Network Size Estimation using Distributed Orthogonalization

Ondrej Slučiak, *Student Member, IEEE,* and Markus Rupp, *Senior Member, IEEE*

***Abstract*—We present novel distributed algorithms for estimating the number of nodes in a wireless sensor network without any a-priori knowledge or node preferences. The algorithms originate from distributed forms of Gram-Schmidt orthogonalization algorithms where the goal is to distributively find a set of orthogonal vectors. Using concepts from linear algebra, by finding the number of independent (orthogonal) vectors, we also find the number of nodes in a network.**

***Index Terms*—distributed algorithm, network size, orthogonal vectors, wireless sensor networks.**

I. Introduction

A typical requirement in distributed algorithms is that the number of nodes N is known at each node beforehand. This knowledge is especially crucial in distributed algorithms that are based on consensus algorithms, e.g., [1]–[3].

Known methods for obtaining the number of nodes require, however, either some node *tagging*, when each node has a predefined ID (by a manufacturer) and nodes store tables containing these IDs, or, similarly, an *ordered numbering* scheme is used, in which the nodes exchange and store only the maximal value (max-consensus). Obviously, the drawback of these methods is a necessity of pre-set "keys" on nodes [4].

Another option is to use an average consensus algorithm. Initializing one node to value 1 while all others to 0, the algorithm converges to $1/N$ at each node [5], [6]. The drawback of this method is a preference of one node (leader) to others. The leader can be pre-set, or, found using a max-consensus algorithm [6]. Although it may be argued that this approach is preferably better and simpler than the previous ones, they both possess, fundamentally, the same disadvantages.

Probabilistic approaches have been also proposed (e.g., [7], [8]), where the estimation accuracy grows with the network size and typically an a-priori knowledge of some parameters is required. Also such algorithms are based on some type of a consensus algorithm, e.g., max-consensus [8], and may have complicated stopping criteria [7]. Nevertheless, these methods can provide good estimates in case of large and dynamic networks.

In this paper we present novel algorithms based on a distributed orthogonalization [9]–[11] for obtaining the number of nodes (network size) N in a Wireless Sensor Network (WSN) without any predefined node IDs, non-uniform initialization of the values in the network, or *any* other a-priori knowledge, with *simple local* stopping criteria.

The authors are with the Institute of Telecommunications, Vienna University of Technology, A-1040 Vienna, Austria (e-mail: {Ondrej.Sluciak, Markus.Rupp}@tuwien.ac.at). This work was supported by the Austrian Science Fund (FWF) under project grant S10611-N13 within the National Research Network SISE. Downloadable supplementary material is available at http://ieeexplore.ieee.org, provided by the authors.

Notation: We describe the algorithms from the *global* (network) point of view, i.e., vector $\mathbf{x} \in \mathbb{R}^{N\times 1}$ represents N scalars x_k, one at each node k. Vector $\mathbf{1}$ denotes a (column) vector of all ones. Matrix $\mathbf{W}$ represents the weight matrix which describes the weighted connections in the network [1], [12]. The average degree in such network is denoted by $\bar{d}$. The notation $\left[\mathbf{W}^{(I)}\right]_{k,1:N}$ represents the k-th row of matrix $\mathbf{W}$ after I iterations[1]. An element-wise multiplication is denoted as $\circ$. The operation $\mathbf{X} \circledast \mathbf{Y}$ is defined as follows: Having two matrices $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)$ and $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m)$, the resulting matrix $\mathbf{Z} = \mathbf{X} \circledast \mathbf{Y}$ is a stacked matrix of all matrices $\mathbf{Z}_i$ such that $\mathbf{Z}_i = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i) \circ ((\underbrace{1, 1, \dots, 1}_{i}) \otimes \mathbf{y}_{i+1})$ ($\otimes$ denotes Kronecker product; $\forall i = 1, 2, \dots, m-1$), i.e., $\mathbf{Z} = (\underbrace{\mathbf{x}_1 \circ \mathbf{y}_2}_{\mathbf{Z}_1}, \underbrace{\mathbf{x}_1 \circ \mathbf{y}_3, \mathbf{x}_2 \circ \mathbf{y}_3}_{\mathbf{Z}_2}, \dots, \underbrace{\mathbf{x}_{m-2} \circ \mathbf{y}_m, \mathbf{x}_{m-1} \circ \mathbf{y}_m}_{\dots \mathbf{Z}_{m-1}})$, thus creating a big matrix containing combinations of column vectors. This later corresponds in our algorithm to the off-diagonal elements of the matrix $\mathbf{R}$ (See [11] for more details).

II. Determining network size

Applying the ideas from [9]–[11], where the goal is to find a factorization of a matrix $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m) \in \mathbb{R}^{N\times m}$ such that $\mathbf{A} = \mathbf{QR}$ (matrix $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m) \in \mathbb{R}^{N\times m}$ contains orthogonal column vectors; and the matrix $\mathbf{R} \in \mathbb{R}^{m\times m}$ is an upper-triangular matrix), we propose an algorithm for finding the number of nodes N in a network.

The main idea is the following: Since matrix $\mathbf{Q} \in \mathbb{R}^{N\times m}$ is a matrix of orthogonal vectors, it contains *at most* N *independent* vectors spanning the linear space $\mathbb{R}^N$, i.e., $m \le N$. Thus, finding the maximum number of independent vectors is equivalent to finding the number of nodes N in the network.

A. Static and dynamic average consensus algorithm

The "*static*" distributed average consensus algorithm in a WSN computes at each node an estimate of the global *mean* (average) of distributed initial data $\mathbf{x}(0)$. In every iteration t each node updates its estimate using the weighted data received from its neighbors. The global view delivers

$$\mathbf{x}(t) = \mathbf{W}\mathbf{x}(t-1). \tag{1}$$

The selection of the *weight matrix* $\mathbf{W}$, containing the weighted connections in the network, crucially influences the convergence of the average consensus algorithm [1], [12].

In contrast to the static average consensus algorithm, Eq. (1), which computes the mean of constant values, the "*dynamic*"

[1] $\lim_{I\to\infty} \mathbf{W}^I = \frac{1}{N}\mathbf{1}\mathbf{1}^\top \Rightarrow \lim_{I\to\infty} \left[\mathbf{W}^{(I)}\right]_{k,1:N} = \frac{1}{N}\mathbf{1}^\top, k=1,2,\dots,N.$

ALGORITHM 1: DISTRIBUTED NETWORK SIZE ESTIMATION USING STATIC CONSENSUS (DNS–SC)

At each node k initiate two random values $(a_{k,1}, a_{k,2})$, i.e., from a global point of view the input matrix $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2) \in \mathbb{R}^{N\times 2}$ $(m=2)$.

- Initialization:
$$\overline{\mathbf{q}} = \mathbf{0}, \quad l = 1.$$
- While $|\overline{\mathbf{q}}| < \epsilon \quad (0 < \epsilon \ll 1)$
  1) Repeat for columns $i = l, l+1, \ldots, m$
     a) At each node $q_{k,i} = a_{k,i}$, i.e., $\mathbf{q}_i = \mathbf{a}_i$.
     b) Repeat for columns $j = 1, 2, \ldots, i-1$:
     Using a consensus algorithm, compute the sum of $\mathbf{s}_1 \triangleq \mathbf{q}_j \circ \mathbf{a}_i$ (approximation of the inner product $\langle \mathbf{q}_j, \mathbf{a}_i \rangle$), i.e., at each node k after I consensus iterations obtain
$$\mathbf{R}^{(k)}_{j,i} = \left[\mathbf{W}^{(I)}\right]_{k,1:N} \mathbf{s}_1,$$
$$q_{k,i} = q_{k,i} - q_{k,j}\mathbf{R}^{(k)}_{j,i}.$$
     c) Using a consensus algorithm, compute the sum of $\mathbf{s}_2 \triangleq \mathbf{q}_i \circ \mathbf{q}_i$ (approximation of $\|\mathbf{q}_i\|_2^2$), i.e., at each node k after I consensus iterations obtain
$$\mathbf{R}^{(k)}_{i,i} = \left[\mathbf{W}^{(I)}\right]_{k,1:N} \mathbf{s}_2,$$
$$q_{k,i} = q_{k,i} / \sqrt{\mathbf{R}^{(k)}_{i,i}}.$$
  2) Using a consensus algorithm, compute sum of $\mathbf{s}_3 \triangleq \mathbf{q}_i \circ \mathbf{q}_{i-1}$ (approximation of the inner product $\langle \mathbf{q}_i, \mathbf{q}_{i-1} \rangle$, i.e., at each node k after I consensus iterations obtain
$$\overline{\mathbf{q}} = \left[\mathbf{W}^{(I)}\right]_{k,1:N} \mathbf{s}_3 \tag{3}$$
  3) At each node $k = 1,2,\ldots,N$ add a new random number c_k, i.e.,
     a) $l = m$, b) $m = m+1$, c) $\mathbf{a}_m = (c_1, c_2, \ldots, c_N)^\top$.
- The detected number of nodes $\hat{N}$ is equal to m.

consensus algorithm is, under some conditions, able to track the mean of a slowly time-variant signal $\mathbf{s}(t)$ [10],

$$\mathbf{x}(t) = \mathbf{W}\left[\mathbf{x}(t-1) + \triangle\mathbf{s}(t)\right]. \tag{2}$$

Due to lack of space we refer the reader to [10], [11] for more details about the dynamic average consensus and distributed Gram-Schmidt orthogonalization algorithm. In [11] we provide a deeper analysis of the properties as well as the proof of convergence of the algorithm which we apply in this paper.

B. Distributed estimation of network size

Assuming a static synchronous connected network, we propose two algorithms utilizing the "static" (*DNS–SC*) and "dynamic" (*DNS–DC*) consensus. Both algorithms are applications of the algorithms for *distributed Gram-Schmidt orthogonalization* which we introduced in [10]. Hereafter, the vector $\overline{\mathbf{q}} \in \mathbb{R}^{N\times 1}$ denotes an approximation of the inner product of two consequent vectors $\mathbf{q}_i$ and $\mathbf{q}_j$ at *all* the nodes, i.e., $\overline{\mathbf{q}} = (\overline{q}_1, \overline{q}_2, \ldots, \overline{q}_N)^\top$ (as defined in Eq. (3) in Alg. 1, and $\Psi^{(4)}$ in Alg. 2, respectively), thus, if $\mathbf{q}_i \perp \mathbf{q}_j$, then $\overline{\mathbf{q}} \approx (0, 0, \ldots, 0)^\top$.

1) DNS–SC: Considering the DNS–SC (see Alg. 1), we see that the number of consensus iterations I must be set such that the consensus algorithm has sufficiently converged. This can be obtained by predefining the number of iterations, or, by observing difference (error) in consecutive iterations. If the difference in consecutive steps is smaller than some parameter δ,

ALGORITHM 2: DISTRIBUTED NETWORK SIZE ESTIMATION USING DYNAMIC CONSENSUS (DNS–DC)

At each node k initiate two random values $(a_{k,1}, a_{k,2})$, i.e., from a global point of view the input matrix $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2) \in \mathbb{R}^{N\times 2}$ $(m=2)$.

- Initialization $(t = 0)$:
$$\begin{aligned} \tilde{\mathbf{U}}(0) &= \mathbf{A} \circ \mathbf{A}, & \hat{\mathbf{U}}(0) &= \mathbf{A}, \\ \tilde{\mathbf{Q}}(0) &= \mathbf{A} \circ \mathbf{A}, & \hat{\mathbf{Q}}(0) &= \mathbf{A}, \\ \tilde{\mathbf{P}}(0) &= \mathbf{A} \circledast \mathbf{A}, & \overline{\mathbf{q}}(0) &= \mathbf{a}_1 \circ \mathbf{a}_2. \end{aligned}$$
- While $t < I$,
  1) $t = t+1$,
  2) Compute locally at each node, i.e., globally $(\forall i = 1, 2, \ldots m)$:
$$\begin{aligned} \mathbf{p}_i(t) &= \textstyle\sum_{j=1}^{i-1} \frac{\tilde{\mathbf{p}}_j(t-1) \circ \hat{\mathbf{q}}_j(t-1)}{\tilde{\mathbf{q}}_j(t-1)}, \\ \hat{\mathbf{u}}_i(t) &= \mathbf{a}_i - \mathbf{p}_i(t), \\ \hat{\mathbf{q}}_i(t) &= \frac{\hat{\mathbf{u}}_i(t)}{\sqrt{\tilde{\mathbf{u}}_i(t-1)}}. \end{aligned}$$
  3) At each node $k = 1, 2, \ldots, N$ store a row of $\mathbf{Q}$, i.e.,
$$\mathbf{q}_k(t) = (\hat{q}_{k,1}(t), \hat{q}_{k,2}(t), \ldots, \hat{q}_{k,m}(t)),$$
  4) Aggregate data for the dynamic consensus (cf. Eq. (2))
$$\begin{aligned} \Psi^{(1)} &= \tilde{\mathbf{U}}(t-1) + \hat{\mathbf{U}}(t) \circ \hat{\mathbf{U}}(t) - \hat{\mathbf{U}}(t-1) \circ \hat{\mathbf{U}}(t-1) \\ \Psi^{(2)} &= \tilde{\mathbf{Q}}(t-1) + \hat{\mathbf{Q}}(t) \circ \hat{\mathbf{Q}}(t) - \hat{\mathbf{Q}}(t-1) \circ \hat{\mathbf{Q}}(t-1) \\ \Psi^{(3)} &= \tilde{\mathbf{P}}(t-1) + \hat{\mathbf{Q}}(t) \circledast \mathbf{A} - \hat{\mathbf{Q}}(t-1) \circledast \mathbf{A} \\ \Psi^{(4)} &= \overline{\mathbf{q}}(t-1) + \hat{\mathbf{q}}_m(t) \circ \hat{\mathbf{q}}_{m-1}(t) - \\ &\qquad - \hat{\mathbf{q}}_m(t-1) \circ \hat{\mathbf{q}}_{m-1}(t-1) \end{aligned}$$
$$\underbrace{\mathbf{\Psi}(t)} = \underbrace{\mathbf{X}(t-1)} + \underbrace{\Delta\mathbf{S}(t)}$$
  5) Broadcast $\mathbf{\Psi}(t) = \left(\Psi^{(1)}, \Psi^{(2)}, \Psi^{(3)}, \Psi^{(4)}\right)$ to the neighbors, i.e.,
$$\underbrace{\left(\tilde{\mathbf{U}}(t), \tilde{\mathbf{Q}}(t), \tilde{\mathbf{P}}(t), \overline{\mathbf{q}}(t)\right)}_{\mathbf{X}(t)} = \mathbf{W}\mathbf{\Psi}(t).$$
  6) If $(|\overline{\mathbf{q}}(t)| < \epsilon \quad \wedge \quad t > m)$, then:
     At each node $k = 1, 2, \ldots, N$ add a new random number c_k $(\mathbf{c} = (c_1, c_2, \ldots, c_N)^\top;\ m = m+1)$ and set $t = 0$, i.e.,
$$\begin{aligned} &\mathbf{a}_m = \mathbf{c}, && \tilde{\mathbf{u}}_m(0) = \mathbf{c} \circ \mathbf{c}, && \tilde{\mathbf{q}}_m(0) = \mathbf{c} \circ \mathbf{c}, \\ &\hat{\mathbf{u}}_m(0) = \mathbf{c}, && \hat{\mathbf{q}}_m(0) = \mathbf{c}, && \overline{\mathbf{q}}(0) = \hat{\mathbf{q}}_m(0) \circ \hat{\mathbf{q}}_{m-1}(0). \end{aligned}$$
- The detected number of nodes $\hat{N}$ is equal to m.

the algorithm stops. We discuss the influence of number of iterations I in the Sec. III. The necessity of waiting for the consensus algorithm to converge before computing the next step (see Steps 1b, 1c, and 2 in DNS–SC) may be considered as a drawback [10]. On the other hand, this approach is robust and if the synchronicity among the nodes is guaranteed, the number of the nodes is always obtained exactly (see the simulations).

As the local stopping criteria for the algorithm we propose two approaches: (1) observe the first occurrence of a non-orthogonal vector, i.e., when $|\overline{q}_k| \geq \epsilon$ (denoted as DNS–SC; Alg. 1); (2) locally at each sensor k compute the *local factorization error*, $\|\mathbf{a}_k - \mathbf{q}_k \mathbf{R}^{(k)}\|$, (denoted as *DNS–SC–b* in the Simulation Sec. III). In the case of DNS–SC–b we do not need to compute $\overline{\mathbf{q}}$ (cf. (3)), thus, we reduce the overall number of sent messages in each node from $O(I(\hat{N}-1)(\hat{N}+2))$ to $O(I(\hat{N}-1)(\hat{N}+1))$.

Note that similarly as in [6] the average consensus algorithm in DNS–SC can be easily replaced by any other aggregation algorithm, e.g., an asynchronous push-sum algorithm [5], [9], making the algorithm asynchronous.

2) DNS–DC: To reduce the number of necessary transmissions, we proposed in [10] a novel orthogonalization algorithm based on the dynamic consensus algorithm, Eq. (2). Originating from this approach, we present an algorithm for obtaining the number of nodes (denoted as DNS–DC; see Alg. 2).

The main modification to our algorithm in [10] is that here we do not need to compute *orthonormal* vectors (orthogonal is sufficient) and we do not need to store the whole matrix $\mathbf{R}$. Also, since the proposed decision rule for adding a new column (new random value at each node) is given by the inner product of the last two consequent column vectors $\hat{\mathbf{q}}_{m-1}$ and $\hat{\mathbf{q}}_m$, this has to be computed distributively (see $\mathbf{\Psi}^{(4)}$), as well. If the absolute value of the inner product $|\overline{q}_k(t)|$ at each node is smaller than a given ϵ, we add a new value and find a new vector $\hat{\mathbf{q}}_m$. If the absolute value of the inner product does not drop in I consensus iterations below ϵ, the algorithm stops and the number of nodes is given by the number of stored values m at each node. The number of transmitted messages is dramatically reduced, i.e., each node has to broadcast $O(I\hat{N})$ number of messages (cf. $O(I\hat{N}^2)$ in DNS–SC).

As mentioned in [10], since the factorization error for the DNS–DC behaves in the same manner as the orthogonality error, observing the factorization error locally at each node may serve as a stopping criterion for the orthogonality error. In that case, we have to compute the full matrix $\mathbf{R}$, but we do not need to compute the value of $\overline{\mathbf{q}}(t)$ (i.e., $\Psi^{(4)}$ in Step 4 in DNS–DC). Analoguously to DNS–SC–b, we denote such modified version of the algorithm as *DNS–DC–b* in the Sec. III.

Note that, as we showed in [11], the numerical accuracy of the distributed orthogonalization algorithm highly depends on the condition number of the input matrix and network topology. Therefore, by adding random numbers (c_k) to all nodes, there is no guarantee that the input (global) matrix $\mathbf{A}$ will not be ill-conditioned and thus this may lead to a premature stopping of the DNS–DC and imprecise determination of the network size, an effect we observe in our simulations.

III. Simulation

In our simulations we use the Metropolis weight matrix $\mathbf{W}$ [11], [12]. We note that also for obtaining these weights the knowledge of N is not necessary.

A typical behaviour of the orthogonality error is depicted in Fig. 1, where we observe that after some iterations the orthogonality error does not decrease anymore. This means that after adding the 9th column the algorithm cannot find, within the last I iterations, any vector orthogonal to the previous vectors $\mathbf{Q}$ and stops.

We further study the number of detected number of nodes $\hat{N}$ while changing the size of the network from $N = 2$ to $N = 120$, for two different topologies – complete (fully connected)[2] and geometric (randomly deployed nodes communicating only with the nodes within some radius (WSN)). We consider connected, static networks with synchronous transmissions. In case of a geometric topology we observe, e.g., in Fig. 2a, that with increasing number of nodes the performance

[2] A complete topology is simulated only to allow for comparisons. Naturally, it is straightforward to obtain a number of nodes in a complete topology just by counting the received messages in every node.

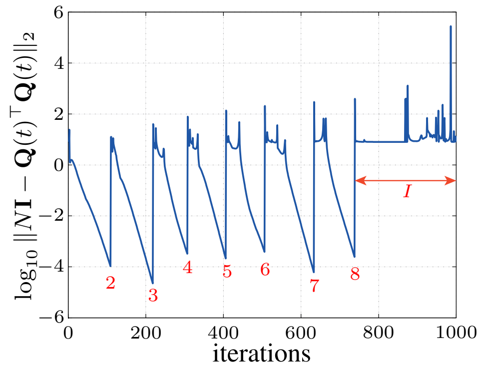


Fig. 1: Example of the evolution of the orthogonality error (DNS–DC) in a geometric network, $\overline{d}=4$, $\hat{N}=8$, $\epsilon = 0.0001$.

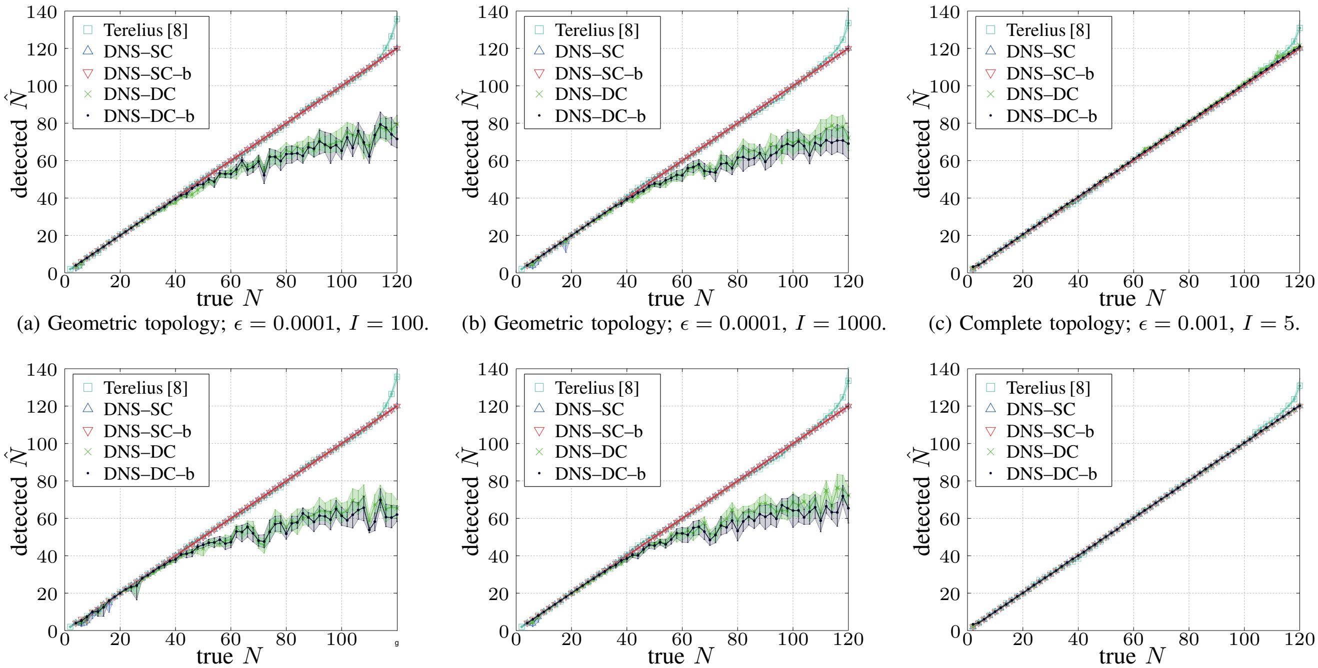
of the algorithms DNS–DC and DNS–DC–b worsens. For a complete topology, the performance depicted in Fig. 2c and Fig. 2f is independent on the number of nodes (lines overlap). Algorithms DNS–SC and DNS–SC–b perform well for all cases. We also observe that the stopping criterion ϵ influences the algorithm's precision. The difference between the two approaches is caused by the usage of the dynamic consensus algorithm and, as mentioned in Sec. II-B2, by its undesirable numerical properties. In Fig. 2 we compare our algorithms with a state-of-the-art algorithm [8], which uses a max-consensus, with overall number of transmitted numbers $O(MI_m)$. Parameters M and γ are defined in [8] and influence the maximum likelihood estimation. I_m is the number of max-consensus iterations and may differ from I (See [8]).

	N \ I	5	20	80	100
DNS–SC	0–40	0	0	0	0
	41–80	0	0	0	0
	81–120	0	0	0	0
DNS–SC–b	0–40	0	0	0	0
	41–80	0	0	0	0
	81–120	0	0	0	0
DNS–DC	0–40	0.0142	0.0142	0.0345	0.0313
	41–80	0.0017	0.0017	0.0016	0.0015
	81–120	0.0013	0.0013	0.0013	0.0018
DNS–DC–b	0–40	0.0412	0.0412	0.0187	0.0327
	41–80	0.0030	0.0030	0.0026	0.0032
	81–120	0.0035	0.0035	0.0035	0.0036
Terelius [8],	0–40	0.0848	0.0839	0.079	0.083
($M=100, \gamma=0$,	41–80	0.083	0.0783	0.0819	0.08
$I_m=I$)	81–120	0.0802	0.0854	0.0817	0.0833

TABLE I: Rel. estimation error. Complete topology, $\epsilon=0.0001$.

	N \ I	100	500	1000	2000
DNS–SC	0–40	0.0371	0.0145	0.0245	0.0075
	41–80	0	0	0	0
	81–120	0	0	0	0
DNS–SC–b	0–40	0.0099	0	0	0
	41–80	0	0	0	0
	81–120	0	0	0	0
DNS–DC	0–40	0.0386	0.0188	0.0253	0.0184
	41–80	0.1559	0.1494	0.1364	0.1188
	81–120	0.3226	0.3115	0.3005	0.2908
DNS–DC–b	0–40	0.0183	0.0038	0.0025	0.0025
	41–80	0.1610	0.1495	0.1322	0.1122
	81–120	0.3478	0.3266	0.3373	0.3085
Terelius [8],	0–40	0.0966	0.0834	0.0828	0.08
($M=100, \gamma=0$,	41–80	0.0806	0.0824	0.0847	0.0787
$I_m=I$)	81–120	0.0825	0.0868	0.084	0.0872

TABLE II: Rel. est. error. Geometric topology, $\epsilon=0.0001$.



(a) Geometric topology; $\epsilon = 0.0001$, $I = 100$. (b) Geometric topology; $\epsilon = 0.0001$, $I = 1000$. (c) Complete topology; $\epsilon = 0.001$, $I = 5$.

(d) Geometric topology; $\epsilon = 0.00001$, $I = 100$. (e) Geometric topology; $\epsilon = 0.00001$, $I = 1000$. (f) Complete topology; $\epsilon = 0.0001$, $I = 5$.

Fig. 2: Detected number of nodes vs. true number of nodes; 95% CI based on 12 runs. For the Alg. [8]: $M = 100, \gamma = 0, I_m = I$.

Furthermore, in Tab. I and Tab. II we show the dependence on the number I of iterations of the consensus algorithms. The shown error is a relative, average detection error over 12 independent initializations (runs) for the given range of number of nodes ($N \in (0, 40)$, $N \in (41, 80)$, $N \in (81, 120)$) and defined as $E_{\text{init}}\{\frac{|\hat{N}-N|}{N}\}$. We observe in Tab. I that for a complete topology, the correct number is obtained for very few consensus steps I. On the other hand, in case of a geometric topology, we observe in Tab. II that even after many consensus iterations, the relative detection error decreases only very slowly or remains constant. In both cases the best performance is achieved for $N < 40$, as can also be seen from Fig. 2a–Fig. 2f. The average relative error of [8] remains almost constant ($\sim 8\%$ for selected parameters $M = 100, \gamma = 0$).

We observe that the DNS–SC and DNS–SC–b perform always better than DNS–DC and DNS–DC–b but at the cost of higher communication. The estimation error in case of DNS–DC and DNS–DC–b is caused by the instability of the algorithm since by adding random numbers we cannot guarantee that the *global* factorized matrix has appropriate numerical properties.

IV. Conclusion

We presented two novel algorithms for estimating the number of nodes N in a network based on distributed Gram-Schmidt orthogonalization. The algorithms do not require any a-priori knowledge on the network topology. By finding the maximum number of orthogonal vectors we find locally at each node the number of nodes in the network. We observed that the estimation error depends on the network topology and the chosen stopping criteria. Although we studied only static networks, it would be of interest to design a distributed algorithm which could dynamically "track" the dimension of a space and thus track time-varying number of nodes.

References

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[2] O. Slučiak, O. Hlinka, M. Rupp, F. Hlawatsch, and P. M. Djurić, "Sequential likelihood consensus and its application to distributed particle filtering with reduced communications and latency," in *Proc. 45th Asilomar Conf. on Sig., Syst., Comp.*, Pacific Grove, CA, USA, Nov. 2011, pp. 1766–1770.

[3] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4334–4349, Aug. 2012.

[4] C. Budianu, S. Ben-David, and L. Tong, "Estimation of the number of operating sensors in large-scale sensor networks with mobile access," *IEEE Trans. on Sig. Proc.*, vol. 54, no. 5, pp. 1703–1715, May 2006.

[5] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proceedings of 44th Annual IEEE Symposium on Foundations of Computer Science*, Oct. 2003, pp. 482–491.

[6] I. Shames, T. Charalambous, C. N. Hadjicostis, and M. Johansson, "Distributed Network Size Estimation and Average Degree Estimation and Control in Networks Isomorphic to Directed Graphs," in *50th Annual Allerton Conference on Communication, Control, and Computing*, 2012.

[7] C. Baquero, P. S. Almeida, R. Menezes, and P. Jesus, "Extrema Propagation: Fast Distributed Estimation of Sums and Network Sizes," *IEEE Trans. on Par. and Dist. Systems*, vol. 23, no. 4, pp. 668–675, 2012.

[8] H. Terelius, D. Varagnolo, and K. H. Johansson, "Distributed size estimation of dynamic anonymous networks," *51st IEEE Conference on Decision and Control*, 2012.

[9] H. Straková, W. N. Gansterer, and T. Zemen, "Distributed QR factorization based on randomized algorithms," in *Proc. of the 9th International Conference on Parallel Processing and Applied Mathematics, Part I*, ser. Lecture Notes in Computer Science, vol. 7203, 2012, pp. 235–244.

[10] O. Slučiak, H. Straková, M. Rupp, and W. N. Gansterer, "Distributed Gram-Schmidt orthogonalization based on dynamic consensus," in *Proc. 46th Asilomar Conf. Sig., Syst., Comp.*, Pacific Grove, CA, Nov. 2012.

[11] O. Slučiak, H. Straková, M. Rupp, and W. N. Gansterer, "Dynamic average consensus and distributed orthogonalization," *IEEE Trans. on Signal Processing*, 2013, (submitted; [Online] Available: http://publik.tuwien.ac.at/files/PubDat_216777.pdf).

[12] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *International Conference on Information Processing in Sensor Networks*, Los Angeles, USA, Apr. 2005, pp. 63–70.