Convergence of Distributed WSN algorithms: the wake-up scattering problem

Daniele Fontanelli, Luigi Palopoli, Roberto Passerone *

Dipartimento di Ingegneria e Scienza dell'Informazione
University of Trento - Trento, Italy

Abstract. In this paper, we analyze the problem of finding a periodic schedule for the wake-up times of a set of nodes in a Wireless Sensor Network that optimizes the coverage of the region the nodes are deployed on. An exact solution of the problem entails the solution of an Integer Linear Program and is hardly viable on low power nodes. Giusti et. al. [6] have recently proposed an efficient decentralized approach that produces a generally good suboptimal solution. In this paper, we study the convergence of this algorithm by casting the problem into one of asymptotic stability for a particular class of linear switching systems. For general topologies of the WSN, we offer local stability results. In some specific special cases, we are also able to prove global stability properties.

1 Introduction

In the past few years, Wireless Sensor Networks (WSN) have emerged as one of the most interesting innovations introduced by the ICT industry. Their potential fields of application cover a wide spectrum, including security, disaster management, agricultural monitoring and building automation.

The most relevant feature of a WSN is that it is a dynamic distributed system, in which complex tasks are performed through the coordinated action of a large number of small devices (nodes). The integrity of the network, however, can be affected if nodes become suddenly unoperational, especially when the system is deployed in a remote environment. Therefore, a prominent issue is the ability of the WSN to robustly fulfill its goals, countering possible changes in the environment and/or in the network. The same level of importance is commonly attached to the system lifetime. Since replacing batteries may be too expensive and since even modern scavenging mechanisms cannot drain large quantities of energy from the environment, a WSN is required to minimize energy dissipation. Therefore, the main stay of the research on WSN are distributed algorithms for data processing and resource management that attain an optimal trade-off between functionality, robustness and lifetime.

* This work was supported by the EC under contract IST-2008-224428 "CHAT - Control of Heterogeneous Automation Systems"

A popular way for pursuing this result is the application of "duty-cycling". The idea is to keep a node inactive for a long period of time when its operation is not needed, and then to awake it for a short interval to perform its duties (e.g., to sense the surrounding environment). The performance of the WSN heavily depends on the application and on how this duty-cycling is scheduled. The determination of an optimal duty-cycle schedule has been the subject of intense research. In this paper we focus on the problem of determining a schedule that maximizes the average area "sensed" by the network, given a desired value for the lifetime. While an optimal solution can be found using a centralized formulation [17,4,9,1], we are interested in this paper in analyzing strategies where the schedule is computed online by the nodes themselves.

A very simple and effective heuristic to obtain a suboptimal solution is the wake-up scattering algorithm presented in [6], which we describe in Section 2. The idea is to "scatter" the execution of neighboring nodes, under the assumption that two nodes communicating with each other also share large portions of the covered area, and should therefore operate at distinct times. Experimental evidence suggests that the solution thus found is frequently very close to the optimal one (its distance ranging from 15% to 5%). The schedule is computed from a random solution by iteratively adjusting the wake-up times using information from the neighbors. In this paper, we offer a theoretical study of the algorithm by formally proving its convergence.

Our first contribution is to model the evolution of the system by a state-space description, in which state variables represent the distance between the wake-up times of an appropriate subset of the nodes. The model is generally a switching linear system, in which the dynamic matrix can change depending on the ordering of the distances between the node wake-up times. The problem of convergence of the wake-up scattering algorithm can be cast into a stability problem for this system. For each and every of the linear dynamics composing the switching system, we prove the existence of a subspace composed of equilibrium points for the system. We also show that this equilibrium set is asymptotically stable under the hypothesis that it does not coincide with a switching surface.

This local stability can be strengthened if additional hypotheses are made on the topology of the network. In the particular case in which each node can communicate with the ones whose wake-up times are the closest to its own (the nearest neighbors), we show a particular coordinate transformation, whereby the dynamics of the autonomous linear system are governed by a doubly stochastic dynamic matrix. The stability of this type of systems is well studied [14], and it recently found an interesting application in the consensus problem (see [15,12,13,5] and references therein). In particular, under the restrictive assumptions stated above (visibility of the nearest neighbors), the wake-up scattering problem can be viewed as a deployment task, solved with respect to time, over a cyclic set of possible configurations [10,8,11]. However, as shown below, there are reasonable situations under which switches in the linear dynamics can happen and the classical analysis on consensus problems cannot be applied to the convergence of the wake-up scattering. Intuitively, the reason of this divergence is the fact that while agents moving on a line are "physically" prevented from overtaking each other, this limitations does not apply to the wake-up times of the nodes. Indeed, as shown below, nodes can change their relative time positions if they do not see each other. As a final result for the paper, we prove *global* convergence for some particular topologies for which the visibility of the nearest neighbor does not hold. In our view, this is the first step toward more general global stability results for the wake-up scattering problem.

The paper is organized as follows. In Section 2, we provide some background information about the wake-up scattering algorithm. In Section 3, we construct a state-space model for the evolution of the system. In Section 4, we describe the stability results, which constitute the trunk of the paper. In Section 5, we show some simple numerical examples that clarify the results of the paper and the potential of the algorithm. Finally, in Section 6, we state our conclusions and outline future developments.

2 Background

In this paper we analyze the problem of reducing the power consumption (and therefore extend the lifetime) of a sensor network while providing continuous node coverage over a monitored area. To save power, we switch nodes off for a period of time if another node covering the same area is guaranteed to be active. This technique results in a (typically periodic) schedule of the wake-up intervals of the nodes.

An optimal schedule may be computed either centrally, before deployment, or online by the network itself, in a distributed fashion. Online techniques are preferable in those cases in which the network topology may change, or is not known a priori, and access to a central server is expensive or not available. These techniques typically use information from neighboring nodes to iteratively refine the local schedule [6,19,18,7,3,2]. Of particular interest, in this case, is determining whether the distributed algorithm converges to a solution, how far the solution is from optimal, and how long the transient of the computation lasts.

Here, we consider the scheme proposed by Giusti et al. [6], and focus on the problem of convergence. The considered algorithm computes a periodic schedule over an epoch E, where each node wakes up for only a defined interval of time W to save power. The procedure optimizes the coverage by *scattering* the wake-up times of neighboring nodes (nodes that can communicate directly over the radio channel), i.e., nodes are scheduled so that they wake up as far in time as possible from neighboring nodes. The rationale behind this approach is the assumption that neighboring nodes are more likely to cover the same area. This is true when the sensing range and the radio range are comparable in length. Scattering, in this case, results in a schedule where the wake-up intervals of nodes covering the same area do not overlap, thus achieving a better coverage. While this assumption is clearly an approximation, the technique is extremely simple and relies solely on connectivity, instead of requiring exact position information.

More in detail, the wake-up scattering algorithm proposed in [6] starts from a random schedule and then proceeds in rounds. At every round, nodes broadcast their current wake-up time to all their neighbors. With this information, a node may construct a local copy of the current schedule, limited to information related to its neighboring nodes. By inspecting this schedule, nodes update their wake-up time to fall exactly in the middle between the closest neighboring nodes that wake up immediately before and immediately after their current position in the schedule. This way, a node tries to maximize its distance in time from the closest (in time) neighboring node.

To formalize this procedure, consider N nodes $n_1, \ldots, n_N$ and let E be the duration of the epoch. We denote by $w_i \in [0, E]$ the wake-up time of node n_i. Let also $\mathcal{V}_i$ be the set of nodes visible from node n_i ($i \notin \mathcal{V}_i$). The wake-up time of node n_i at step k is then updated as follows:

$$w_i^{k+1} = (1-\alpha)\, w_i^k + \frac{\alpha}{2}\left(\min_{j\in\mathcal{V}_i}\{w_j^k : w_j^k \geq w_i^k\} + \max_{j\in\mathcal{V}_i}\{w_j^k : w_j^k \leq w_i^k\}\right) \mod E, \tag{1}$$

where $\alpha \in [0, 1]$ controls the speed at which the position of the node in the scheduled is updated during an iteration. The formula is ill-defined if the set $\{w_j^k : w_j^k \geq w_i^k\}$ is empty (because n_i is the last node to wake up in the schedule among its neighbors). In that case, according to the proposed algorithm, the empty set is replaced with the set $\{w_j^k : w_j^k + E \geq w_i^k\}$, i.e., we wrap around the schedule to consider the next nodes to wake up, which will be in the following epoch. A similar wrap around is required when $\{w_j^k : w_j^k \leq w_i^k\}$ is empty. Taking this and the remainder operation into account makes the analysis of the model difficult. In the next section we describe how to simplify the formulation by switching our attention from the wake-up time to the distance in the schedule between the nodes.

3 System Model

To study the convergence of the algorithm in Equation (1), it is convenient to reason about the distance between the nodes (their relative position), rather than about their absolute position in time (with the additional advantage of abstracting away the exact position, which is irrelevant). The distance between two nodes is always positive and between 0 and E. For each pair of nodes (n_i, n_j) we define two distances: one, denoted $\overrightarrow{d}_{i,j}$ that goes forward in time, the other, denoted $\overleftarrow{d}_{i,j}$ that goes backward. Since distances are always positive, we have

$$\overrightarrow{d}_{i,j} = \begin{cases} w_j - w_i & \text{if } w_i \leq w_j, \\ w_j - w_i + E & \text{otherwise.} \end{cases}$$

$$\overleftarrow{d}_{i,j} = \begin{cases} w_i - w_j & \text{if } w_j \leq w_i, \\ w_i - w_j + E & \text{otherwise.} \end{cases}$$

From the definition above it follows that

$$\overleftarrow{d}_{i,j} = E - \overrightarrow{d}_{i,j}, \tag{2}$$

and hence

$$\max_{j\in\mathcal{V}_i}(\overrightarrow{d}_{i,j}) = \max_{j\in\mathcal{V}_i}(E - \overleftarrow{d}_{i,j}) = E - \min_{j\in\mathcal{V}_i}(\overleftarrow{d}_{i,j}).$$

We are interested in computing the new distance between every pair of nodes after an update. To do so, we first compute the amount Δ by which each node moves after the update. This is given by

$$\Delta_i^k = w_i^k - w_i^{k-1} = \frac{\alpha}{2}\left(\min_{l\in\mathcal{V}_i}(\overrightarrow{d}_{i,l}^{\,k}) - \min_{l\in\mathcal{V}_i}(\overleftarrow{d}_{i,l}^{\,k})\right).$$

The distance between two nodes at iteration $k+1$ can be computed as the distance at iteration k corrected by the displacement. Hence,

$$\overrightarrow{d}_{i,j}^{\,k+1} = \overrightarrow{d}_{i,j}^{\,k} - \Delta_i^k + \Delta_j^k, \tag{3}$$

$$\overleftarrow{d}_{i,j}^{\,k+1} = \overleftarrow{d}_{i,j}^{\,k} + \Delta_i^k - \Delta_j^k. \tag{4}$$

The distance between two nodes remains bounded by 0 and E during the iterations, i.e., the distances always belong to the set $\mathcal{S}_E = \{x \in \mathbb{R} | 0 \le x \le E\}$.

Theorem 1. *Let n_i and n_j be nodes that see each other (i.e., $n_j \in \mathcal{V}_i$ and $n_i \in \mathcal{V}_j$). Let $\overrightarrow{d}_{i,j}^{\,0}, \overleftarrow{d}_{i,j}^{\,0} \in \mathcal{S}_E$. Then, for all $k > 0$, $\overrightarrow{d}_{i,j}^{\,k}, \overleftarrow{d}_{i,j}^{\,k} \in \mathcal{S}_E$.*

Proof. By adding (3) and (4) we have $\overrightarrow{d}_{i,j}^{\,k+1} + \overleftarrow{d}_{i,j}^{\,k+1} = \overrightarrow{d}_{i,j}^{\,k} + \overleftarrow{d}_{i,j}^{\,k}$. From (2) we have $\overrightarrow{d}_{i,j}^{\,0} + \overleftarrow{d}_{i,j}^{\,0} = E$, therefore, by induction, $\overrightarrow{d}_{i,j}^{\,k} + \overleftarrow{d}_{i,j}^{\,k} = E$. We will now bound the displacement of the nodes at each iteration.

$$\Delta_i^k = \frac{\alpha}{2}\left(\min_{l\in\mathcal{V}_i}(\overrightarrow{d}_{i,l}^{\,k}) - \min_{l\in\mathcal{V}_i}(\overleftarrow{d}_{i,l}^{\,k})\right) \le \frac{\alpha}{2}\overrightarrow{d}_{i,j}^{\,k}$$

Also, since $\overrightarrow{d}_{i,j} = \overleftarrow{d}_{j,i}$ (proof left to the reader),

$$\Delta_j^k = \frac{\alpha}{2}\left(\min_{l\in\mathcal{V}_j}(\overrightarrow{d}_{j,l}^{\,k}) - \min_{l\in\mathcal{V}_j}(\overleftarrow{d}_{j,l}^{\,k})\right) = \frac{\alpha}{2}\left(\min_{l\in\mathcal{V}_j}(\overleftarrow{d}_{l,j}^{\,k}) - \min_{l\in\mathcal{V}_j}(\overrightarrow{d}_{l,j}^{\,k})\right) \ge -\frac{\alpha}{2}\overrightarrow{d}_{i,j}^{\,k}$$

Therefore $\Delta_i^k - \Delta_j^k \le \frac{\alpha}{2}\overrightarrow{d}_{i,j}^{\,k} + \frac{\alpha}{2}\overrightarrow{d}_{i,j}^{\,k} = \alpha\,\overrightarrow{d}_{i,j}^{\,k} \le \overrightarrow{d}_{i,j}^{\,k}$. Hence, from (3),

$$\overrightarrow{d}_{i,j}^{\,k+1} \ge 0. \tag{5}$$

Similarly, one shows that $\overleftarrow{d}_{i,j}^{\,k+1} \ge 0$. Therefore, since their sum is E and they are positive, we obtain the result.

The theorem above shows that nodes that see each other do not overtake each other after an update, since their distance remains positive and bounded by E.

To prove the stability of the update rule, i.e., that the wake-up intervals converges to a periodic schedule preserving the WSN power while ensuring the area coverage, a state space description is needed. At this point, a straightforward choice for the state vector is to contain all the possible $N(N-1)$ wake–up distances among all the N nodes of the WSN. However, since in the update equations (3) and (4) only the distances between nodes that see each other are involved, a more interesting choice is to select the state variables among such node distances.

Therefore, without loss of generality, consider $\mathcal{V}_i \neq \emptyset$, $\forall i = 1, \ldots, N$, i.e. each node sees at least another node[1]. Consider a state vector $\mathbf{x}$ whose entries are the distances $\overrightarrow{d}_{i,l}$, $\forall l \in \mathcal{V}_i$ and for $i = 1, \ldots, N$. Similarly, let $\mathbf{y}$ be the vector of distances $\overleftarrow{d}_{i,l}$, $\forall l \in \mathcal{V}_i$ and for $i = 1, \ldots, N$. Trivially, the number of elements N_x in the state vector $\mathbf{x}$ depends on the visibility graph.

Introducing the notation $\bar{\Delta}^k_{\overrightarrow{d}} = \frac{\alpha}{2}\left(\min_{l \in \mathcal{V}_j}(\overrightarrow{d}^k_{j,l}) - \min_{l \in \mathcal{V}_i}(\overrightarrow{d}^k_{i,l})\right)$, $\bar{\Delta}^k_{\overleftarrow{d}} = \frac{\alpha}{2}\left(\min_{l \in \mathcal{V}_j}(\overleftarrow{d}^k_{j,l}) - \min_{l \in \mathcal{V}_i}(\overleftarrow{d}^k_{i,l})\right)$, we can rewrite (3) and (4) as:

$$\overrightarrow{d}^{k+1}_{i,j} = \overrightarrow{d}^k_{i,j} + \bar{\Delta}^k_{\overrightarrow{d}} - \bar{\Delta}^k_{\overleftarrow{d}}, \quad (6)$$

$$\overleftarrow{d}^{k+1}_{i,j} = \overleftarrow{d}^k_{i,j} - \bar{\Delta}^k_{\overrightarrow{d}} + \bar{\Delta}^k_{\overleftarrow{d}}. \quad (7)$$

With the proposed choice of the state variables, the update displacement $\bar{\Delta}^k_{\overrightarrow{d}}$ only depends on the distances in $\mathbf{x}$, and $\bar{\Delta}^k_{\overleftarrow{d}}$ only on the distances in $\mathbf{y}$. Rewriting (7) and (8) in matrix notation, yields

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}^{k+1} = \begin{bmatrix} \mathbf{x}^k + \Gamma'_x \mathbf{x}^k + \Gamma''_x \mathbf{y}^k \\ \mathbf{y}^k - \Gamma'_x \mathbf{x}^k - \Gamma''_x \mathbf{y}^k \end{bmatrix}, \quad (8)$$

where matrices Γ'_x, Γ''_x collects the $\pm\alpha/2$ factors.

Observing that the invariance property (2) can be written as $\mathbf{y} = E\mathbf{1} - \mathbf{x}$, where $\mathbf{1} \in \mathbb{R}^{N_x}$ is the column vector with all entries equal to 1, the discrete time evolution of system (**??**) is simplified as:

$$\mathbf{x}^{k+1} = (I_{N_x} + \Gamma'_x - \Gamma''_x)\mathbf{x}^k + E\Gamma''_x\mathbf{1} = A\mathbf{x}^k + bE. \quad (9)$$

Therefore the stability of the system is related to the eigenvalues of A and to the time response to the constant input E.

Since Γ'_x in (**??**) is related only to $\bar{\Delta}^k_{\overrightarrow{d}}$, it contains two entries in each row that are equal to $\alpha/2$ and $-\alpha/2$ respectively. Similar considerations apply to Γ''_x, as summarized in the following theorem.

Theorem 2. *¡ The rows of the system matrices Γ'_x and Γ''_x have exactly two entries that are not equal to zero. Furthermore, the sum of the elements of each rows is zero.*

A consequence of Theorem 2 is that the discrete time system (6) is autonomous

$$\mathbf{x}^{k+1} = (I_{N_x} + \Gamma'_x - \Gamma''_x)\mathbf{x}^k = A\mathbf{x}^k, \quad (10)$$

[1] Blind nodes have no dynamics and do not participate to the dynamic of the other nodes

and it has, at least, one eigenvalue equals to one. Hence, $\mathbf{x}$ contains the number of elements that are sufficient for the whole system dynamic description. Nevertheless, it may contain redundant variables, as the following example highlights.

Example 1. Let us consider a network with $N = 3$, where n_1 sees only n_2 and n_3 sees only n_2. Without loss of generality, assume $w_1 < w_2 < w_3$. The proposed choice of state variables yields $\mathbf{x} = [\overrightarrow{d}_{1,2},\ \overrightarrow{d}_{2,3},\ \overrightarrow{d}_{3,2},\ \overrightarrow{d}_{2,1}]^T$. Notice that $\overrightarrow{d}_{2,1} = E - \overrightarrow{d}_{1,2}$. Hence it can be erased without loss of information. Nonetheless, erasing $\overrightarrow{d}_{2,1}$ (and, hence, substituting the term $\overrightarrow{d}_{2,1} = \overleftarrow{d}_{1,2}$ in $\bar{\Delta}^k_{\overleftarrow{d}}$ of equation (7)) yields to an input vector $\hat{b} = [\alpha/2, 0, 0]^T$ that makes the stability analysis more complex in the general case.

As shown in Section **??**, if two nodes do not see each other, they can overtake each other. This behavior, together with the fact that the updating equations (7) and (8) are nonlinear, makes the matrices Γ'_x and Γ''_x time variant. Therefore, the overall system dynamics is switching. Defining $\sigma(k)$ as the switching signal, that takes values $1, \ldots, S$, the switching system $\mathbf{x}^{k+1} = A_{\sigma(k)}\mathbf{x}^k$ is thus derived, with system matrices $\{A_1,\ A_2,\ \ldots,\ A_S\}$. The region of the state space in which the system evolves using a dynamic A_i is a convex polyhedron delimited by a set of subspaces of the type $x_i < x_j$, for appropriate choices of i and j.

On the other hand, in view of Theorem 1, a node cannot overtake any other node that it sees. Therefore, if all nodes see their nearest neighbors, the application of Equation (7) and (8) always produces the same dynamic A_1 and the system evolves with a linear and time–invariant dynamics. In the general case, the number S of linear dynamics is upper bounded by the number of pairs of nodes that do not see each other.

In the rest of the section, we will first study the stability properties of each linear dynamic system A_i. Then we will extend our analysis to the global stability properties for some specific topologies. For the sake of brevity, we will not focus on the case $N > 2$, since for $N = 2$ the study of the behavior of the system is straightforward.

4 Stability Analysis

4.1 Local analysis

As discussed above, the evolution of the system is generally described by a linear switching system. Our first task is to study the evolution of the system in each of its linear dynamics.

Consider the set $\mathcal{S}_E^{N_x} = \{\mathbf{x} \in \mathbb{R}^{N_x} | 0 \leq x_i \leq E\}$. The stability of each of the linear dynamics of the system is showed in the subsequent Lemma.

Lemma 1. *Given the system* $\mathbf{x}^{k+1} = A\mathbf{x}^k$ *and* $\mathbf{x}^0 \in \mathcal{S}_E^{N_x}$ *the following statements hold true:*

– $\mathbf{x}^k \in \mathcal{S}_E^{N_x}\ \forall k > 0$;

– *the system is stable;*
– *the equilibrium points* $\bar{\mathbf{x}}$ *belong to a linear subspace defined by the* $m \geq 1$ *eigenvectors* $\mathbf{v}_i$ *associated to the* m *eigenvalues* $\lambda_i = 1$.

Proof. By construction, each element of the state vector $\mathbf{x}$ is a distance between two nodes that see each other, therefore, by Theorem 1, the nodes do not overtake each other during an update – i.e. the set $\mathcal{S}_E^{N_x}$ is invariant for A. Since the distances $\mathbf{x}^0$ are positive by hypothesis, the first statement holds.

The diagonal elements of the matrix A are equal to $1 - l\alpha/2$, where l is an integer number in the set $\{0, \ldots, 4\}$. Indeed, each element $\overrightarrow{d}_{i,j} \in \mathbf{x}$ may appear up to four times in the update equations (7) and (8). Applying the Gerschgorin principle to each row and recalling that $0 < \alpha < 1$, it follows that the eigenvalues associated to $l = 4$ and $l = 3$ are inside the unit circle, while for $l = 2$ we have at most an eigenvalue $\lambda = 1$. The case of $l = 1$ is more challenging, since the Gerschgorin principle cannot be applied to demonstrate convergence, although it states that $\mathrm{Re}(\lambda_i) > -1$. However, since each element of the state vector $\mathbf{x}$ is a distance between two nodes that see each other, Theorem 1 holds, so it is not possible to have expansive dynamics, i.e. $\max_i |\lambda_i| = 1$. Finally, since cancellations are not possible, the presence of 1 on the diagonal of A ($l = 0$) means that the associated distance does not contribute to its dynamic and to any other dynamic, i.e. if the distance is $\overrightarrow{d}_{i,j}$, there is a node closer to i than j and vice-versa. Therefore, applying the Gerschgorin principle to the column results, again, in an eigenvalue $\lambda_i = 1$. Notice that if m eigenvalues $\lambda_i = 1$, they must be simple, i.e. associated to distinct eigenvectors $\mathbf{v}_i$, since, again, no expansive dynamics are allowed.

To show the stability of the system to a point $\bar{\mathbf{x}} = \sum_{i=1}^{m} \beta_i \mathbf{v}_i$, it is sufficient to prove that there is not a persistent oscillation in the system. Trivially, oscillating modes exists if there will be one or more simple eigenvalues $\lambda_i = -1$, excluded by the aforementioned Gerschgorin analysis, or in the presence of complex eigenvalues. Since complex eigenvalues do not exist (indeed $\mathbf{x}^k \geq 0 \ \forall k > 0$), the Lemma is proved.

As an immediate consequence of the above, if we perturb the system state from an equilibrium point that is not on a switching surface, i.e., a surface delimited by constraints of the type $\overrightarrow{d}_{i,j} = \overrightarrow{d}_{i,l}$, $l, j \in \mathcal{V}_i$ and $\forall i = 1, \ldots, N$, the system will recover its equilibrium. In plain words, the wake-up scattering algorithm converges if we initialize the vector of wake-up times with an initial value close to a fixed point and far enough from a switching surface. Unfortunately, since the property is local, we are not able to easily quantify the maximum amount of the allowed perturbation.

4.2 Global analyis

The result described above does not *per se* ensure convergence starting from a general initial condition. However, there are some interesting topologies for which such global “provisions” can indeed be given.

Visibility of the nearest neighbors. In case of visibility of the nearest neighbors, the topology of the system structurally prevents any switch. Therefore the local stability results that we stated above for each linear dynamic have, in this case, global validity. In other words, the wake-up scattering converges from any positive initial assignment for the wake-up times.

The complete visibility topology, i.e., $\mathcal{V}_i = j, \forall i, j = 1, \dots, N, \forall j \neq i$, and the cyclic topology have been explicitly considered in the literature on the consensus problem ([8,11]). Indeed, choosing only the entries of $\mathbf{x}$ equal to $\min_{l \in \mathcal{V}_i}(\overrightarrow{d}_{i,l})$ for $i = 1, \dots, N$, hence $\mathbf{x} \in \mathbb{R}^N$, the discrete time system (9) is again autonomous and its dynamic matrix A turns out to be doubly stochastic and circulant ([12]). Therefore, it is possible to determine the closed form of its eigenvalues and the equilibrium point ([11]), the rate of convergence with respect to the number of nodes ([15]) and the network communication constraints to accomplish the desired task ([5]). We will not consider this case any further (a numeric example is presented below).

Removing one link to the nearest neighbor. For this case, we start from a complete graph and remove one link to the nearest neighbor. In this case, we end up with two nodes, say j and p, that do not see each other and are the nearest ones to each other. This situation is depicted in Figure 1.(A). Consider the two update laws with respect to node i (nearest neighbor to j):

$$\begin{aligned} \overrightarrow{d}_{i,p}^{k+1} &= \overrightarrow{d}_{i,p}^{k} + \frac{\alpha}{2}(\overrightarrow{d}_{p,f}^{k} - \overrightarrow{d}_{i,j}^{k}) - \frac{\alpha}{2}(\overrightarrow{d}_{i,p}^{k} - \overrightarrow{d}_{z,i}^{k}) \\ \overrightarrow{d}_{i,j}^{k+1} &= \overrightarrow{d}_{i,j}^{k} + \frac{\alpha}{2}(\overrightarrow{d}_{j,f}^{k} - \overrightarrow{d}_{i,j}^{k}) - \frac{\alpha}{2}(\overrightarrow{d}_{i,j}^{k} - \overrightarrow{d}_{z,i}^{k}) \end{aligned}$$

The update law of the distance $\overrightarrow{d}_{j,p}^{k} = \overrightarrow{d}_{i,p}^{k} - \overrightarrow{d}_{i,j}^{k}$ is

$$\overrightarrow{d}_{j,p}^{k+1} = \overrightarrow{d}_{i,p}^{k+1} - \overrightarrow{d}_{i,j}^{k+1} = \left(1 - \frac{\alpha}{2}\right) \overrightarrow{d}_{j,p}^{k} + \frac{\alpha}{2}(\overrightarrow{d}_{p,f}^{k} - \overrightarrow{d}_{j,f}^{k}) = (1-\alpha)\, \overrightarrow{d}_{j,p}^{k}.$$

Thereby, even though the nodes do not see each other, they do not overtake each other and will, eventually, occupy the same time position. Hence, the switching never happens and the system will converge as in the complete visibility topology with $N-1$ nodes.

Removing four links to the nearest neighbor.

Consider a more involved condition, in which two pairs of nodes do not see their nearest neighbor, as in Figure 1.(B). Consider the distances of $\overrightarrow{d}_{j,p}^{k} = \overrightarrow{d}_{i,p}^{k} - \overrightarrow{d}_{i,j}^{k}$ and $\overrightarrow{d}_{l,f}^{k} = \overrightarrow{d}_{l,w}^{k} - \overrightarrow{d}_{f,w}^{k}$. Noticing that $\overrightarrow{d}_{p,f}^{k} - \overrightarrow{d}_{j,l}^{k} = \overrightarrow{d}_{l,f}^{k} - \overrightarrow{d}_{j,p}^{k}$, one gets

$$\begin{aligned} \overrightarrow{d}_{j,p}^{k+1} &= (1-\alpha)\, \overrightarrow{d}_{j,p}^{k} + \frac{\alpha}{2} \overrightarrow{d}_{l,f}^{k} \\ \overrightarrow{d}_{l,f}^{k+1} &= (1-\alpha)\, \overrightarrow{d}_{l,f}^{k} + \frac{\alpha}{2} \overrightarrow{d}_{j,p}^{k}, \end{aligned}$$

that is a linear system with a pair of real eigenvalues: $\lambda_1 = 1 - \alpha/2$ and $\lambda_2 = 1 - 3\alpha/2$. We can distinguish two case: 1) $\lambda_1 > 0$, $\lambda_2 > 0$, 2) $\lambda_1 > 0$ or $\lambda_2 < 0$.

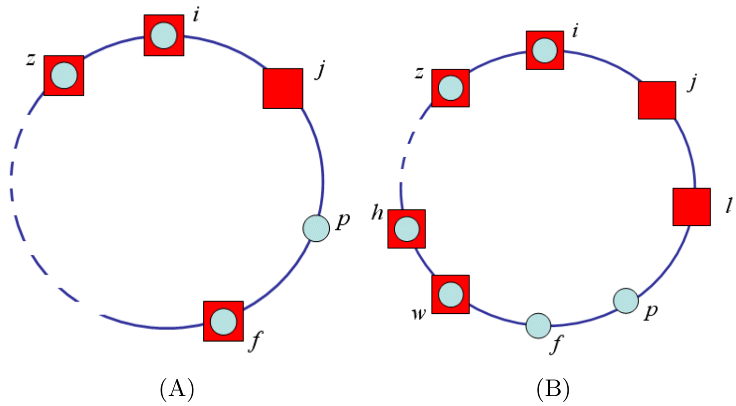


Fig. 1. Visibility and proximity of the partial visibility topology. Nearest neighbors are the closest nodes in clockwise and counter–clockwise direction, i.e. the nearest neighbor to l are p and j. Visibility is depicted with squares and circles respectively. (A) $p \notin \mathcal{V}_j$ and $j \notin \mathcal{V}_p$. (B) $p, f \notin \mathcal{V}_j$, $p, f \notin \mathcal{V}_l$, $j, l \notin \mathcal{V}_p$, $j, l \notin \mathcal{V}_f$.

In the first case (corresponding to $0 < \alpha \leq 2/3$), we can easily see that for each initial condition there is a maximum time beyond which switchings no longer occur. Then, recalling Lemma **??**, global stability is ensured. In the second case, we cannot rule out an infinite number of switchings determined by the oscillating behavior of the power of the negative real eigenvalue. Even so, we get $\overrightarrow{d}^{\,k}_{j,p} \to 0$ and $\overrightarrow{d}^{\,k}_{l,f} \to 0$. As a result the two pair of nodes will eventually behave as a single node with complete visibility and the network will stabilize on the equilibrium point a network with complete visibility would have with $N - 2$ nodes.

The same approach can also be used if, for example, node f is removed from the network, which implies that the symmetry among the nodes with partial neighbor visibility is no longer valid. From the previous analysis it follows that

$$\overrightarrow{d}^{\,k+1}_{j,p} = (1 - \alpha)\, \overrightarrow{d}^{\,k}_{j,p} + \frac{\alpha}{2} \overrightarrow{d}^{\,k}_{l,w}.$$

Since $\overrightarrow{d}^{\,k}_{l,w} > 0$ for $k > 0$, $\overrightarrow{d}_{j,p} \to \overrightarrow{d}_{l,w}/2 > 0$. The same statement holds also for $\overrightarrow{d}^{\,k}_{p,l}$, hence node p will converge to the midpoint between j and l. The number of switching is then limited in time also in this case.

5 Numerical examples

In this section, we provide some numerical evidence of the effectiveness of the approach. The section is composed of two parts. In the first part, we show the convergence properties of the algorithm in a simple example. In the second one, we show how the algorithm converges to a schedule achieving a good coverage of the area the WSN is deployed on.

Fig. 2. Visibility Graph for the first scenario considered in Section 5.1

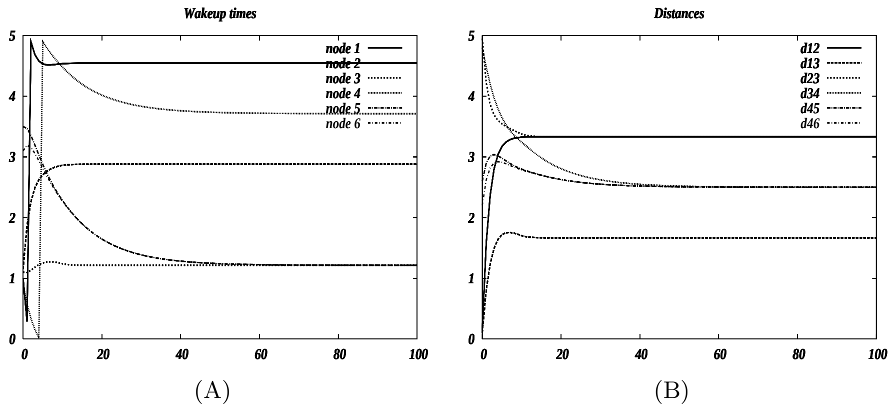


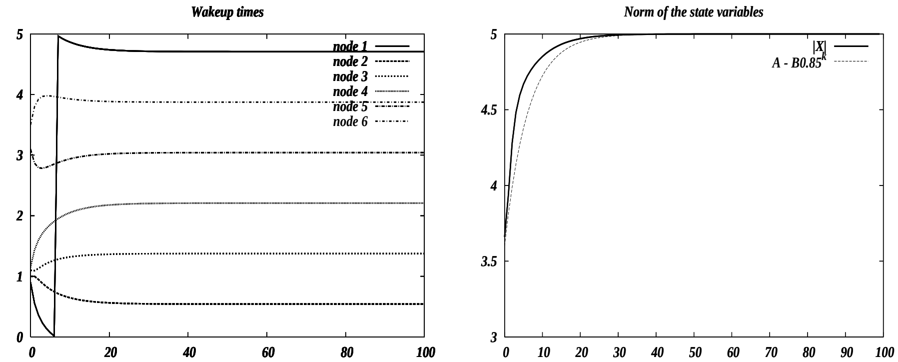
Fig. 3. Convergence in the case of partial visibility. (A) wake-up times, (B) state variables

5.1 Convergence properties

For this set of simulations, we consider a set of 6 nodes in two different scenarios: complete visibility and partial visibility. In the latter case, we assume the visibility graph shown in Figure 2. In both cases we consider a duration for the epoch (i.e., the period used for the schedule) equal to 5 and we set initial wake-up times to the value $w(0) = [1, 1.15, 1.1, 0.9, 3.5, 3.1]$ and the parameter $\alpha = 0.3$.

For partial visibility, the application of the wake-up scattering algorithm yields the evolution of the wake-up times depicted in Figure 3.(A). As it is possible to see, some of the nodes "overtake" each other. However, as we discussed above, if we study the dynamics of the distances between the wake-up times of the nodes that see each other (which can be considered as state variables), we deal with a convergent linear dynamic as shown in Figure 3.(B).

In the case of complete visibility, as discussed above we can make much stronger claims than simple convergence. Indeed, not only are we able to conclude that the wake-up times of the nodes will be evenly spaced out (in the steady state), but we can also compute the rate of convergence. In Figure 4.(A), we report the evolution of the wake-up times. As we discussed in Section 4, to properly describe the dynamics of the system, it is appropriate to rename the



(A) (B)

Fig. 4. Convergence in the case of total visibility. (A) wake-up times, (B) state variables

nodes so that their initial wake-up times are ordered in increasing order. After this renaming a convenient choice of state variables is:

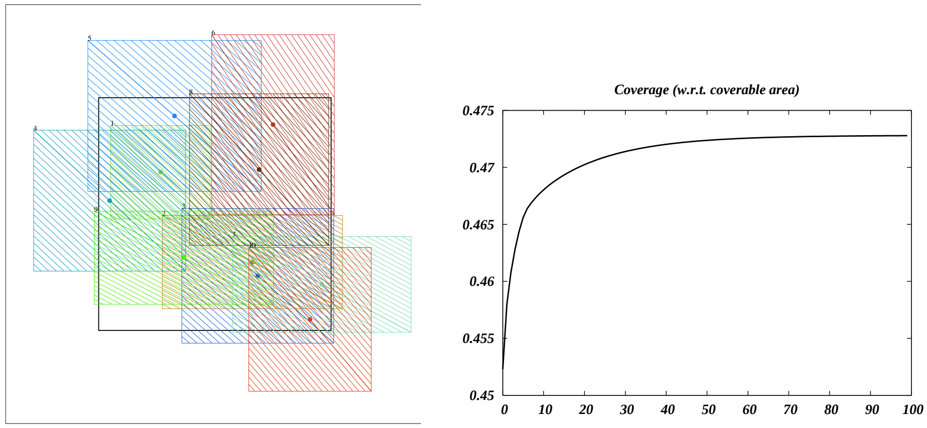
$$X = \begin{bmatrix} \overrightarrow{d}_{1,2} - \overrightarrow{d}_{2,3} \\ \overrightarrow{d}_{2,3} - \overrightarrow{d}_{3,4} \\ \cdots \\ \overrightarrow{d}_{5,6} - \overrightarrow{d}_{6,1} \\ \overrightarrow{d}_{1,2} + \overrightarrow{d}_{2,3} + \overrightarrow{d}_{3,4} + \overrightarrow{d}_{4,5} + \overrightarrow{d}_{5,6} + \overrightarrow{d}_{6,1} \end{bmatrix}$$

As discussed above, this vector provably converges to $[0, 0, \ldots, E]$ with a rate dictated by the second eigenvalue. Its dynamic matrix is a circulant matrix and its eigenvalues can be computed in closed form. The largest eigenvalue is 1 and the second one is 0.85 (see [11]) . Therefore, the convergence decay rate is $\approx 0.85^k$, as shown in Figure 4.(B).

5.2 Coverage properties

In order to show the performance of the wake-up scattering algorithm for the coverage problem, we consider a very simple deployment consisting of 10 nodes. For the sake of simplicity and without loss of generality, we consider a rectangular sensing range for the nodes. The nodes are randomly distributed over a 500×500 bi-dimensional area. The resulting deployment is shown in Figure 5.(A). We consider a period for the schedule equal to 5 time units and a wake-up interval for the nodes equal to 1. Therefore, each node is awake for 20% of the total time.

Several regions of the considered arena are covered by multiple nodes. Therefore, a good schedule is one where the wake-up times of nodes sharing "large" areas are far apart. Using the algorithm presented in [16], we come up with an optimal schedule, where an average of 52.94% of the "coverable" area (i.e., the area actually within the sensing range of the nodes) is actually covered. The application of the wake-up scattering algorithm over 100 iterations, assuming complete



(A) (B)

Fig. 5. The coverage scenario. (A) Spatial distribution of the nodes, (B) Evolution of the ratio between covered area and coverable area.

visibility between the nodes, produces the result shown in Figure 5.(B). The attained relative coverage is 47.3%. The deviation from the optimal solution is in this case lower than 10% of the optimal coverage. The result is particularly interesting because the coverage problem is known to be exponential, while the wake-up scattering algorithm operates in polynomial time and is entirely distributed.

6 Conclusions

In this paper, we have presented convergence results of a distributed algorithm used for maximizing the lifetime of a WSN. We have focused our attention on an algorithm recently proposed in the literature, showing how its convergence can be cast into a stability problem for a linear switching system. We have found local stability results in the general case, and global stability results for specific topologies of the WSN.

Several issues have been left open and will offer interesting opportunities for future research. The first obvious point to address is to study global stability for general topologies. Another point is to study conditions under which the wake-up scattering algorithm produces a good coverage, developing improvements for the cases in which the result is not satisfactory. From a practical view–point, the scattering algorithm here presented reduces its performance if node clock synchronization is not guaranteed or in the presence of communication delays. Future analysis will consider algorithm convergence also in the presence of such random nuisances.

References

1. A. Alfieri, A. Bianco, P. Brandimarte, and C. F. Chiasserini. Maximizing system lifetime in wireless sensor networks. *European Journal of Operational Research*, 127(1):390–402, August 2007.
2. B. Cărbunar, A. Grama, J. Vitek, and O. Cărbunar. Redundancy and coverage detection in sensor networks. *ACM Transaction on Sensor Networks*, 2(1):94–128, February 2006.
3. Q. Cao, T. Abdelzaher, T. He, and J. Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *Proc. of the 4th Int. Symp. on Information Processing in Sensor Networks (IPSN)*, April 2005.
4. M. Cardei, M. Thai, Y. Li, and W. Wu. Energy-efficient target coverage in wireless sensor networks. In *Proc. of INFOCOM*, 2005.
5. R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri. Communication constraints in the average consensus problem. *Automatica*, 44(3):671–684, 2008.
6. A. Giusti, A.L. Murphy, and G.P. Picco. Decentralized Scattering of Wake-up Times in Wireless Sensor Networks. In *Proc. of the 4th European Conf. on Wireless Sensor Networks (EWSN)*, LNCS 4373, pages 245–260. Springer, January 2007.
7. C. Hsin and M. Liu. Network coverage using low duty-cycled sensors: Random & coordinated sleep algorithms. In *Proc. of the 3th Int. Symp. on Information Processing in Sensor Networks (IPSN)*, 2004.
8. A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. on Automatic Control*, 48(6):988–1001, 2003.
9. H. Liu, X. Jia, P. Wan, C. Yi, S. Makki, and N. Pissinou. Maximizing lifetime of sensor surveillance systems. *IEEE/ACM Trans. on Networking*, 15(2):334–345, 2007.
10. J.A. Marshall, M.E. Broucke, and B.A. Francis. Formations of vehicles in cyclic pursuit. *IEEE Trans. on Automatic Control*, 49(11):1963–1974, 2004.
11. S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, 2006.
12. S. Martinez, F. Bullo, J. Cortes, and E. Frazzoli. On synchronous robotic networks - Part I: Models, tasks and complexity. *IEEE Trans. on Automatic Control*, 52(12):2199–2213, 2007.
13. S. Martinez, F. Bullo, J. Cortes, and E. Frazzoli. On synchronous robotic networks - Part II: Time complexity of rendezvous and deployment algorithms. *IEEE Trans. on Automatic Control*, 52(12):2214–2226, 2007.
14. Carl D. Meyer, editor. *Matrix analysis and applied linear algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.
15. R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi–agent systems. *Proc. of IEEE*, 95(1):215–233, January 2007.
16. Luigi Palopoli, Roberto Passerone, Gian Pietro Picco, Amy L. Murphy, and Alessandro Giusti. Maximizing sensing coverage in wireless sensor networks through optimal scattering of wake-up times. Technical Report DIT-07-048, Dipartimento di Informatica e Telecomunicazioni, University of Trento, July 2007.
17. S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *Proc. of the IEEE Int. Conf. on Communications (ICC)*, June 2001.
18. D. Tian and N.D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *First ACM Int. Wkshp. on Wireless Sensor networks and Applications (WSNA)*, 2002.

19. F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. PEAS: A robust energy conserving protocol foo long-lived sensor networks. In 3^{rd} *Int. Conf. on Distributed Computing Systems (ICDCS'03)*, May 2003.