eta') &= \sum_{i=1}^{K}\sum_{j=1}^{n'_i}\left(1-\frac{1}{j}\right)^L \\ &= \sum_{i=1}^{K}\sum_{j=1}^{n_i}\left(1-\frac{1}{j}\right)^L - \left(1-\frac{1}{n_s}\right)^L + \left(1-\frac{1}{n_t+1}\right)^L < \varphi(K, L, \eta),\end{aligned}$$

where the last inequality follows from the assumption $n_s \ge n_t + 2$. Now we can apply this transformation repeatedly until we obtain one of the partitions such that for any $i \ne j$ we have $|n_i - n_j| \le 1$. This leads us to the conclusion that the most balanced distribution of the stations' initial values maximizes the message complexity $\mathbf{E}\left[\mathrm{MC}^{K,L}_{L_n;x}\right]$ of HistMean algorithm for the outermost vertex of line graph.

To complete the proof of Lemma 4.5 it suffices to show $\mathbf{E}\left[\mathrm{MC}^{K,L}_{L_n;x}\right] = \Theta(\ln n)$. Let us assume then that K and L are some fixed constants and let $m > 0$ be fixed. After some simple algebraic transformations and using the fact that for $l > 1$ we have $\lim_{m\to\infty} H^{(l)}_m = \zeta(l)$ we get

$$m - \sum_{j=1}^{m}\left(1-\frac{1}{j}\right)^L = -\sum_{j=1}^{m}\sum_{l=1}^{L}\binom{L}{l}\left(-\frac{1}{j}\right)^l = \sum_{l=1}^{L}(-1)^{l+1}\binom{L}{l}H^{(l)}_m = LH_m + \mathrm{O}(1)\,. \tag{4}$$

In the case of most balanced partition for each $1 \le i \le K$ we have $\left\lfloor \frac{n}{K} \right\rfloor \le n_i \le \left\lceil \frac{n}{K} \right\rceil$. Therefore, from equations 3 and 4 and the fact that $H_{\frac{n}{K}} = \ln n + \mathrm{O}(1)$ we obtain $\mathbf{E}\left[\mathrm{MC}^{K,L}_{L_n;x}\right] = K \cdot L \cdot \ln n + \mathrm{O}(1) = \Theta(\ln n)$ as required. □

Remark 4.6. Suppose that for some graph $G = (V, E)$ representing the network there exists some station $x \in V$ that for all $1 \le r \le D$ the number of r-neighbors of x in G (i.e. $|\{v \in V : d(v, x) = r\}|$) is less or equal to K. It is worth mentioning that in such cases there exists some distribution of the initial values $\vec{T} = (T_v)_{v\in V}$ stored by the stations that leads to the same expected message complexity for x as for the outermost vertex x^* of the line graph with the same number of vertices as G and the dataset $\vec{T}$, i.e. $\mathrm{MC}^{K,L}_{G;x} = \mathrm{MC}^{K,L}_{L_{|V|};x^*}$. It suffices that for each $u \neq v$ such that $d(u, x) = d(v, x) = r$ the initial values T_u and T_v fall into different bins of the histogram. Then, if there is n_j stations with values from j^{th} bin, $1 \le j \le K$, then there is exactly n_j rounds $r_{i_1}, \ldots, r_{i_{n_j}}$ when x receives exactly one Exp(1) vector related to this bin.

In fig. 4 we present the outcomes of simulations for determining the number of transmissions performed during executions of our algorithm for the case when average message complexity is with high probability very close to the upper bound, i.e. for outermost vertices 1 and n of line graph L_n with $V = \{1, \ldots, n\}$ and the initial values $(T_v)_{v\in V}$ stored by stations distributed uniformly over the unit interval. The parameters of the algorithm HistMean were set to $K = L = 20$ (left) and $K = 4$, $L = 50$ (right). For each network size n in the range from 50 to 5000 (with step 10) we performed 100 independent trials. We can observe that the growth rate of the number of messages sent by the outermost vertices is logarithmic in the network size, thus confirming our bound proven in Lemma 4.5 (the gray curve corresponds to the theoretical upper bound on expected number of transmissions which is asymptotically equal to $K \cdot L \cdot \ln n$). Moreover, one can notice that the outcomes of single trials are highly concentrated around the mean, what also coincides with our theoretical results discussed above.

5 EXPERIMENTS

In this section we present the results of numerical experimental analysis of approximation errors of average value returned by our distributed averaging algorithm HistMean described in Section 4. The simulations were performed for different network sizes and various sets of input data representing the values observed initially by the stations. Let us notice that the error of obtained estimates does not depend on the network topology. It may only depend on the number of stations the network consists of and the distribution of values being averaged. As formally proven in Section 3, the topology of underlying connected undirected graph representing the network is irrelevant when considering the precision of obtained estimates and hence we do not want to put too much focus on some particular topologies when discussing evaluation of algorithm's precision. Nevertheless, the network topology has significant impact on the algorithm's running time and the overall message complexity. We showed that the linear allocation of sensors is the worst case both for time complexity and energy efficiency of the proposed averaging procedure, thus this topology was adopted for most of the simulations.

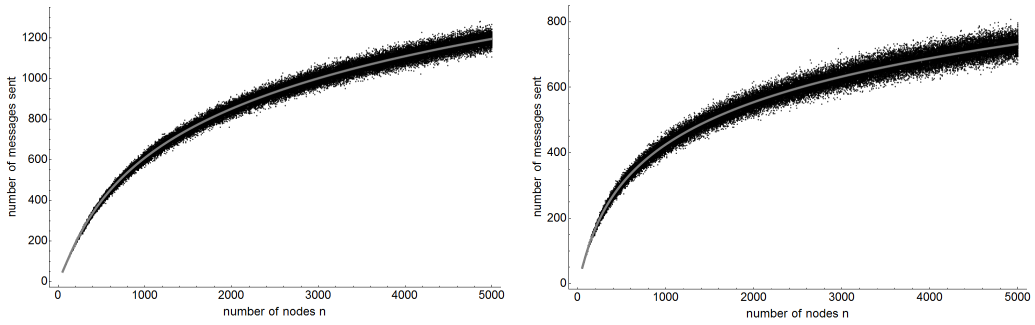

Fig. 4. Estimations of the expected message complexity $\mathbf{E}\left[\mathrm{MC}^{K,L}_{L_n;1}\right]$ obtained using algorithm HistMean with $K = L = 20$ (left) and $K = 4$, $L = 50$ (right) for the outermost vertices of line graph L_n when the stations' initial data are randomly generated from the uniform distribution over the unit interval. The gray curve corresponds to the theoretical upper bound on $\mathbf{E}\left[\mathrm{MC}^{K,L}_{L_n;1}\right] \approx K \cdot L \cdot \ln n$ for most balanced partition of n values into K bins.

At the end of Section 3 we showed that we should check the precision of proposed estimator on a symmetric distribution concentrated at two extreme points. This case will be discussed in Sec. 5.1. In the next section we will show how our estimator behaves on randomly distributed data.

Let us notice that in our experiments we take into account both kinds of errors. The first one is due to the discretization error (see Theorem 2.1) and is controlled by the number K of sub-intervals the range of observed data is divided into. The second one is due to probabilistic nature of the approximate counters used in our algorithm and it depends on the number L of probabilistic counters attached to every sub-interval.

5.1 Worst case

Fig. 5 depicts the outcomes of simulations for the worst case data for different network sizes n varying from 50 to 10000 with step 10. For each n we performed 100 independent experiments where $n/2$ nodes have the value 0 and the remaining $n/2$ the value 1. The parameters were set to $K = 4$ and $L = 50$. We can observe that in all experiments our algorithm calculates the average with 20% precision.

Fig. 6 shows the maximal and average errors (represented by black squares and gray dots, respectively) for these experiments as a function of the network size. We can observe that regardless of the number of nodes in almost all experiments the average is approximated with 20% precision and the mean error is about 5%.

Finally, Fig. 7 and 8 show the mean and maximal errors, respectively, in experiments performed for data concentrated at two extreme points where a fraction p of n nodes have value 0 and $(1 - p)$ value 1 for $n = 100$, 1000 and 10000 and for p from the set $\{0.05i : 1 \leq i < 20\}$. As previously, we chose $K = 4$ and $L = 50$. For each n and p 1000 independent experiments were performed. We can observe that both mean and maximal error of the proposed estimator do not depend on the network size and decrease as the distribution of the values becomes more skewed.

Notice that for data concentrated at end points with the same numbers of values (stations' observation) at both ends the discretization error does not occur, so we are observing only the errors generated by probabilistic counters.

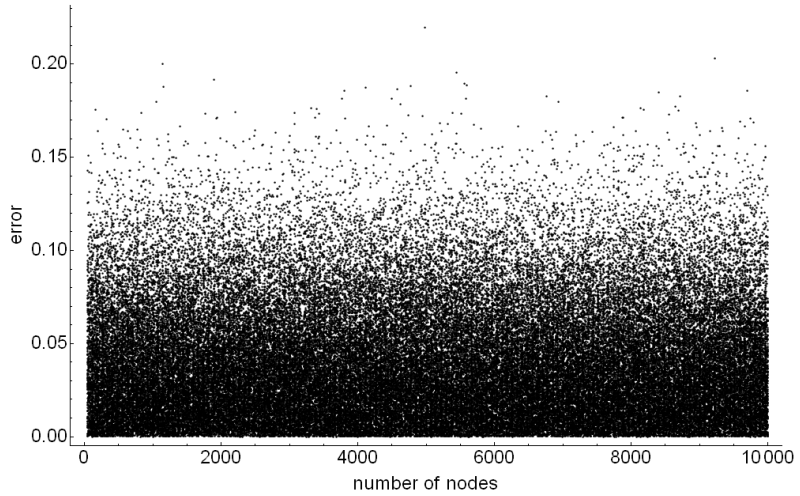

Fig. 5. Errors of algorithm HistMean with parameters $K = 4$, $L = 50$ for data concentrated at end points with respect to the number of nodes in the network.

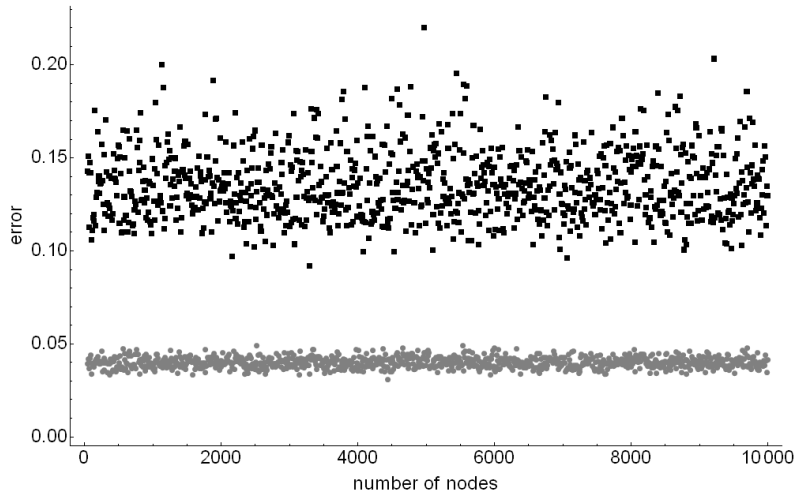

Fig. 6. Maximal (black squares) and average (gray dots) errors of algorithm HistMean with parameters $K = 4$, $L = 50$ for data concentrated at end points with respect to the number n of nodes in the network. For each n we ran 100 experiments.

5.2 Uniform and Normal Distribution

Fig. 9 presents the outcomes of simulations of the algorithm HistMean for different network sizes n for the case where randomly generated data are distributed *uniformly* over the unit interval. We performed 100 independent experiments for each n in the range from 50 to 5000 with step 10. In

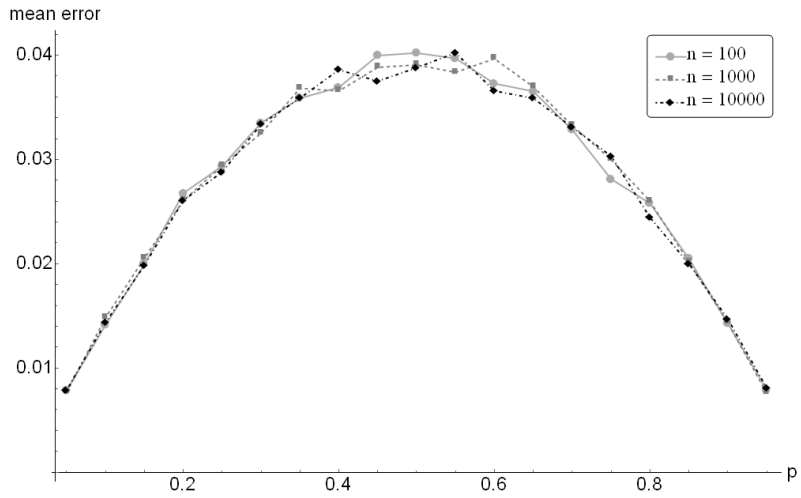

Fig. 7. Mean errors of algorithm HistMean with parameters $K = 4$, $L = 50$ for data concentrated at end points with respect to the fraction p of nodes with minimal value. Experiments were repeated independently 1000 times for networks of size 100, 1000 and 10000.

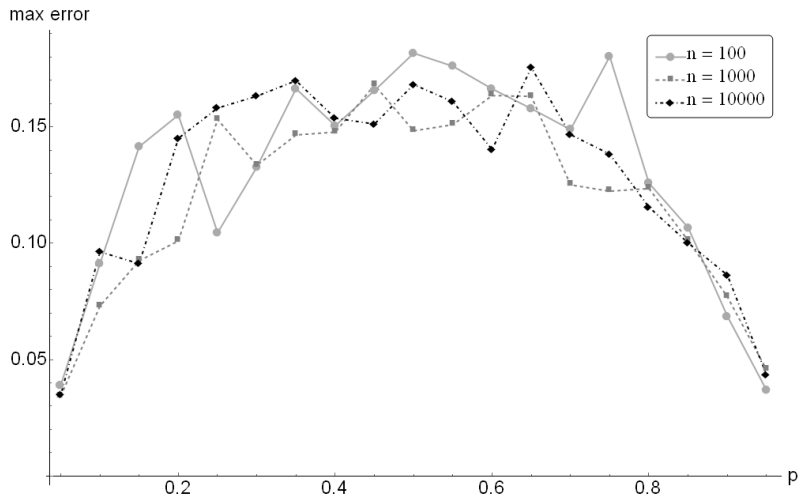

Fig. 8. Maximal errors of algorithm HistMean with parameters $K = 4$, $L = 50$ for data concentrated at end points with respect to the fraction p of nodes with minimal value. Experiments were repeated independently 1000 times for networks of size 100, 1000 and 10000.

each experiment the interval between the minimal and maximal value was split into $K = 20$ equal sub-intervals and $L = 20$ probabilistic counters were used. The maximal and average errors as a function of the network size are shown in Fig. 10. We can see that for each n the mean error of our estimator is below 2% and in all experiments the approximation error does not exceed 8%.

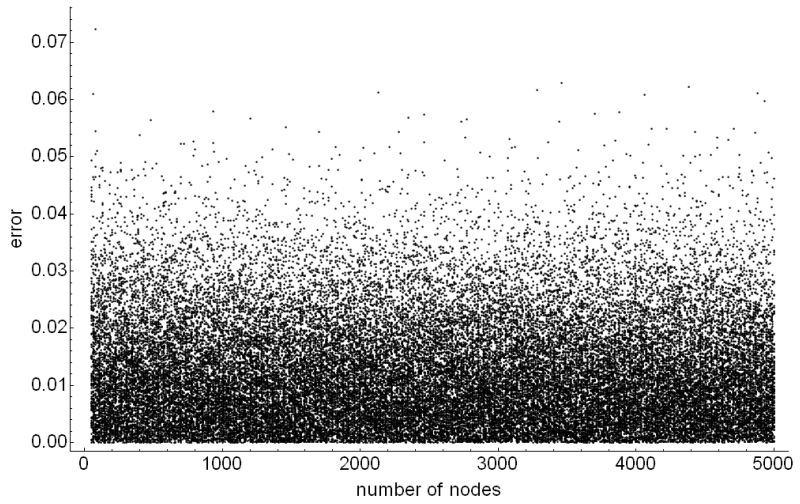

Fig. 9. Errors of algorithm HistMean with parameters $K = 20$, $L = 20$ for randomly generated data from uniform distribution over $[0, 1]$ with respect to the number of nodes in the network.

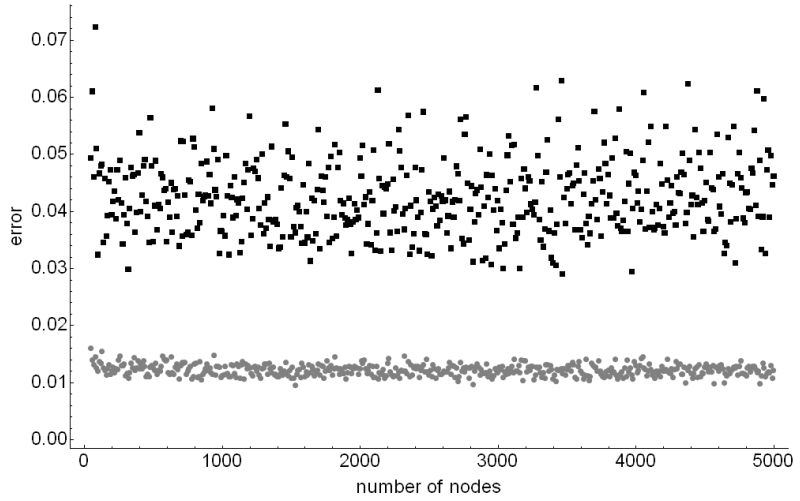

Fig. 10. Maximal (black squares) and average (gray dots) errors of algorithm HistMean with parameters $K = 20$, $L = 20$ for randomly generated data from uniform distribution over $[0, 1]$ with respect to the number n of nodes in the network. For each n we ran 100 experiments.

We performed similar experiments to the previous ones for random data following the *normal distribution* with mean 1000 and variance equals to 100. As before, for each network size n between 50 and 5000 (with step 10) we ran 100 independent simulations with the same choice of parameters of the algorithm (i.e. $K = L = 20$). Fig. 11 and 12 depict the errors of individual experiments and

the maximal and average errors for each n, respectively. Observe that in this case the average is estimated with 5% precision in each try. Moreover, the mean error of obtained approximation is not greater that 1%. [1]

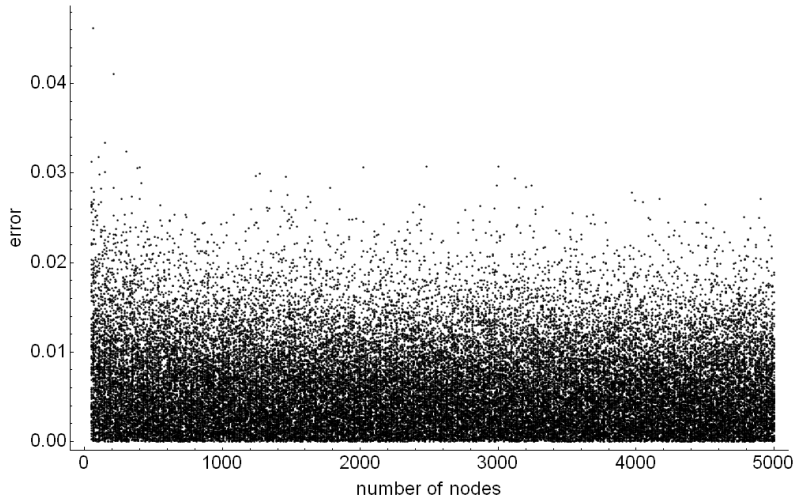

Fig. 11. Errors of algorithm HistMean with parameters $K = 20$, $L = 20$ for random data following normal distribution with mean equals to 1000 and variance 100 with respect to the number of nodes in the network.

5.3 Non-symmetric Distributions

In this section we focus on two examples of non-symmetric distributions used for generating random values for simulations of algorithm HistMean. The first one is the *exponential distribution* with the rate equal to 1. As in the normal and uniform cases, we performed 100 independent experiments for network sizes $n \in \{50, 60, \ldots, 5000\}$. We set both parameters K and L to 20. The outcomes of individual simulations are depicted in Fig. 13 and the maximal and mean values of the approximation errors are presented in Fig. 14. Note that the maximal error of all obtained estimates is less than 5% and the mean error is about 1%. We can also see that for this distribution of observed data the maximal and average errors decrease as network size grows.

Fig. 15 and 16 show the results of simulations (with the same choice of parameters as before) for data generated randomly according to the following skewed distribution. The interval between minimal and maximal values (set to 0 and 100, respectively) was split into $K = 20$ sub-intervals. In almost all of them there is only one observation except some two consecutive intervals i and $i + 1$ with the same number (about $n/2$, where n is the network size) of observed values chosen uniformly at random. In the presented case i was set to 4. We see that for $n > 200$ the maximal and mean errors are below 2% and 1%, respectively. One can also observe that the errors stabilize as n gets larger. In fact, this is consistent with the results from Corollary 2.7. For such distribution of sensory data, for moderate values of n (in the order of thousands) and the choice of $L = 20$ probabilistic counters for each bin the main contribution to the approximation error is due to constant discretization error discussed in Section 2 (cf. Theorem 2.1). It is worth noting that similar behavior of the mean

[1] Certain improvements on the algorithm's accuracy in the case of increasing values of parameter n is due to the fact that the number K is too large for small number of nodes (see the remarks at the end of Section 2.1 regarding probabilistic counters).

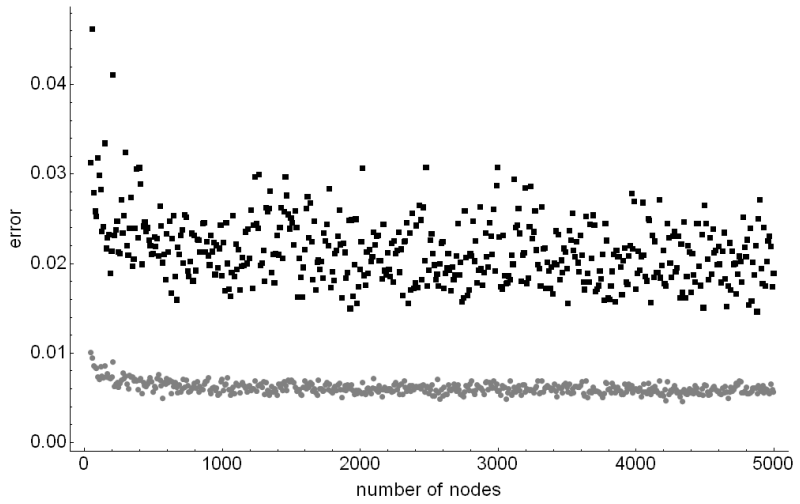

Fig. 12. Maximal (black squares) and average (gray dots) errors of algorithm HistMean with parameters $K = 20, L = 20$ for random data following normal distribution with mean equals to 1000 and variance 100 with respect to the number n of nodes in the network. For each n we ran 100 experiments.

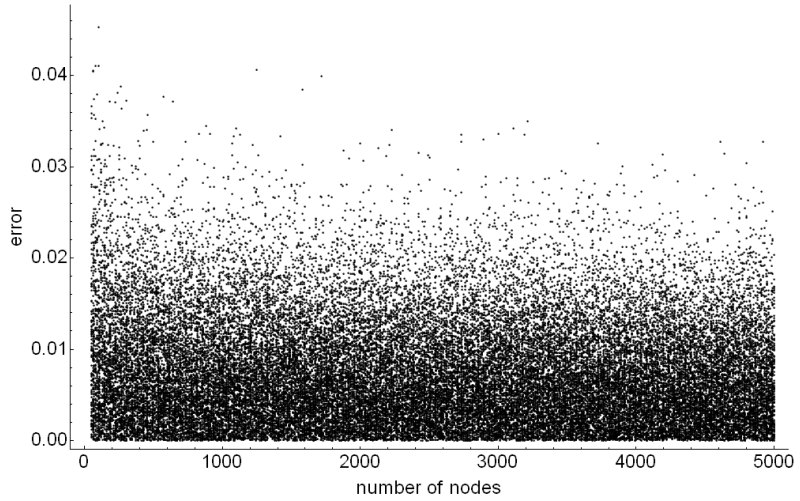

Fig. 13. Errors of algorithm HistMean with parameters $K = 20, L = 20$ for randomly generated data according to exponential distribution with rate 1 with respect to the number of nodes in the network.

and maximal errors can be observed when moving the peak of the distribution towards any of the extreme intervals.

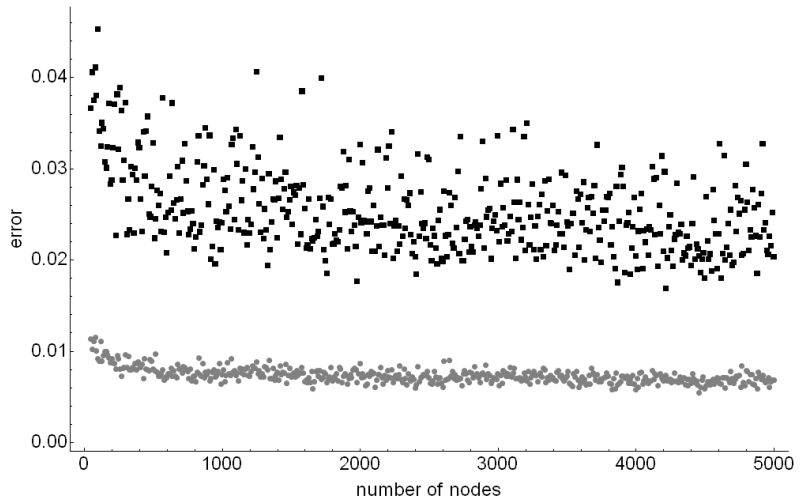

Fig. 14. Maximal (black squares) and average (gray dots) errors of algorithm HistMean with parameters $K = 20, L = 20$ for randomly generated data according to exponential distribution with rate 1 with respect to the number n of nodes in the network. For each n we ran 100 experiments.

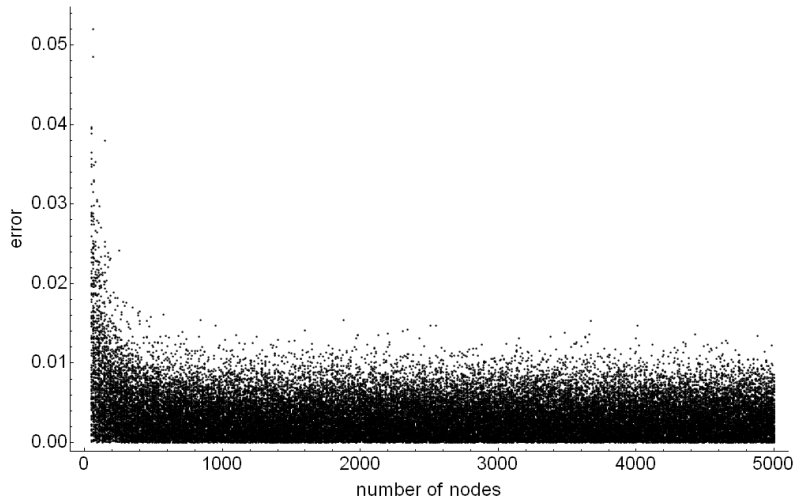

Fig. 15. Errors of algorithm HistMean with parameters $K = 20, L = 20$ for randomly generated data from non-symmetric distribution concentrated at two consecutive intervals i and $i + 1$ for $i = 4$ with respect to the number of nodes in the network.

6 NETWORK SIZE ESTIMATION

One of the examples of exploiting additional data gathered during distributed computation of the average is to make use of the histogram for approximating the number of devices the network

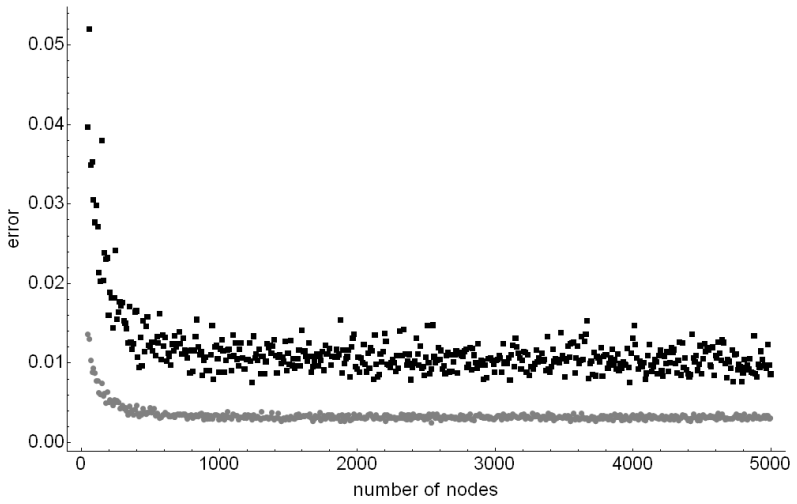

Fig. 16. Maximal (black squares) and average (gray dots) errors of algorithm HistMean with parameters $K = 20, L = 20$ for randomly generated data from non-symmetric distribution concentrated at two consecutive intervals i and $i + 1$ for $i = 4$ with respect to the number n of nodes in the network. For each n we ran 100 experiments.

consists of. Such possibility may be treated as a "side effect" of our algorithm. This can be done by adopting the idea of network size estimation proposed in [Baquero et al. 2012].

Let us consider the scenario where the number of stations n is initially not known and suppose that the parameters K and L of the algorithm HistMean are fixed. For an arbitrary fixed sequence $\vec{T}$ of values observed by the stations and the histogram $\vec{H} = (H_i)_{i=1\ldots K}$ of the dataset $\vec{T}$, our algorithm, in addition to the estimate of $\mathrm{avg}(\vec{T})$, results in an approximate histogram $\vec{C}_L = (C_{L,i})_{i=1\ldots K}$. As previously mentioned, each $C_{L,i}$ is an unbiased estimator of the number H_i of observations falling into i^{th} bin with variance equals to $\frac{H_i^2}{L-2}$. Hence, from the linearity of expectation it follows immediately that $\hat{N} = \sum_{i=1}^{K} C_{L,i}$ is an unbiased estimator of the network size n. Moreover, all realizations of exponential random variables used in the algorithm for approximate counting are generated independently, thus the approximate counters $C_{L,i}$ are independent and $\mathbf{var}\left[\hat{N}\right] = \sum_{i=1}^{K} \mathbf{var}\left[C_{L,i}\right] = \frac{1}{L-2}\sum_{i=1}^{K} H_i^2$. One can easily check that for any fixed n this formula for the variance is minimized when the numbers H_i are equal, i.e. $H_i = \frac{n}{K}$ for $i = 1, \ldots, K$, and maximized for highly biased distribution of the observed values with $n - 1$ and 1 values in two outermost bins. In these cases the variance of $\hat{N}$ is $\frac{n^2}{K^2(L-2)}$ and $\frac{n^2-2n+2}{L-2} \approx \frac{n^2}{L-2}$, respectively.

These results coincide with the outcomes of simulations performed for different sets of observed data and different network sizes n. Fig. 17 depicts the results of experiments for estimation of the number of stations in the network using our averaging algorithm HistMean with $K = L = 20$ in the cases when the observed data are distributed uniformly and come from highly biased distribution with $n - 1$ observed values in the first interval and 1 in the last one. For each network size n in the range from 50 to 5000 (with step 10) we performed 100 independent trials.

It is worth mentioning that it is possible to achieve an arbitrary accuracy of estimates of the network size by increasing the number L of counters for each interval. For more detailed discussion

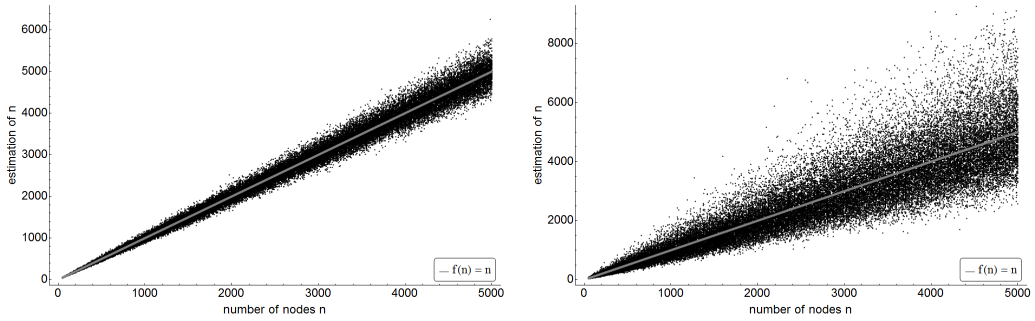

Fig. 17. Estimations of the network size n obtained using algorithm HistMean with $K = 20$ and $L = 20$ for randomly generated data from the uniform distribution over the unit interval (left) and the case when $n - 1$ stations have value 0 and one has value 1 (right).

regarding the precision of network size estimations obtained using this kind of approach we refer to [Baquero et al. 2012].

7 CONCLUSIONS

The proposed in this paper method of computing the average value in distributed environment may be summarized as follows: represent the observed data by a histogram with K bins and use a sequence of L independent probabilistic counters connected with each bin to approximately count the number of observations falling into each bin. This can be done in an efficient way using broadcasting and the extreme propagation technique. The worst case for the presented approach are symmetric data concentrated at two end points. Using 200 probabilistic counters per each node we obtain the precision of order 20% for the worst case data. However, at the end of the execution of our algorithm each node has at its disposal an approximate histogram, so it can recognize this phenomena and may take appropriate actions. It is also possible to utilize these additional data for other purposes, such as obtaining a precise estimation of the number of devices in the network. Moreover, the proposed solution leads to the situation of reaching the true consensus – each station has eventually the same estimate of the actual average of observed values.

ACKNOWLEDGMENTS

The work of the first author was supported by the National Science Centre Poland (NCN) under Grant No. 2013/09/B/ST6/02258. The work of the second author was supported by the Polish Ministry of Science and Higher Education under Research Project No. 0402/0206/16.

The authors would like to thank the anonymous reviewers for their insightful comments and valuable remarks. Their comments have drawn our attention to several publications directly related to the topic of our research. Both authors also wish to thank to professor Mirosław Kutyłowski for his useful hints, remarks and discussions.

REFERENCES

Deepak Ajwani, Andreas Beckmann, Ulrich Meyer, and David Veith. 2012. I/O-efficient approximation of graph diameters by parallel cluster growing - a first experimental study. In *ARCS 2012 Workshops, 28. Februar - 2. März 2012, München, Germany*. 493–504. http://ieeexplore.ieee.org/document/6222204/

Paulo Sérgio Almeida, Carlos Baquero, Martín Farach-Colton, Paulo Jesus, and Miguel A. Mosteiro. 2017. Fault-tolerant aggregation: Flow-Updating meets Mass-Distribution. *Distributed Computing* 30, 4 (01 Aug 2017), 281–291. https://doi.org/10.1007/s00446-016-0288-5

Tuncer Can Aysal, Mehmet Ercan Yildiz, Anand D. Sarwate, and Anna Scaglione. 2009. Broadcast Gossip Algorithms for Consensus. *Signal Processing, IEEE Transactions on* 57, 7 (July 2009), 2748–2761. https://doi.org/10.1109/TSP.2009.2016247

Carlos Baquero, Paulo Sérgio Almeida, and Raquel Menezes. 2009. Fast Estimation of Aggregates in Unstructured Networks. In *ICAS*, Radu Calinescu, Fidel Liberal, Mauricio Marín, Lourdes Peñalver Herrero, Carlos Turro, and Manuela Popescu (Eds.). IEEE Computer Society, 88–93.

Carlos Baquero, Paulo Sérgio Almeida, Raquel Menezes, and Paulo Jesus. 2012. Extrema Propagation: Fast Distributed Estimation of Sums and Network Sizes. *IEEE Trans. Parallel Distrib. Syst.* 23 (2012), 668–675.

Florence Benezit, Patrick Thiran, and Martin Vetterli. 2009. Interval consensus: From quantized gossip to voting. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. 3661–3664. https://doi.org/10.1109/ICASSP.2009.4960420

Patrick Billingsley. 2012. *Probability and Measure* (anniversary ed.). Wiley.

Przemyslaw Blaskiewicz, Miroslaw Kutylowski, Wojciech Wodo, and Kamil Wolny. 2012. Extreme Propagation in an Ad-Hoc Radio Network - Revisited. In *Computational Collective Intelligence. Technologies and Applications - 4th International Conference, ICCCI 2012, Ho Chi Minh City, Vietnam, November 28-30, 2012, Proceedings, Part II*. 142–151.

Miguel Borges, Paulo Jesus, Carlos Baquero, and Paulo Sérgio Almeida. 2012. *Spectra: Robust Estimation of Distribution Functions in Networks*. Springer Berlin Heidelberg, Berlin, Heidelberg, 96–103. https://doi.org/10.1007/978-3-642-30823-9_8

Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. 2006. Randomized Gossip Algorithms. *IEEE/ACM Trans. Netw.* 14, SI (June 2006), 2508–2530. https://doi.org/10.1109/TIT.2006.874516

Jorge C. S. Cardoso, Carlos Baquero, and Paulo Sérgio Almeida. 2009. Probabilistic Estimation of Network Size and Diameter. In *2009 Fourth Latin-American Symposium on Dependable Computing*. 33–40. https://doi.org/10.1109/LADC.2009.19

Chun-houh Chen, Wolfgang Hrdle, Antony Unwin, Chun-houh Chen, Wolfgang Hrdle, and Antony Unwin. 2008. *Handbook of Data Visualization (Springer Handbooks of Computational Statistics)* (1 ed.). Springer-Verlag TELOS, Santa Clara, CA, USA.

Flavio Chierichetti, Silvio Lattanzi, and Alessandro Panconesi. 2010. Rumour Spreading and Graph Conductance. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '10)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1657–1663. http://dl.acm.org/citation.cfm?id=1873601.1873736

Jacek Cichoń and Karol Gotfryd. 2016. Average Counting via Approximate Histograms - Preliminary Report. In *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*. 360–365. https://doi.org/10.1109/ISMS.2016.40

Jacek Cichoń, Jakub Lemiesz, Wojciech Szpankowski, and Marcin Zawada. 2012b. Two-phase cardinality estimation protocols for sensor networks with provable precision. In *2012 IEEE Wireless Communications and Networking Conference, WCNC 2012, Paris, France, April 1-4, 2012*. IEEE, 2009–2013. https://doi.org/10.1109/WCNC.2012.6214120

Jacek Cichoń, Jakub Lemiesz, and Marcin Zawada. 2011. On Cardinality Estimation Protocols for Wireless Sensor Networks. In *ADHOC-NOW (Lecture Notes in Computer Science)*, Hannes Frey, Xu Li, and Stefan Rührup (Eds.), Vol. 6811. Springer, 322–331.

Jacek Cichoń, Jakub Lemiesz, and Marcin Zawada. 2012a. On Message Complexity of Extrema Propagation Techniques. In *Ad-hoc, Mobile, and Wireless Networks, ADHOC-NOW 2012 (LNCS)*, Xiang-Yang Li, Symeon Papavassiliou, and Stefan Rührup (Eds.), Vol. 7363. Springer, 1–13.

Jacek Cichoń and Wojciech Macyna. 2011. Approximate Counters for Flash Memory. In *17th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2011, Toyama, Japan, August 28-31, 2011, Volume 1*. IEEE Computer Society, 185–189. https://doi.org/10.1109/RTCSA.2011.81

J. Considine, Feifei Li, G. Kollios, and John Byers. 2004. Approximate aggregation techniques for sensor databases. In *Data Engineering, 2004. Proceedings. 20th International Conference on*. 449–460. https://doi.org/10.1109/ICDE.2004.1320018

Alexandros G. Dimakis, Anand D. Sarwate, and Martin J. Wainwright. 2008. Geographic Gossip: Efficient Averaging for Sensor Networks. *IEEE Transactions on Signal Processing* 56, 3 (2008), 1205–1216. https://doi.org/10.1109/TSP.2007.908946

Philippe Flajolet. 1985. Approximate Counting: A Detailed Analysis. *BIT* 25, 1 (1985), 113–134.

Philippe Flajolet, Éric Fusy, Olivier Gandouet, and et al. 2007. Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm. In *AOFA '07: PROCEEDINGS OF THE 2007 INTERNATIONAL CONFERENCE ON ANALYSIS OF ALGORITHMS*. DMTCS proc. AH, 127–146.

Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. 2012. Networks Cannot Compute Their Diameter in Sublinear Time. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '12)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1150–1162. http://dl.acm.org/citation.cfm?id=2095116.2095207

Wilfried Gansterer, Gerhard Niederbrucker, Hana Straková, and Stefan Schulze Grotthoff. 2013. Scalable and fault tolerant orthogonalization based on randomized distributed data aggregation. 4 (11 2013), 480–488.

George Giakkoupis and Thomas Sauerwald. 2012. Rumor Spreading and Vertex Expansion. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '12)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1623–1641. http://dl.acm.org/citation.cfm?id=2095116.2095245

Maya Haridasan and Robbert van Renesse. 2008. Gossip-based Distribution Estimation in Peer-to-peer Networks. In *Proceedings of the 7th International Conference on Peer-to-peer Systems (IPTPS'08)*. USENIX Association, Berkeley, CA, USA. http://dl.acm.org/citation.cfm?id=1855641.1855654

Zaobo He, Zhipeng Cai, Siyao Cheng, and Xiaoming Wang. 2015. Approximate aggregation for tracking quantiles and range countings in wireless sensor networks. *Theoretical Computer Science* 607 (2015), 381 – 390. https://doi.org/10.1016/j.tcs.2015.07.056 Combinatorial Optimization and Applications.

Stephan Holzer, David Peleg, Liam Roditty, and Roger Wattenhofer. 2014. Brief Announcement: Distributed 3/2-Approximation of the Diameter. In *28th International Symposium on Distributed Computing (DISC), Austin, Texas, USA*.

Franck Iutzeler, Philippe Ciblat, Walid Hachem, and Jérémie Jakubowicz. 2012. New broadcast based distributed averaging algorithm over wireless sensor networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. 3117–3120. https://doi.org/10.1109/ICASSP.2012.6288575

Paulo Jesus, Carlos Baquero, and Paulo Sérgio Almeida. 2015. Flow updating: Fault-tolerant aggregation for dynamic networks. *J. Parallel and Distrib. Comput.* 78 (2015), 53 – 64. https://doi.org/10.1016/j.jpdc.2015.02.003

R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. 2000. Randomized Rumor Spreading. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS '00)*. IEEE Computer Society, Washington, DC, USA, 565–. http://dl.acm.org/citation.cfm?id=795666.796561

Akshay Kashyap, Tamer Başar, and R. Srikant. 2007. Quantized Consensus. *Automatica* 43, 7 (July 2007), 1192–1203. https://doi.org/10.1016/j.automatica.2007.01.002

David Kempe, Alin Dobra, and Johannes Gehrke. 2003. Gossip-Based Computation of Aggregate Information. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*. IEEE Computer Society, 482–491. https://doi.org/10.1109/SFCS.2003.1238221

Ji Li, Siyao Cheng, Zhipeng Cai, Jiguo Yu, Chaokun Wang, and Yingshu Li. 2017. Approximate Holistic Aggregation in Wireless Sensor Networks. *ACM Trans. Sen. Netw.* 13, 2, Article 11 (April 2017), 24 pages. https://doi.org/10.1145/3027488

Robert Morris. 1978. Counting large numbers of events in small registers. *Commun. ACM* 21, 10 (October 1978), 840–842. https://doi.org/10.1145/359619.359627

Suman Nath, Phillip B. Gibbons, Srinivasan Seshan, and Zachary R. Anderson. 2004. Synopsis Diffusion for Robust Aggregation in Sensor Networks. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*. ACM, New York, NY, USA, 250–262. https://doi.org/10.1145/1031495.1031525

David Peleg, Liam Roditty, and Elad Tal. 2012. *Distributed Algorithms for Network Diameter and Girth*. Springer Berlin Heidelberg, Berlin, Heidelberg, 660–672.

Liam Roditty and Virginia Vassilevska Williams. 2013. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. 515–524. http://doi.acm.org/10.1145/2488608.2488673

Jan Sacha, Jeff Napper, Corina Stratan, and Guillaume Pierre. 2010. Adam2: Reliable Distribution Estimation in Decentralised Environments. In *Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems (ICDCS '10)*. IEEE Computer Society, Washington, DC, USA, 697–707. https://doi.org/10.1109/ICDCS.2010.16

D. W. Scott. 1979. On optimal and data-based histograms. *Biometrika* 66, 3 (1979), 605–610.

Christopher G. Small. 2010. *Expansions and Asymptotics for Statistics*. CRC Press, Boca Raton, London, New York. https://books.google.co.uk/books?id=uXexXLoZnZAC

Received November 2016; revised September 2017; accepted January 2018