ACHIEVING CONSENSUS IN SELF-ORGANIZING WIRELESS SENSOR NETWORKS:
THE IMPACT OF NETWORK TOPOLOGY ON ENERGY CONSUMPTION

Sergio Barbarossa, Gesualdo Scutari
Univ. of Rome “La Sapienza”
Via Eudossiana 18, 00184 Rome, ITALY
e-mail:{sergio,aldo.scutari}@infocom.uniroma1.it

Ananthram Swami
ARL
Adelphi, MD 20783, USA
email: a.swami@ieee.org

ABSTRACT

Achieving consensus on common global parameters through totally decentralized algorithms is a key problem in wireless sensor networks. In centralized schemes, the need to send sensor data to a fusion center causes congestion around the sink node. With distributed schemes, this congestion is not created, and the network becomes more resilient to node failures and attacks (e.g., an attack on the fusion node could be catastrophic). Distributed agreement algorithms have been studied for a long time; see, e.g. [1]. In recent years, they have received considerable attention, in view of their potential application in sensor networks. The so-called consensus average method is an example of an algorithm achieving consensus in a totally distributed way [2]. Significant contribution have also come from the context of multiagent coordination and flocking [3].

One of the most critical aspects of these consensus algorithms is that they are iterative algorithms where, at each step, the network nodes exchange data among each other to achieve agreement. The network need not be fully connected; indeed, consensus is achieved locally, with local islands of agreement expanding, and resulting in global agreement, under appropriate conditions. Consider a network of \( N \) nodes. Let \( p_{Ti} \) denote the transmit power of the \( i \)-th node, assumed to be constant across the time necessary to achieve consensus. Then, the overall energy spent to reach a common estimate (decision), within a given accuracy (reliability), is

\[
E = T_{\text{conv}} \sum_{i=1}^{n} P_{Ti},
\]

where \( T_{\text{conv}} \) is the time necessary to achieve global consensus. The convergence rate has been derived under a variety of situations and it is strictly related to the network topology. More specifically, modeling the network as a graph described by the Laplacian \( L \), the convergence properties of distributed consensus algorithms depend on the graph spectral properties, i.e., the set of eigenvalues \( \lambda_i(L) \), \( i = 1, \ldots, n \), of \( L \). In particular, if the network is connected, the smallest eigenvalue \( \lambda_1(L) \) is zero and it has algebraic multiplicity one. Furthermore, if the network is connected, the second smallest eigenvalue \( \lambda_2(L) \), known as the network algebraic connectivity, provides important properties about network connectivity. More specifically, in most consensus algorithms, as we will see in the next section, the convergence rate is directly proportional to \( \lambda_2(L) \). Given the critical role played by \( \lambda_2(L) \), many papers have concentrated on how to maximize \( \lambda_2(L) \) in order to minimize the convergence time, either rewiring the networks, i.e. by changing the network topology, as in [4], or by assigning different weights to each link and then optimizing the weight distribution, as in [5].

In general, however, the most critical parameter in wireless sensor networks is typically the energy consumption, which is directly proportional to the convergence time and to the transmit power. On one hand, to save energy, we would like to use the minimum transmit power that ensures network connectivity. But a small transmit power has an effect on the network topology, as in [4], or by assigning different weights to each link and then optimizing the weight distribution, as in [5].

1. INTRODUCTION

Achieving consensus on a common global parameter through totally distributed algorithms is a key problem in wireless sensor networks. In centralized schemes, the need to send sensor data to a fusion center causes congestion around the sink node. With distributed schemes, this congestion is not created, and the network becomes more resilient to node failures and attacks (e.g., an attack on the fusion node could be catastrophic). Distributed agreement algorithms have been studied for a long time; see, e.g. [1]. In recent years, they have received considerable attention, in view of their potential application in sensor networks. The so-called consensus average method is an example of an algorithm achieving consensus in a totally distributed way [2]. Significant contribution have also come from the context of multiagent coordination and flocking [3].

One of the most critical aspects of these consensus algorithms is that they are iterative algorithms where, at each step, the network nodes exchange data among each other to achieve agreement. The network need not be fully connected; indeed, consensus is achieved locally, with local islands of agreement expanding, and resulting in global agreement, under appropriate conditions. Consider a network of \( N \) nodes. Let \( p_{Ti} \) denote the transmit power of the \( i \)-th node, assumed to be constant across the time necessary to achieve consensus. Then, the overall energy spent to reach a common estimate (decision), within a given accuracy (reliability), is

\[
E = T_{\text{conv}} \sum_{i=1}^{n} P_{Ti},
\]

where \( T_{\text{conv}} \) is the time necessary to achieve global consensus. The convergence rate has been derived under a variety of situations and it is strictly related to the network topology. More specifically, modeling the network as a graph described by the Laplacian \( L \), the convergence properties of distributed consensus algorithms depend on the graph spectral properties, i.e., the set of eigenvalues \( \lambda_i(L) \), \( i = 1, \ldots, n \), of \( L \). In particular, if the network is connected, the smallest eigenvalue \( \lambda_1(L) \) is zero and it has algebraic multiplicity one. Furthermore, if the network is connected, the second smallest eigenvalue \( \lambda_2(L) \), known as the network algebraic connectivity, provides important properties about network connectivity. More specifically, in most consensus algorithms, as we will see in the next section, the convergence rate is directly proportional to \( \lambda_2(L) \). Given the critical role played by \( \lambda_2(L) \), many papers have concentrated on how to maximize \( \lambda_2(L) \) in order to minimize the convergence time, either rewiring the networks, i.e. by changing the network topology, as in [4], or by assigning different weights to each link and then optimizing the weight distribution, as in [5].

In general, however, the most critical parameter in wireless sensor networks is typically the energy consumption, which is directly proportional to the convergence time and to the transmit power. On one hand, to save energy, we would like to use the minimum transmit power that ensures network connectivity. But a small transmit power has an effect on the network topology, as in [4], or by assigning different weights to each link and then optimizing the weight distribution, as in [5].

One of the most critical aspects of these consensus algorithms is that they are iterative algorithms where, at each step, the network nodes exchange data among each other to achieve agreement. The network need not be fully connected; indeed, consensus is achieved locally, with local islands of agreement expanding, and resulting in global agreement, under appropriate conditions. Consider a network of \( N \) nodes. Let \( p_{Ti} \) denote the transmit power of the \( i \)-th node, assumed to be constant across the time necessary to achieve consensus. Then, the overall energy spent to reach a common estimate (decision), within a given accuracy (reliability), is

\[
E = T_{\text{conv}} \sum_{i=1}^{n} P_{Ti},
\]

where \( T_{\text{conv}} \) is the time necessary to achieve global consensus. The convergence rate has been derived under a variety of situations and it is strictly related to the network topology. More specifically, modeling the network as a graph described by the Laplacian \( L \), the convergence properties of distributed consensus algorithms depend on the graph spectral properties, i.e., the set of eigenvalues \( \lambda_i(L) \), \( i = 1, \ldots, n \), of \( L \). In particular, if the network is connected, the smallest eigenvalue \( \lambda_1(L) \) is zero and it has algebraic multiplicity one. Furthermore, if the network is connected, the second smallest eigenvalue \( \lambda_2(L) \), known as the network algebraic connectivity, provides important properties about network connectivity. More specifically, in most consensus algorithms, as we will see in the next section, the convergence rate is directly proportional to \( \lambda_2(L) \). Given the critical role played by \( \lambda_2(L) \), many papers have concentrated on how to maximize \( \lambda_2(L) \) in order to minimize the convergence time, either rewiring the networks, i.e. by changing the network topology, as in [4], or by assigning different weights to each link and then optimizing the weight distribution, as in [5].

In general, however, the most critical parameter in wireless sensor networks is typically the energy consumption, which is directly proportional to the convergence time and to the transmit power. On one hand, to save energy, we would like to use the minimum transmit power that ensures network connectivity. But a small transmit power has an effect on the network topology, as in [4], or by assigning different weights to each link and then optimizing the weight distribution, as in [5].

Univ. of Rome ?La Sapienza, Via Eudossiana 18, 00184 Rome, ITALY

Approved for public release; distribution unlimited


Unclassified

Unclassified

Unclassified

Same as Report (SAR)

4

Unclassified
nodes have the same transmit power. The goal of this paper is to derive the transmit power that yields the minimum energy consumption, for a given topology. We will consider networks with deterministic or random spatial distribution of the nodes. Interestingly, the comparison will clarify whether it is better to distribute sensors randomly or over a regular grid, for any given density and coverage area.

2. CONSENSUS ALGORITHMS

Let us denote with \( x_0 \) the \( n \times 1 \) vector containing the data collected by all the nodes. Consensus algorithms typically work by endowing each node with a dynamical system that evolves in time as a function of its own state and of a combination of the states of the other nodes. Each system is initialized with the local measurement. If we denote with \( x(t) \) the state vector, at time \( t \), the evolution of all the states is described by the following equation [4]

\[
\dot{x}(t) = -Lx(t)
\]

where \( L \) is the Laplacian of the graph. The Laplacian is defined as follows. Given an oriented graph \( G \) composed of \( N \) vertices and \( E \) edges, the incidence matrix \( B \) is the \( N \times E \) matrix such that \( [B]_{ij} = 1 \) if the edge \( j \) is incoming to vertex \( i \), \( [B]_{ij} = -1 \) if the edge \( j \) is outgoing from vertex \( i \), and 0 otherwise. Given the \( N \times 1 \) vector \( 1 \), composed of all ones, it is easy to check that the incidence matrix satisfies \( 1^T B = 0^T \). Given \( B \), the symmetric \( N \times N \) matrix \( L \triangleq BB^T \), is called the Laplacian of \( G \), and it is independent of the graph orientation. If we associate a positive number \( a_0 \) to the \( i \)-th edge and we build the diagonal matrix \( D_a \triangleq \text{diag}(a) \), with \( a \triangleq [a_1, \ldots, a_E]^T \), the Laplacian of the corresponding graph (called the weighted Laplacian) is written as \( L_a \triangleq BD_a B^T \). The Laplacian and the weighted Laplacian have several important properties, such as: \( L \) is always positive semi-definite with the smallest eigenvalue equal to 0; the algebraic multiplicity of the null eigenvalue is equal to the number \( n_c \) of connected components of the graph. Hence, if the graph is connected, \( n_c = 1 \) and \( \text{rank}(L) = N - 1 \).

By exploiting the properties of the Laplacian, it is easy to check that if the state vector \( x(t) \) is initialized with the local measurement, i.e. \( x(0) = x_0 \), and the network is connected, then \( x(t) \) converges to the average consensus vector

\[
x(t) \rightarrow \frac{1}{N} 1 1^T x_0.
\]

In words, all states converge to the average value.

In the presence of coupling noise, i.e. adding a noise vector \( v(t) \) to (1), the system still converges in the mean. However, as already noticed in [5], the running average \( a(t) := 1/N \sum_{i=1}^{N} x_i(t) \) is a random walk in this case, so that its variance increases unboundedly with time. To alleviate this problem, an algorithm was derived in [5] to compute the weights \( a_i \) that minimize the mean square deviation, i.e. \( \Delta(t) := E\{\sum_{i=1}^{N} (x_i(t) - a(t))^2\} \).

To avoid the random walk problem, it is sufficient to associate the average consensus to the derivative of the state, rather than to the state itself. An example of this alternative approach was proposed in [6], [7], where the state evolution was described by the following equation, for \( i = 1, \ldots, N \),

\[
\dot{x}_i(t) = g_i(x_0) + \frac{K}{c_i} \sum_{j=1}^{N} a_{ij} f(x_j(t) - x_i(t)), \tag{3}
\]

where \( g_i(x_0) \) is a function of the local measurement \( x_0 \), \( K \) is a global control loop gain; \( c_i \) is a local coefficient that quantifies the attitude of the \( i \)-th sensor to adapt its values as a function of the signals received from the other nodes: The larger \( c_i \) is, the less likely is the \( i \)-th node to change its original decision \( g_i(x_0) \). The function \( f(\cdot) \) is, in general, a non-linear function describing the mutual coupling between sensors and it depends on the radio interface[2]. The running decision, or estimate, of each sensor is encoded in its state derivative \( \dot{x}_i(t) \).

Using the incidence matrix \( B \), we can rewrite the system (3) in compact form as

\[
\dot{x}(t) = g(x_0) - K D_c^{-1} B \Delta f \left( B^T x(t) \right), \tag{4}
\]

where \( x(t) \triangleq [x_1(t), \ldots, x_N(t)]^T \), \( D_c \triangleq \text{diag} \{c_1, \ldots, c_N\} \); \( \Delta \) is an \( E \times E \) diagonal matrix, whose diagonal entries are all the weights \( a_{ij} \), indexed from 1 to \( E \); the symbol \( f(x) \) has to be interpreted as the vector whose \( k \)-th component is \( f(x_k) \). It was proved in [7] that, if the network is connected, the function \( f(x) \) is monotonically increasing and odd[3], and \( K \) is greater than the threshold

\[
K_U = \frac{2 \|D_c \Delta \omega\|_2}{f_{\max} \lambda_2(L_a)}, \tag{5}
\]

where \( f_{\max} = \max f(x) \), \( \Delta \omega := g(x_0) - \omega^* 1 \), with

\[
\omega^* = \frac{c^T g(x_0)}{1^T c} = \frac{\sum_{i=1}^{N} c_i g_i(x_0)}{\sum_{i=1}^{N} c_i}, \tag{6}
\]

then the networks achieves global agreement in the sense that all the state derivatives converge to the common value \( \omega^* \), i.e.

\[
\dot{x}_i(t) \rightarrow \omega^*, \quad \forall i. \tag{7}
\]

This result was used in [6] and [8] to obtain globally optimal decision schemes, for estimation or detection, respectively.

[1] An orientation of a graph \( G \) is the assignment of a direction to each edge.

[2] Without loss of generality, \( f(x) \) is normalized so that \( df(0)/dx = 1 \). A different value of \( df(0)/dx \) can always be included in \( K \).

[3] Linear coupling, i.e., \( f(x) = x \), is then included as a particular case.
in a totally distributed way. In the case of linear coupling \( f_{\text{max}} = \infty \), the critical value \( K_U = 0 \) and the network always converges (provided that it is connected).

3. ENERGY CONSUMPTION

All the consensus techniques presented in the previous section have a convergence rate proportional to the algebraic connectivity \( \lambda_2(L_a) \). Hence, from the point of view of convergence time, they can all be treated using the same formulation. In the case of a weighted Laplacian, \( L_a := BD_aB^T \), the weights \( a_{ij} \) could be chosen to maximize the algebraic connectivity \( \lambda_2(L_a) \) of the weighted graph [5]. However, sometimes, as in the case studied in this paper, the weights are simply the channel coefficients and so they are given.

Let us evaluate now the total energy spent to reach the final estimate, within a specified accuracy. For mathematical tractability reasons, we consider the simple case where all the nodes transmit with the same power \( p_T \). In a wireless network, the connectivity (and thus the Laplacian of the graph) depends on the transmit power of each node (as well as the properties of the propagation medium, such as the power-law attenuation factor, fading and shadowing), so that the algebraic connectivity \( \lambda_2(L_a) \) depends on \( p_T \). The convergence time is proportional to \( 1/\lambda_2(L_a) \). The total energy consumption is then

\[
\mathcal{E} = \frac{N p_T}{\lambda(L_a(p_T))},
\]

where we have made explicit the dependence of \( \lambda(L_a(p_T)) \) on the transmit power \( p_T \).

Let us now see how the network topology changes with \( p_T \). The topological model depends on two sources of randomness, in general: the spatial distribution of the nodes and the channel fading. Furthermore, a link between two nodes is established if the received power is greater than a threshold \( p_{\text{Rmin}} \) (the SINR model).

Let us consider the case with no fading. The received power \( p_R \) depends on the transmit power \( p_T \) and on the transmit/receive distance \( r \) as \( p_R = \frac{p_T}{r^\eta} \), where \( \eta \) is the path loss exponent (typically between 2 and 6). This means that, in the absence of fading, there is a link between any two nodes \( i \) and \( j \) if their distance \( r_{ij} \) is smaller than a critical range \( r_0 := \left( \frac{p_T}{p_{\text{Rmin}}} \right)^{1/\eta} \). If the nodes are randomly distributed on a plane and the links are established according to the previous rule, the graph describing the network is a random geometric graph. We consider a ring topology first, where it is possible to obtain closed form expressions. Then we will consider the more practical planar case.

Ring topology

A ring topology is composed of nodes uniformly spaced over a ring of radius \( R \). This is a regular topology where all the nodes have the same degree, say \( d \). Furthermore, the Laplacian has a circulant Toeplitz structure that facilitates closed-form computation of the algebraic connectivity:

\[
\lambda_2 = 4 \sum_{i=1}^{d/2} \sin^2 \left( \frac{\pi i}{N} \right) = d + 2 \sin \left( \frac{\pi d}{N} \right) \cos \left( \frac{\pi}{N} \left( 1 + \frac{d}{2} \right) \right)
\]

For fixed \( d \), \( \lambda_2 \) decreases as \( N \) increases; for fixed \( d/N \), it increases with \( N \). In our set-up the degree \( d \) is a function of the transmit power, i.e. \( d = d(p_T) \). The total energy consumption is then

\[
\mathcal{E} = \frac{N p_T}{4 \sum_{i=1}^{d(p_T)/2} \sin^2 \left( \frac{\pi i}{N} \right)}
\]

As an example, in Fig. 1, we report the energy consumption, as a function of the transmit power \( p_T \), for different values of the path loss exponent. Interestingly, we observe that when

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{energy_consumption.png}
\caption{Total energy consumption as a function of the local transmit power for a ring topology network.}
\end{figure}

the cost of communication is low (the path exponent is small), it is better to have full connectivity to reach consensus more rapidly. Conversely, when the cost of communication is high (the path loss exponent is high), it is better to have minimum connectivity to save energy.

Planar topology

An interesting question arising in the design of sensor networks is: Given a certain coverage area and number of nodes, is it better to distribute the nodes randomly or over a regular grid? To provide an answer to this question, we considered two planar networks, composed of \( N \) nodes distributed over a square of unit side: one with nodes randomly distributed over the square and the other with nodes placed over a rectangular grid. To get rid of any undesired border effect, we consider a toroidal surface. In both cases, we consider a power decaying law \( p_R = p_T/(1 + r^2) \), to avoid the unrealistic situation in which the receive power is greater than the transmit power.

In Fig. 2, we report the total energy consumption \( \mathcal{E} \), as computed in (8), vs. the transmit power \( p_T \). The number of nodes

\[
\text{\textbf{Fig. 1. Total energy consumption as a function of the local transmit power for a ring topology network.}}
\]
is $N = 100$. In the random graph case, we report a few simulations run over 100 independent generations of the spatial points. From Fig. 2, we can draw two important conclusions: 1) there always exists an optimal $p_T$ that minimizes the overall energy consumption; 2) the random graph requires, practically, the same global energy than the regular graph. As expected, the algebraic connectivity for the two topologies is very similar, and this explains the result of Fig. 2.

The behavior shown in Fig. 2 can be explained by recalling the results of [9], where it was shown that all nodes of a random geometric graph, with nodes located over a unit area toroidal surface, tend to have, for large $N$, the same degree $d = \pi Nr_0^2$, with high probability (i.e., with probability greater than $1 - 1/N^2$). But a uniform grid, located again on a unit area toroidal surface, is also a regular graph with degree approximately equal to $\pi Nr_0^2$, for large $N$. Hence, for large $N$, uniform and random geometric graphs tend to behave similarly. This statement is also confirmed by Fig. 3, where we report the algebraic connectivity for the same case analyzed in Fig. 2. The results shown in this paper represent only a first step in the minimization of the energy necessary to reach a consensus through a decentralized, iterative, mechanism. Several extensions are worth of being analyzed. We have considered, for simplicity, only the case in which the transmit power is the same for all the nodes. This has been a simplifying assumption that has reduced the number of unknowns to one. In general, it will be interesting to formulate the optimization problem where we minimize the total energy consumption $(8)$ with the respect to the vector of powers transmitted by each node $p = (p_1, p_2, \ldots, p_N)$. Another extension concerns the inclusion of proper channel fading models. Moreover, in sensor networks, energy spent for processing by the receiver can be a significant factor that should properly be taken into account.

4. REFERENCES


