Network topology optimization for accelerating consensus algorithms under power constraints

César Asensio-Marco, Baltasar Beferull-Lozano
Group of Information and Communication Systems
Instituto de Robótica y Tecnologías de la Información & las Comunicaciones (IRTIC)
Universidad de Valencia
46980, Paterna (Valencia), Spain
Email: {Cesar.Asensio, Baltasar.Beferull}@uv.es

***Abstract*—The average consensus algorithm is a well known distributed process in which the nodes iteratively communicate with the nodes within their communication range in order to obtain an estimation of the global average. These repeated communications, when performed in a uniformly randomly deployed network, such as a Wireless Sensor Network, lead to several nodes consuming much more power than others, thus reducing the lifetime of the whole network. This paper proposes a fully distributed method that allows the network nodes to suitably decide which subset of communications provides the best performance during the consensus process in terms of convergence time and power efficiency. Our method simultaneously improves both the convergence of the consensus algorithm and the lifetime of the whole network. Moreover, as a benchmark, we propose a convex optimization problem whose results can be compared with those obtained by our distributed approach. Simulation results are presented to show the efficiency of our proposal, comparing our two methods with existing approaches in the related literature.**

I. Introduction

Wireless Sensor Networks (WSNs) are generally composed by hundreds/thousands of low cost tiny nodes that are limited in terms of power and processing capabilities. These nodes are randomly and uniformly deployed over the area to be sensed and are required to distributively obtain global estimations from locally collected measurements. A well known example of these distributed capabilities is the average consensus algorithm in which nodes iteratively exchange information with all of their immediate neighbors to obtain the global average [1][2]. An interesting variation of this deterministic scheme is the so-called gossip algorithm [4] [5], where each node randomly picks up a neighbor to pair-wise average the data. The major advantage of gossip algorithms is its simplicity, where a random mechanism is introduced. However, in a uniformly randomly deployed network, both protocols drive the network nodes to a different lifetime due to the different number of communications that they perform during the cooperative process. Therefore, it is necessary to design a distributed method to optimize the network topology, so that the power distribution of the nodes becomes more uniform, while both the convergence time and the power consumption are improved. The convergence time, the power consumption and the network lifetime are closely related among themselves. In [7] [8], it has been shown that the total power consumption depends on both the number of iterations needed to converge and the power consumption in each of these iterations, which in turn depends on the network topology and its associated algebraic connectivity [3].

This work was supported by the Spanish MEC Grants TEC2010-19545-C04-04 "COSIMA", CONSOLIDER-INGENIO 2010 CSD2008-00010 "COMONSENS", the European STREP Project "HYDROBIONETS" Grant no. 287613 within the FP7 Framework Programme and by a Telefonica Chair.

Most of the related work that focuses on optimizing the network topology to improve the average consensus algorithm is centralized [7-10] and/or only focuses on minimizing the convergence time [9-12], but not on minimizing the power consumption. Additionally, none of these previous works focuses on optimizing the power distribution among the nodes, which is a crucial factor that affects the network lifetime.

In this work, we focus on improving the convergence time of the average consensus algorithm, while keeping the power consumption at every node bounded. This is achieved by changing the network topology, that is, by adding and removing links from some suitable nodes in such a way that the power consumption needed in one iteration of the consensus and the network lifetime are simultaneously improved. Our main contributions can be summarized as follows:

- The topology is optimized in a distributed way leading to consensus algorithms that need less iterations to converge.
- The power consumption at every iteration is reduced and the maximum power consumption of a node is bounded.
- The previous two points ensure that both the total power consumption and the network lifetime are improved.

The remainder of this paper is structured as follows: some background on consensus problems is presented in Section II. The power and the lifetime model of the network are given in Section III. In Section IV, we present how to improve consensus algorithms by changing the initial power distribution by using a totally distributed method. We then propose, in Section V, an optimization problem to solve the same problem, whose results we use as a benchmark to compare with. Section VI is devoted to validate our claims by comparing our results with existing approaches presented in the literature. Finally, the conclusions of this work are summarized in Section VII.

II. Consensus algorithms

In this section, we revise some graph theory concepts that we use throughout this paper. First, when an information flow exists among the nodes of a network, we can model it as a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, consisting of a set $\mathbf{V}$ of N nodes and a set $\mathbf{E}$ of M links. We denote a link as a pair of nodes (i, j), which is established if the distance between these two nodes, denoted by r_{ij}, is smaller than a certain threshold distance R. Given a graph, we can assign an $N \times N$ adjacency matrix $\mathbf{A}$, where an entry is equal to 1 if $(i, j) \in \mathbf{E}$ and 0 otherwise. The set of neighbors of a node i is defined as $\mathbf{\Omega_i} = \{j \in \mathbf{V} : (i, j) \in \mathbf{E}\}$ and the degree matrix $\mathbf{D}$ is a diagonal matrix, whose entries are given by $d_i = |\mathbf{\Omega_i}|$. Then, the Laplacian of a graph is a matrix defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, whose smallest eigenvalue is equal to zero.

Let us assume that the sensor measurements of nodes have some initial data at time slot $k = 0$. We collect them in a vector, which we call the initial state vector $\mathbf{x}(0)$, thus the average of the initial state $\mathbf{x}(0)$ is $\mathbf{x}_{\text{avg}} = \frac{\mathbf{1}\mathbf{1}^t\mathbf{x}(0)}{N}$, where $\mathbf{1}$ denotes the all ones column vector. We consider the general linear update of the state of each sensor i at time k, using only local data exchange, namely:

$$\mathbf{x}_i(k+1) = \sum_{j \in \mathbf{\Omega_i}} W_{ij}\mathbf{x}_j(k) \quad \forall i = 1, 2...N \tag{1}$$

where $\mathbf{W}$ denotes the mixing matrix, which in this paper is taken as:

$$\mathbf{W} = \mathbf{I} - \alpha\mathbf{L} \tag{2}$$

where $\mathbf{I}$ denotes the identity matrix and α is a constant independent of time whose optimal value in terms of convergence time, for a given network topology, is given by [9]:

$$\alpha = \frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})} \tag{3}$$

where $\lambda_2(\mathbf{L})$ and $\lambda_N(\mathbf{L})$ represent the second and the N-th largest eigenvalues of the Laplacian matrix.

Since our definition of consensus requires that every node in the network converges to the average, we have to ensure that the network graph is connected, that is, there is a multi-hop path between every pair of nodes. However, when a network is uniformly randomly deployed over the area to be sensed, there exists a probability greater than zero that this network results disconnected [13]. Then, in order to ensure network connectivity with high probability, a large value of R must be generally used. In particular, assuming a unit square area, it has been shown in [4][13] that in order to ensure connectivity with probability at least $1 - \frac{1}{N^2}$, the value of the parameter R has to be greater or equal than $\sqrt{2\log(N)/N}$, which we denote as R_{max}. However, this implies having a large power consumption, as illustrated in Fig. 1, where the probability of connection, denoted by p_{C}, is shown vs. the power spent to establishing the topology. We are interested on reducing the power consumption in the network, but maintaining large values of p_{C}.

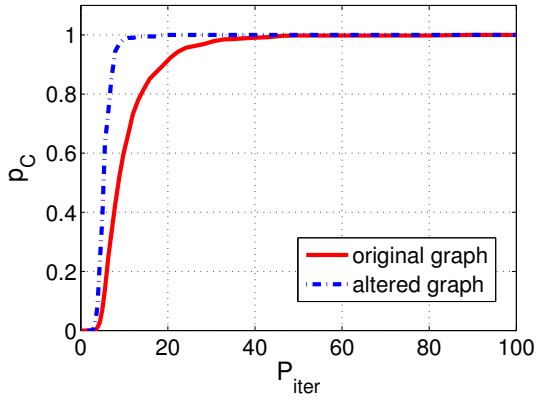

Fig. 1. Probability of ensuring connectivity in a uniformly randomly deployed network of $N = 100$ nodes as a function of the power spent to establish it.

III. Power consumption

In a WSN, the way in which the power consumption is distributed among the nodes determines the performance of the consensus algorithm in terms of convergence time and power requirements. In this work, we consider the simultaneous improvement of three different concepts related to power consumption. First, we focus on reducing the power consumption per iteration, which we denote as P_{iter}. This parameter is defined as the summation of the power consumption of every node in one iteration of the average consensus algorithm:

$$P_{\text{iter}} = \sum_{i \in \mathbf{V}} P_i \tag{4}$$

where $P_i = \sum_{(i,j) \in \mathbf{E}} P_{ij}$ and P_{ij} denotes the power that node i requires to reach node j. For simplicity, we assume that $P_{ij} = r_{ij}^{\gamma}$, where $\gamma \geq 2$ is the path loss exponent. More realistic models are discussed in [14]. Then, P_{iter} influences the network connectivity and the degree distribution of the nodes. Fig. 1 shows that by redistributing some power from the high degree nodes to the low degree nodes, as we explain later, we can reduce P_{iter} while having large values of p_{C}.

Since this work is focused on consensus algorithms that require all the nodes to achieve consensus [1][2], we use the concept of network lifetime presented in [15][16], namely, the network lifetime is defined as the time until the first node run out of batteries, however our results are easily extensible to several nodes failure. Then, in order to increase the lifetime of the network, we need to reduce the maximum power consumption per node, that is, $P_{i_{\text{max}}} = \max(P_1, P_2, ..., P_N)$.

Finally, we focus on reducing the total power consumption, which is defined as the product of P_{iter} times the number of iterations needed to achieve consensus. Therefore, our goal is to redistribute the power consumption among the nodes in order to improve the convergence rate of the consensus process, while reducing, at the same time, P_{iter} and $P_{i_{\text{max}}}$. It is clear that if these three parameters are simultaneously reduced, the total power consumption and the lifetime of the network are also improved.

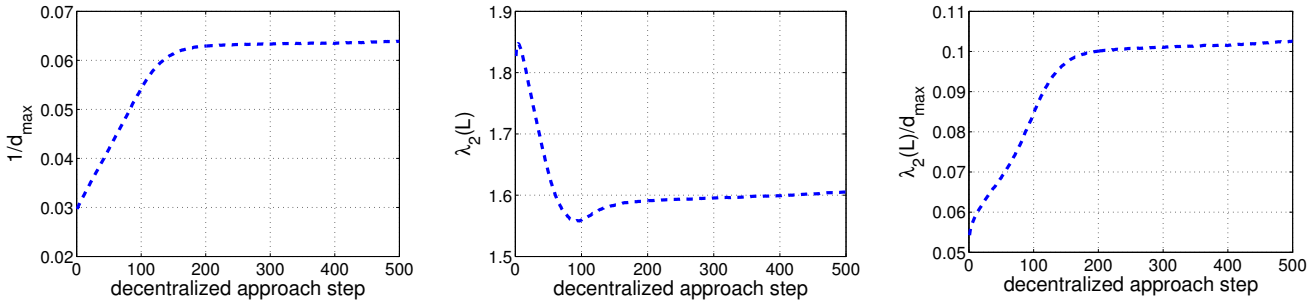

Fig. 2. Average results over 1000 different topologies that are randomly deployed and each of them is composed by 100 nodes. These graphs have been generated by using our decentralized approach. The different parameters $\left\{\frac{1}{d_{\max}}, \lambda_2(\mathbf{L}), \frac{\lambda_2(\mathbf{L})}{d_{\max}}\right\}$ as a function of the decentralized approach step are shown from left to right. Note that this leads to reduce $\lambda_2(\mathbf{W}) = 1 - \frac{1}{d_{\max}}\lambda_2(\mathbf{L})$, which is the eigenvalue that determines the convergence time [9].

IV. Decentralized approach

Since our emphasis, in this paper, is to develop a distributed method, the network nodes are limited to only using local communications and local information to decide which links are used or not during the consensus process. This implies certain restrictions such as the impossibility of using the optimal α (3), because every node would need to know the spectral properties of $\mathbf{L}$. Then, this value of α cannot be calculated by each node in a practical scenario due to its inherent complexity. This motivates us to find a low complexity distributed method that employs a value of α that does not require global knowledge. A simple expression for α that ensures convergence [9] is $\alpha = \frac{1}{d_{\max}}$, which can be easily calculated in a distributed way.

A uniformly randomly deployed network presents nodes having large and low power consumption, which are generally associated with nodes that communicate with a large and a low number of nodes respectively. We call high degree nodes the former and low degree nodes the latter. Our method follows the intuition given by Fig. 1, which consists on: 1) adding links to the low degree nodes, whose effect is to increase the value of $\lambda_2(\mathbf{L})$ [3], while increasing the value of p_C and 2) reducing the number of links of the high degree nodes, whose effect is to reduce the value of $\lambda_2(\mathbf{L})$, but also to increase α. These two alterations in the topology lead to reduce the value of $\lambda_2(\mathbf{W}) = 1 - \frac{1}{d_{\max}}\lambda_2(\mathbf{L})$, reducing the convergence time as a consequence [9]. Fig. 2 shows the evolution of these parameters and the existence of a point from which the network becomes almost regular ($d_i \approx d_j \ \forall i, j \in \mathbf{V}$) and from which the value of $\lambda_2(\mathbf{W})$ is not further improved.

Algorithm 1 allows the low degree nodes to increase their connectivity, increasing the value of $\lambda_2(\mathbf{L})$ and **Algorithm 2** controls the corresponding extra power consumption by reducing the connectivity of the high degree nodes, which reduces $\lambda_2(\mathbf{L})$ and increases the value of α, as shown in Fig. 2. As a result, a more uniform power distribution is obtained, see Fig. 4. Additionally, in order to explicitly control the network lifetime, we introduce the parameter $0 < \beta_2 \leq 1$. This parameter allows us to control the maximum power consumption among the network nodes, increasing the lifetime of the network as a result.

Finally, Fig. 2 shows that $\lambda_2(\mathbf{W}) = 1 - \frac{1}{d_{\max}}\lambda_2(\mathbf{L})$ is reduced, which leads to improve the convergence rate of the consensus algorithm. This effect together with the effect of keeping P_{iter} bounded, ensures that the power consumption of the whole process is also reduced. Moreover, since the removal of links is only performed between the high degree nodes and the parameter β_2 restricts the maximum power consumption per node and iteration, our algorithm also leads to an extension of the network lifetime.

Algorithm 1 Improving_convergence

Require: $\mathbf{\Omega_i'} = \{j \in \mathbf{V} : R < r_{ij} \leq R_{\max}\}$
Ensure: $P_i \approx \beta_2 P_{i_{\max}}$ || $active\_flag = false$
  $active\_flag = true$
  while $P_i < \beta_2 P_{i_{\max}}$ AND $active\_flag = true$ **do**
    GET a node j from $\mathbf{\Omega_i'}$
    REMOVE j from $\mathbf{\Omega_i'}$
    if $P_j > \beta_2 P_{i_{\max}}$ **then**
      $A(i,j) = 1$, $A(j,i) = 1$
      $d_i = d_i + 1$, $d_j = d_j + 1$
      $P_{\text{extra}} = 2P_{ij}$
      j:Saving_power(P_{extra})
    end if
    if $\mathbf{\Omega_i'}$ is empty **then**
      $active\_flag = false$
    end if
  end while

Algorithm 2 Saving_power

Require: $P_{\text{extra}}, \mathbf{\Omega_j} = \{l \in \mathbf{V} : r_{jl} < R\}$
Ensure: $P_{\text{saved}} \geq P_{\text{extra}}$
  $P_{\text{saved}} = 0$
  while $P_{\text{saved}} < P_{\text{extra}}$ **do**
    GET a node l from $\mathbf{\Omega_j}$
    $A(j,l) = 0$, $A(l,j) = 0$
    $d_j = d_j - 1$, $d_l = d_l - 1$
    $P_{\text{saved}} = P_{\text{saved}} + 2P_{jl}$
  end while

V. Centralized approach

In this section, we assume a centralized scheme in order to obtain a benchmark algorithm to which our distributed method can be compared with. In this centralized method, since global knowledge is available, we use the optimal α (3) in the structure of the weights given by (2). Given these two expressions, the convergence time of the average consensus algorithm has been shown in [9] to be directly dependent on $\lambda_2(\mathbf{W})$. Therefore, our centralized method is based on solving a semi-definite program (SDP) for computing the topology in which $\lambda_2(\mathbf{W})$ is minimum, while the resulting power consumption and the lifetime of the network are simultaneously improved. First of all, the optimization problem to minimize the convergence time can be formulated as:

$$\begin{array}{ll} \text{minimize}_{\{\mathbf{Z},\mathbf{Y}\}} & \lambda_2(\mathbf{W}) \\ \text{s. t.} & \mathbf{W} = \mathbf{I} - \alpha(\mathbf{Z}-\mathbf{Y}) \\ & Y_{ij} = 0 \;\text{ if }\; r_{ij} > R_{\max} \\ & Y_{ij} \in \{0,1\} \;\text{ if }\; r_{ij} \leq R_{\max} \\ & Z_{ij} \geq 1 \text{ if } i = j \\ & Z_{ij} = 0 \text{ if } i \neq j \\ & \mathbf{Y} = \mathbf{Y}^T,\; (\mathbf{Y1})_i = Z_{ii} \end{array}$$

where the degree matrix $\mathbf{D}$ has been renamed as $\mathbf{Z}$ and the adjacency matrix $\mathbf{A}$ as $\mathbf{Y}$ in order to clarify that they are now variables. Note that instead of looking for the optimal weights $\mathbf{W}$ as in [9], we are interested in the optimal topology $\mathbf{A}$ that provides the minimum convergence time. However, this initial optimization problem is not convex, thus the third constraint must be relaxed in order to convexify the problem. Moreover, we need to add extra constrains in order to control the lifetime of the network and the power consumption in each iteration. Finally, by using standard optimization tools [17], the problem can be cast as the following SDP:

$$\begin{array}{ll} \text{minimize}_{\{s,\mathbf{Z},\mathbf{Y}\}} & s \\ \text{s. t.} & \mathbf{W} = \mathbf{I} - \alpha(\mathbf{Z}-\mathbf{Y}) \\ & Y_{ij} = 0 \;\text{ if }\; r_{ij} > R_{\max} \\ & 0 \leq Y_{ij} \leq 1 \;\text{ if }\; r_{ij} \leq R_{\max} \\ & Z_{ij} \geq 1 \text{ if } i = j \\ & Z_{ij} = 0 \text{ if } i \neq j \\ & \mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N} \preceq s\mathbf{I} \\ & \mathbf{Y} = \mathbf{Y}^T,\; (\mathbf{Y1})_i = Z_{ii} \\ & P^{\mathbf{Y}}_{\text{iter}} \leq \beta_1 P^{\mathbf{A}}_{\text{iter}},\; P^{\mathbf{Y}}_i \leq \beta_2 P^{\mathbf{A}}_{i_{\max}} \end{array}$$

The first five constraints are given by the structure of the Laplacian matrix. The sixth constraint guarantees the minimization of the convergence time, while the seventh and eighth are necessary to ensure the stability of the consensus process, ensuring the matrix $\mathbf{W}$ to be doubly stochastic. Finally, the last two constraints impose the power restriction. In particular, we ensure that the power consumption per iteration step in the solution graph $P^{\mathbf{Y}}_{\text{iter}}$ is smaller than in the original graph $P^{\mathbf{A}}_{\text{iter}}$ by taking values of $\beta_1 \leq 1$. Moreover, the maximum power consumption of a node is bounded by the effect of β_2, which allow us to control the power distribution among the nodes.

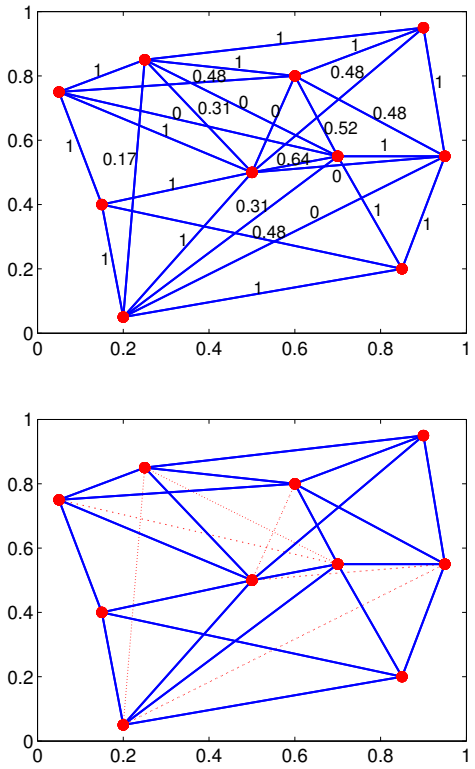

Fig. 3. Illustrative example of the resulting weights of a network with 10 nodes. Originally, $P_{\text{iter}} = 11 \times 10^3$. Increasing R to $R_{\max}$, we have $P_{\text{iter}} = 14.6 \times 10^3$. Solving the optimization problem and projecting the solution, we obtain $P_{\text{iter}} = 9.5 \times 10^3$. The corresponding values of $\lambda_2(\mathbf{W})$ are 0.69, 0.53 and 0.6 respectively.

We are interested on the value of β_2 that maximizes the network lifetime for a given value of P_{iter}. Note that a value of $\beta_2 = 1$ ensures fastest convergence, but we are interested in a topology that provides the maximum network lifetime, which can be associated with slower convergence. The relation between the network lifetime and β_2 is shown in Fig. 5.

Finally, the adjacency matrix that we directly obtain from the relaxed optimization problem can be interpreted in a probabilistic way. It actually provides a probabilistic connectivity pattern that on average presents a better power distribution and leads to a weights matrix that provides much better convergence rate than the original fixed topology. However, since its values are not binary, this does not correspond with any real topology. Thus, we need to obtain a proper topology from the relaxed solution, where the entries of the matrix $\mathbf{Y}$ are zero or one and the elements of the diagonal of $\mathbf{Z}$ are integers. For this purpose, we project the solution by using a simple thresholding technique, in which the smaller entries of $\mathbf{Y}$ correspond to the removed links. This simple projection technique implies that the high-degree nodes are the ones which lose most of the links because they distribute the power P_i among a larger number of links and this is constrained by $\beta_2 P_{i_{\max}}$. Fig. 3 shows an example of this projection and the resulting values of P_{iter} and $\lambda_2(\mathbf{W})$.

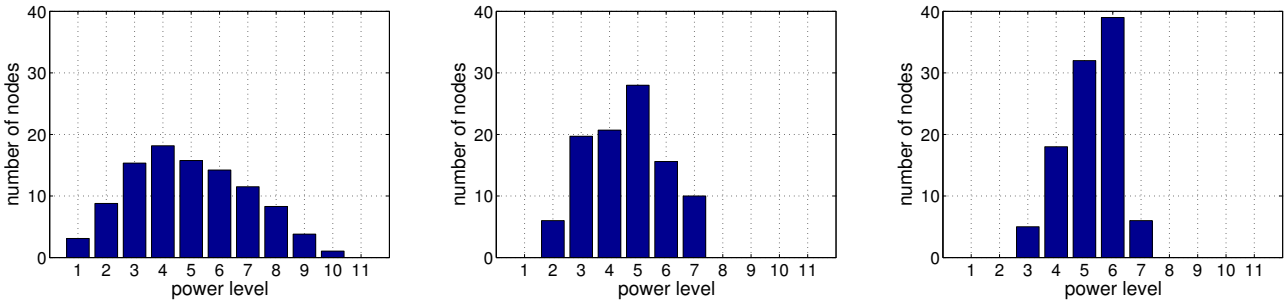

Fig. 4. Power consumption distribution per iteration on the following topologies, from left to right: 1) the original topology, where nodes are connected with the nodes within the initial range, 2) the one resulting from applying our distributed method and 3) the one resulting from applying our centralized method. The value of $P_{i_{\max}}$ is reduced from power level 10 to power level 7 due to the effect of β_2.

VI. Numerical results

In this section, we show the results associated to the network topologies obtained by applying the methods proposed in this paper. In the simulation setup, we use the following values for the main parameters presented in this paper: $\beta_1 = 1$, $\beta_2 = 0.65$ and $\gamma = 2$. All the graphs are presented as average results over 1000 different random topologies composed by 100 nodes and these are deployed over the unit square area. The resulting topologies, when are compared with previous work found in the literature [9][6][10], present the following advantages:

- The convergence rate of the average consensus algorithm is substantially improved because the reduction in the value of $\lambda_2(\mathbf{W})$. Fig. 6 shows this convergence rate when our two methods are used to optimize the network topology. The convergence rate is shown as the maximum deviation from the average value and this is presented in logarithmic scale. This graph shows that our proposed methods present good performance when are compared to [9][6][10], which do not take into account the total power consumption and the lifetime of the network.
- The lifetime of the network is extended. Fig. 4 shows the power distribution of the nodes when our proposed methods are applied. In these graphs, we divide the power consumption per node into ten levels, where the level 10 corresponds to the original value of $P_{i_{\max}}$. Then, we represent the number of nodes having the corresponding power level. It is clear that our methods reduce $P_{i_{\max}}$ because of the effect of β_2, thus increasing the lifetime of the network. This particular effect is shown in detail in Fig. 5, where the parameter β_2 controls the maximum power consumption among the nodes, leading to a more uniform power consumption.
- The total power consumption of the average consensus algorithms is also reduced. Since we are restricting our methods to not increasing the power consumption per iteration and we are also improving the convergence rate, hence the number of iterations, we are reducing the total power consumption that is required to converge. Moreover, since the connectivity of the low degree nodes is increased, p_C is also increased, as shown in Fig. 1.

Finally, Table I, II and III supports numerically our results by comparing all the parameters presented before: convergence rate, total power consumption and lifetime of the network on the topologies resulted from a) applying the two methods proposed in this paper and b) applying the methods presented in the literature [5][6][7]. The lifetime of the network is taken as $\frac{1}{T_{\text{conv}} P_{i_{\max}}}$, where T_{conv} is the number of iterations needed to achieve consensus for a given accuracy of 10^{-6}. Finally, our distributed method performs well when it is compared with the methods presented in the literature regarding the three parameters, and it is competitive when compared with our centralized approach in which global knowledge is used.

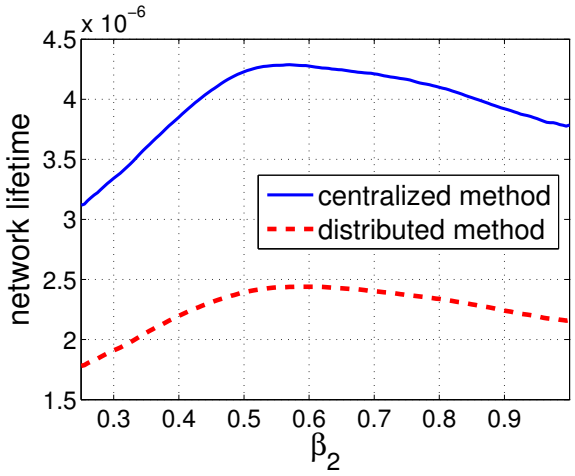

Fig. 5. Lifetime of the network as a function of the parameter β_2.

TABLE I
Simulation Results

Method	T_{conv}	P_{total}	Lifetime
$\alpha = \frac{1}{d_{\max}}$ distributed approach	70	2.5×10^7	$\frac{1}{354000}$
$\alpha = \frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})}$ centralized approach	50	1.6×10^7	$\frac{1}{212200}$
$\alpha = \frac{1}{d_{\max}}$ original topology [9]	144	4.7×10^7	$\frac{1}{866448}$
$\alpha = \frac{2}{\lambda_2(\mathbf{L})+\lambda_N(\mathbf{L})}$ original topology [9]	108	3.4×10^7	$\frac{1}{649836}$
growing well-connected graphs [6]	61	2.1×10^7	$\frac{1}{483680}$
degree-based method [10]	115	2.9×10^7	$\frac{1}{346000}$

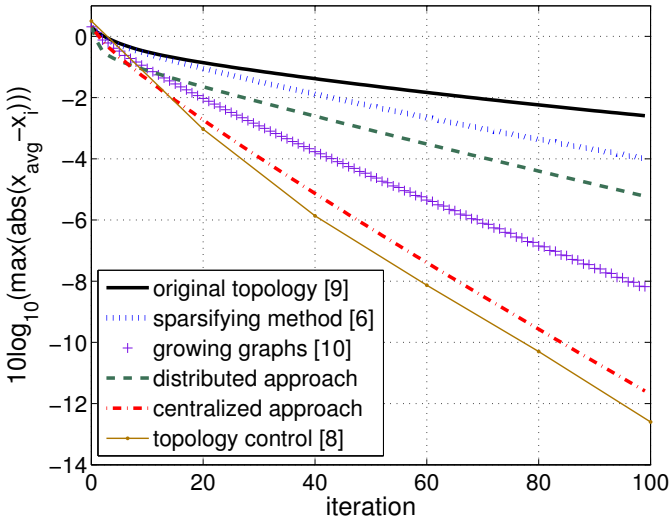

Fig. 6. Convergence time comparison between the two methods proposed in this manuscript and the works presented in [8-10] and [6]. Even the convergence time in [8] is slightly faster, the resulting lifetime of applying our methods is at least 20% longer.

TABLE II
CENTRALIZED METHOD

	T_{conv}	P_{total}	Lifetime	p_C
$\beta_1 = 1, \beta_2 = 0.5$	55	1.7×10^7	$\frac{1}{241200}$	1
$\beta_1 = 1, \beta_2 = 0.75$	48	1.9×10^7	$\frac{1}{228600}$	1
$\beta_1 = 1, \beta_2 = 1$	46	2.2×10^7	$\frac{1}{239100}$	1
$\beta_1 = 1, \beta_2 = 0.6$	50	1.6×10^7	$\frac{1}{212200}$	1
$\beta_1 = 0.9, \beta_2 = 0.6$	51	1.5×10^7	$\frac{1}{216500}$	1
$\beta_1 = 0.8, \beta_2 = 0.6$	54	1.5×10^7	$\frac{1}{228150}$	1

TABLE III
DISTRIBUTED METHOD

	T_{conv}	P_{total}	Lifetime	p_C
$\beta_2 = 0.5$	85	3.0×10^7	$\frac{1}{381000}$	0.995
$\beta_2 = 0.6$	70	2.5×10^7	$\frac{1}{354000}$	0.994
$\beta_2 = 0.75$	68	2.6×10^7	$\frac{1}{373000}$	0.991
$\beta_2 = 1$	62	2.6×10^7	$\frac{1}{398600}$	0.986

VII. CONCLUSIONS

In this paper, we have presented how to optimize the power consumption among the nodes at every iteration of the average consensus algorithm in order to critically reduce the convergence time needed to achieve consensus. In particular, we have proposed both a centralized and a distributed approach to reduce three different concepts of power consumption. We show how to simultaneously reduce the power consumption per iteration and the maximum power consumption per node. The changes on the topology that reduces these power consumptions also focus on reducing the convergence time of the average consensus, then the total power consumption of this process and the lifetime of the network are also improved.

REFERENCES

[1] Xiao, L.; Boyd, S.; Lall, S., "A scheme for robust distributed sensor fusion based on average consensus," *Fourth International Symposium on Information Processing in Sensor Networks*, 2005. IPSN 2005, vol., no., pp. 63-70, 15 April 2005

[2] Olfati-Saber, R.; Fax, J.A.; Murray, R.M., "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE* , vol.95, no.1, pp.215-233, Jan. 2007

[3] Fiedler M., "Algebraic connectivity of graphs," *Journal of Czechoslovak Math*, Vol. 23, pp. 298-305, 1973

[4] Boyd, S.; Ghosh, A.; Prabhakar, B.; Shah, D., "Randomized consensus algorithms," *IEEE Transactions on Information Theory*, vol.52, no.6, pp. 2508-2530, June 2006

[5] Dimakis, A.D.G.; Sarwate, A.D.; Wainwright, M.J.; , "Geographic Gossip: Efficient Averaging for Sensor Networks," *IEEE Transactions on Signal Processing*, vol.56, no.3, pp.1205-1216, March 2008

[6] Asensio-Marco C.; Beferull-Lozano B.; "Accelerating consensus algorithms: Sparsifying networks can be good for you," *IEEE International Conference on Communications*, ICC 2010.

[7] Barbarossa, S.; Scutari, G.; Swami, A., "Achieving Consensus in Self-Organizing Wireless Sensor Networks: The Impact of Network Topology on Energy Consumption," *IEEE International Conference on Acoustics, Speech and Signal Processing*. ICASSP 2007, vol.2, no., pp.II-841-II-844, 15-20 April 2007

[8] S. Sardellitti, S. Barbarossa, A. Swami, "Optimal Topology Control and Power Allocation for Minimum Energy Consumption in Consensus Networks, *IEEE Transactions on Signal Processing*, 2012.

[9] Lin Xiao; Boyd, S., "Fast linear iterations for distributed averaging," *Proceedings. 42nd IEEE Conference on Decision and Control*, 2003. , vol.5, no., pp. 4997-5002 Vol.5, 9-12 Dec. 2003

[10] A. Ghosh and S. Boyd, "Growing Well-Connected Graphs," *Proceedings of the 45nd IEEE Conf. on Decision and Control* (CDC06), pp. 66056611, San Diego, CA, USA, December 2006.

[11] Zhipu Jin; Murray, R.M., "Random consensus protocol in large-scale networks," *Conference on Decision and Control*, 2007 46th IEEE , vol., no., pp.4227-4232, 12-14 Dec. 2007

[12] Sardellitti, S.; Giona, M.; Barbarossa, S.; , "Fast Distributed Average Consensus Algorithms Based on Advection-Diffusion Processes," *IEEE Transactions on Signal Processing*, vol.58, no.2, pp.826-842, Feb. 2010

[13] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions Information Theory*, vol. 46, no. 2, pp. 388-404, Mar. 2000.

[14] S. Vanka, V. Gupta, and M. Haenggi, "Power-Delay Analysis of Consensus Algorithms on Wireless Networks with Interference, *International Journal of Systems, Control, and Communications*, vol. 2, no. 1/2/3, pp. 256-274, 2010.

[15] S. Singh, M. Woo, and C. Raghavendra. "Power-aware routing in mobile ad hoc networks." *In Proceedings ACM International Conference on Mobile Computing and Networking.* Mobicom 98, pages 181-190, 1998.

[16] J. Chang, L. Tassiulas. "Energy conserving routing in wireless ad hoc networks." *In Procedings IEEE International Conference on Computer Communications*. INFOCOM 2000, pages 22-31, 2000.

[17] S. Boyd and L. Vandenberghe, Convex Optimization. *New York: Cambridge Univ. Press*, 2004.