ARTICLE IN PRESS
Ad Hoc Networks xxx (2013) xxx–xxx
Contents lists available at ScienceDirect
Ad Hoc Networks
journal homepage: www.elsevier.com/locate/adhoc

Efficient distributed lifetime optimization algorithm for sensor networks

Jacques Bahi [a], Mohammed Haddad [b,c], Mourad Hakem [a,*], Hamamache Kheddouci [b,c]

[a] DISC Laboratory, Institut FEMTO-ST, UMR CNRS, Université de Franche-Comté, France
[b] LIRIS Laboratory, UMR CNRS, Université de Lyon, F-69003, France
[c] Université Claude Bernard Lyon 1, F-69622, France

ARTICLE INFO

Article history:
Received 12 November 2012
Received in revised form 1 June 2013
Accepted 20 November 2013
Available online xxxx

Keywords:
Lifetime optimization
Sensor networks
Reliability
Self-stabilizing algorithm
Weibull distribution

ABSTRACT

One of the main design challenges in Wireless Sensor Networks (WSN) is to prolong the system lifetime, while achieving acceptable quality of service for applications. In WSN, each sensor node is battery powered and it is not convenient to recharge or replace the batteries in many cases, especially in remote and hostile environments. Due to the limited capabilities of sensor nodes, it is usually desirable that a WSN should be deployed with high density and thus redundancy can be exploited to increase the network's lifetime. In this paper, we introduce an efficient lifetime optimization and self-stabilizing algorithm to enhance the lifetime of wireless sensor networks especially when the reliabilities of sensor nodes are expected to decrease due to use and wear-out effects. Our algorithm seeks to build resiliency by maintaining a necessary set of working nodes and replacing failed ones when needed. We provide some theoretical and simulation results, that fully demonstrate the usefulness of the proposed algorithm.

1. Introduction

Recent years have witnessed significant advances in wireless sensor networks which emerge as one of the most promising technologies for the 21st century [2]. In fact, they present huge potential in several domains ranging from health care applications to military applications. However, the advent of Wireless Sensor Networks (WSN) and its conceptual constraints, have posed a number of research challenges to the networking and distributed computation communities. A problem that has received renewed interest is **lifetime optimization**. Indeed, as the scale is expanding, node failures are more likely to occur and can have an adverse effect on the application. In addition, as sensor nodes have limited resources in terms of memory, energy and computational power, they suffer from increased susceptibility to wear-out effect [23,25,26].

In WSN, if the wear-out failures are not taken into consideration during the execution of the involved application, some nodes may age much faster than the others and become the reliability bottleneck for the network, thus significantly reducing the system's service life. In the literature, this problem has been formulated and studied in various ways. For instance, prior work [14,30,36,37] in lifetime reliability assumes node's failure rates to be independent of their usage times. While this assumption can be accepted for memoryless soft failures, it is obviously inaccurate for the wear-out-related fail-silent (a faulty node does not produce any output) and fail-stop (no node recovery) failures, because the sensor node's lifetime reliability will gradually decrease over time.

Since sensor nodes have limited battery life and without being able to replace batteries, especially in remote and hostile environments, it is desirable that a WSN should be deployed with high density and thus redundancy can be

* Corresponding author.
E-mail addresses: Jacques.Bahi@univ-fcomte.fr (J. Bahi), Mohammed.Haddad@univ-lyon1.fr (M. Haddad), Mourad.Hakem@univ-fcomte.fr (M. Hakem), Hamamache.Kheddouci@univ-lyon1.fr (H. Kheddouci).

1570-8705/$ - see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.adhoc.2013.11.010

Please cite this article in press as: J. Bahi et al., Efficient distributed lifetime optimization algorithm for sensor networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.11.010

exploited to increase the lifetime of the network. In such a high density network, if all sensor nodes were to be activated at the same time, the lifetime would be reduced. Consequently, future software may need to adapt appropriately. This paper presents a novel wear-out-aware lifetime optimization and self-stabilizing algorithm to improve system's lifetime and reliability characteristics of such a non-deterministic architecture. In this approach, some nodes are scheduled to sleep while the remaining working nodes provide continuous service. The objective, is to minimize the number of nodes that remain active while still achieving acceptable quality of service for applications.

Self-stabilizing algorithms [5,16,21] can start execution from arbitrary (illegitimate) configuration and eventually configuration becomes legitimate. By this property, self-stabilizing algorithms tolerate any kind and any finite number of transient faults. In a self-stabilizing model [12], each node has only a partial view of the system, called the *local state*. The node's local state include the state of the vertex itself and the state of its neighborhood. The union of the local states of all the nodes gives the *global state* of the system. Based on its local state, a node can decide to make *a move*. Then, self-stabilizing algorithms are given as a set of rules of the form [**If** $p(i)$ **Then** M], where $p(i)$ is a predicate and M is a move. $p(i)$ is true when state of the node i is locally illegitimate. In this case, the vertex i is called a *privileged/active* vertex. A vertex executes the algorithm as long as it is active (at least one predicate is true). Self-stabilizing algorithms can be designed according to different daemons, also called schedulers. There are two kinds of daemons which are often assumed in the literature of self-stabilizing algorithms: Central and Distributed ones. If the scheduler is central, then at each step, the only one privileged node is arbitrarily selected to make its move. If the scheduler is distributed, then at each step, a non-empty set of privileged nodes is selected to make their moves simultaneously. We assume the distributed model in which each vertex makes its decision independently, so more than one privileged vertex may be selected at the same time by the scheduler. We also assume each sensor node in a network has unique identifier.

The rest of the paper is organized as follows. In Section 2, we present a brief survey of heuristics proposed in the literature to optimize sensor network's lifetime. After some definitions and notations in Section 3, we present in Sections 4, 5 and 7 the design and analysis of the proposed algorithm. Finally, we give some concluding remarks in Section 8.

2. Related work

Most papers on lifetime optimization deal with centralized control over sensor networks which aim at maximizing the lifetime of sensor networks. For instance [29] introduces a heuristic that selects mutually exclusive sets of sensor nodes, where the members of each of those sets together completely cover the monitored area. The intervals of activity are the same for all sets, and only one of these sets is active at any time. [9] presents a graph coloring technique to achieve energy savings by organizing the sensor nodes into a maximum number of disjoint dominating sets (DDS) which are activated successively. The dominating sets do not guarantee the coverage of the whole area.

In [8], the authors propose a heuristic to compute the disjoint set covers (DSC). In order to compute the maximum number of covers, they first transform DSC into a maximum-flow problem (MFP), which is then formulated as a mixed integer programming problem (MIP). Based on the solution of the MIP, they design a heuristic to compute the final number of covers. The results show a slight performance improvement in terms of the number of produced DSC in comparison to [29] but it incurs higher time complexity. Abrams et al. [1] design three approximation algorithms for a variation of the set k-cover problem, where the objective is to partition the sensors into covers such that the number of covers that include an area, summed over all areas, is maximized. Their work builds upon previous work in [29] and the generated cover sets do not provide complete coverage of the monitored areas.

Cardei et al. [10] presents a Linear Programming (LP) solution and a greedy approach to extend the sensor network life time by organizing the sensors into a maximal number of non-disjoint cover sets, *i.e.*, nodes may participate in more than one cover sets. Simulation results show that by allowing sensors to participate in multiple sets, the network lifetime increases compared with related work [8]. In [6], the authors have formulated the lifetime problem and suggested another Linear Programming (LP) technique to solve this problem. A centralized provably near optimal solution based on the Garg–Konemann algorithm [15] is also proposed.

Zorbas et al. [38] present B{GOP}, a centralized coverage algorithm introducing sensor candidate categorization depending on their coverage status and the notion of critical target to call targets that are associated with a small number of sensors. The total running time of their heuristic is $O(mn^2)$ where n is the number of sensors, and m the number of targets. Compared to algorithm's results of Slijepcevic and Potkonjak [29], their heuristic produces more cover sets with a slight growth rate in execution time.

In the case of non-disjoint algorithms [11], sensors may participate in more than one cover set. In some cases this may prolong the lifetime of the network in comparison to the disjoint cover set algorithms but designing algorithms for non-disjoint cover sets generally incurs a higher order of complexity.

All the centralized solutions presented above are developed for maximizing network's lifetime by organizing the sensor nodes into a suitable number of disjoint or nondisjoint set covers that are activated periodically. However, they do not scale better to accommodate larger networks and they do not achieve fault tolerance, *i.e.*, it is assumed that nodes belonging to the same cover set will never fail or become unavailable during the set cover's round.

In order to hide the occurrence of faults, or the sudden unavailability of sensor nodes, some distributed algorithms have been developed in [14,30,36,37,20,7]. The scheduling information is disseminated throughout the network and only sensors in the active state are responsible for monitoring all targets, while all other nodes are in a low-energy sleep mode. The nodes decide cooperatively which of them will remain in sleep mode for a certain period of time.

The authors in [24] leverage prediction to prolong the network life time, by exploiting temporal–spatial correlations among the data sensed by different sensor nodes. Based on Gaussian Process, the authors formulate the issue as a minimum weight submodular set cover problem and propose a centralized and a distributed truncated greedy algorithms (TGA and DTGA). They prove that these algorithms obtain the same set cover.

Lifetime optimization using knowledge about the dynamics of stochastic events has been studied in [17]. They present the interactions between periodic scheduling and coordinated sleep for both synchronous ad asynchronous dense static sensor network. They show that the event dynamics can be exploited for significant energy savings, by putting the sensors on a periodic on/off schedule. This work is based on the fact that leaving an area uncovered some of the time may be acceptable, since an event arriving when there is no active sensor may stay long enough until a sensor becomes active.

In [18], the authors design a polynomial-time, distributed algorithm for maximizing the lifetime of the network. They proved that the lifetime attained by their algorithm approximates the maximum possible lifetime within a logarithmic approximation factor. The proposed algorithm seek to maximize the lifetime of sensor networks by activating sensors based on their residual energy content.

In [19], a general coverage algorithm, which also considers the network connectivity is presented. The proposed protocol, called Probabilistic Coverage Protocol (PCP), works for the common disk sensing model as well as probabilistic sensing model. To support probabilistic sensing models, the authors introduce the notion of probabilistic coverage of a target area with a given threshold θ, which means that an area is considered covered if the probability of sensing an event occurring at any point in the area is at least θ. They prove the correctness of the protocol and provide bounds on its convergence time and message complexity.

Although these algorithms enable applications to achieve very good performance, they suffer from a common drawback, namely their reliance on the assumption that the reliabilities of sensor nodes are not expected to decrease over time. In addition, no guarantee has been proposed to ensure the fact that only one sensor node must be in the active state for each monitoring zone. This leads to situations where two neighboring sensor nodes may be elected at the same time step, and decision of two neighboring nodes may be the same. In this paper, we present an efficient self-stabilizing algorithm to tackle the problem of lifetime optimization in large-scale sensor networks. This study differs from previous works for the following reasons:

- We use a *Weibull distribution* [3,27,28] to take into account wear-out failures and therefore to increase network's lifetime.
- We express an Upper-Bound of the actual number of *probe/reply* messages exchanged during the network's lifetime optimization task.
- We orchestrate the tradeoffs relation between sensor network's reliability and the required number of cover sets, i.e., we would like to determine, for a fixed reliability threshold, what is the minimum number of cover sets that can be used in the system's coverage while achieving the prescribed reliability of the sensor network?
- We provide provable guarantees for the election process of working nodes for each monitoring zone.
- Unlike earlier methods, we use a new concept of Self-Stabilization to achieve significant energy savings.

3. Fundamentals and sensing model

We model the topology of a sensor network by an undirected graph - the communication graph $G = (V, E)$. Let V be the set of nodes (the set of vertices), and E the links between nodes (the set of edges). The nodes are labeled $i = 1, 2, \ldots, n$, and a link between nodes i and j is denoted by (i,j). The set of neighbors of node i is denoted by $N_i = \{j \in V | (i,j) \in E\}$, and the degree (number of neighbors) of node i by $\eta_i = |N_i|$.

We denote by R_s and R_c the sensing range and the radio transmission range of a sensor node, respectively. We assume that sensor nodes are deployed arbitrarily in an area (region) of interest. Similar to [33,34,37], the entire region is divided into square grids of side length ω (see below for the choice of ω) and one sensor node is selected to be awake in each grid. The maximum distance between any two pairs of sensor nodes in adjacent grids is within the transmission range of each other. The sensing and connectivity models we adopt are as in [31,32]. It was proved in [31,32] that if $R_c \geqslant 2 \cdot R_s$, then *coverage* implies *connectivity*. We deduce that in order to maintain coverage, the grid size ω must be chosen as:

$$\omega \leqslant \frac{R_s}{\sqrt{2}} \equiv \omega \leqslant \frac{R_c}{2 \cdot \sqrt{2}}$$

Thus for a large area with size $l \times l$, it requires $\frac{2l^2}{R_s^2}$ sensor nodes to operate in the active state to ensure complete coverage. Nevertheless, in our approach the subdivision of the monitored region into square grids is only performed to get a better evaluation of the coverage performance. In fact, our proposed algorithm can operate on any arbitrary network topology.

4. The proposed algorithm

The goal of the algorithm that we propose here is to maintain a critical level of performance, as nodes wear out and degrade, by keeping a minimum number of sensor nodes in the active mode in WSN. We suppose that we are in the case of high density networks, and not all nodes participate in the network functioning. Some nodes are in an idle state because their coverage zones are actually covered by active nodes. We consider that these idle sensors wakeup periodically to check for eventual node failures and therefore ensure their zones coverage. In case of failures, they decide to pass to active mode and participate in the network's service. However, two questions arise here:

(1) How is the fault detection done?
(2) And how to replace the failed sensors?

Intuitively, when a sleeping node wakes up, it sends a probe request message to check if there exist working nodes in its vicinity. If no working nodes, it starts to operate in the active mode; otherwise, it sleeps again, but the questions (1) and (2) stated above still raise the following problems:

Problem 1. Since sensor nodes are not aware of their neighbors, especially the number of sleeping/passive nodes. How to adjust the wakeup period of these sensors?

Problem 2. During the network's service, how to handle the case where two or more sleeping nodes, would realize at the same time that the working/active node is down?

4.1. Problem I – resolution

To compute node's sleeping wakeup rate, we adopt a Weibull distribution [3,27,28] to mitigate wear-out failures and therefore to increase network's lifetime.

In the following we give a brief description and analysis of the Weibull distribution we use to derive reliable scheme for lifetime optimization in WSN.

4.1.1. Future lifetimes distribution analysis

The Weibull distribution is a continuous distribution. It has been widely used to model the hazard function for many types of equipments/objects especially in the reliability field [28]. The Weibull distribution's popularity results from its flexibility and its ability to provide a reasonably accurate failure analysis and failure forecasts. The Weibull Probability Density Function (PDF) is

$$f(t) = \begin{cases} \frac{\beta(t-\gamma)^{\beta-1}}{\alpha^\beta} e^{-\left(\frac{t-\gamma}{\alpha}\right)^\beta} & t,\alpha,\beta,\gamma > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha > 0$ is the scale parameter, $\beta > 0$ is the shape parameter (it dictates the behavior of the failure rate function) and γ is the location parameter. Note that, when the distribution starts at $t = 0$, then $\gamma = 0$. Thus, the Weibull probability density function reduces to $f(t) = \frac{\beta t^{\beta-1}}{\alpha^\beta} e^{-\left(\frac{t}{\alpha}\right)^\beta}$ for t, α, $\beta > 0$, and its Cumulative Distribution Function (CDF) is $F(t) = \int_0^t f(t) = 1 - e^{-\left(\frac{t}{\alpha}\right)^\beta}$.

Now, suppose that sensor nodes availability lifetimes are represented as a random variable X with probability distribution F, and let t be a non-negative real number. The probability that a node will fail within the next x *time units* given that it has been available for t *time units* is defined as follows:

$F_t(x) = F_{X \geqslant t}(t + x) = \frac{F(t+x)-F(t)}{1-F(t)}, t \geqslant 0$

From this function, which expresses the node's future lifetimes distribution beyond t, it is clear that if nodes availability lifetimes follow an exponential distribution, the amount of time a node has already been available has no impact on how long it is likely to remain available. Formally:

$$F_t(x) = F[X > t + x | X > t] = \frac{F[\{X > t + x\} \cap \{X > t\}]}{F[X > t]}$$
$$= \frac{F[X > t + x]}{F[X > t]} = \frac{e^{-\lambda(t+x)}}{e^{-\lambda t}} = e^{-\lambda x} = F[X > x]$$

For this reason, the exponential distribution is called *memoryless*. Thus, an exponential distribution is not appropriate for wear-out modeling since wear-out is not a *memoryless* phenomenon.

The future lifetime distribution for a Weibull reduces to

$$F_t(x) = e^{-\left(\left(\frac{t+x}{\alpha}\right)^\beta - \left(\frac{x}{\alpha}\right)^\beta\right)}$$

This function clearly depends on t as well as x when $\beta \neq 1$. When $0 < \beta < 1$, the probability that a sensor node will survive another *time unit* increases as t increases. For $\beta > 1$, this probability decreases, and when $\beta = 1$ the distribution reduces to an exponential and therefore *memoryless*. Thus, a Weibull distribution is capable of modeling different aging effects, depending on its shape parameter.

4.1.2. Node's wakeup rate

Intuitively, nodes are initially in the sleeping mode. Each node sleeps for a randomized amount of time generated according to a Weibull Probability Density Function (PDF): $f(t) = \frac{\beta t^{\beta-1}}{\alpha^\beta} e^{-\left(\frac{t}{\alpha}\right)^\beta}$, where $\frac{1}{\alpha} = \lambda$ is the probing rate of the sensor node and t denotes its sleeping time duration. We define a parameter α^* where $\frac{1}{\alpha^*}$ value is used to define the initial average sleeping duration for each passive sensor and therefore the aggregate rate of *probe* messages desired by the application. The higher the frequency is, the higher eventual failures will be detected/handled quickly, but this leads to the increase of the number of involved messages. To better adjust the probing rate, we must take into account the nature of the application. For example, if we have an application responsible for making temperature measurements every 1 hour, it is clear that setting α^* will not be the same for an application whose function is to detect a fire. As the number of sleeping nodes (passive modes) is not known by other working nodes, the initial value of α for each passive node is set to α^*.

To fine tune node's sleeping wakeup rate, and unlike to what has been presented in [36], the parameter $\lambda = \frac{1}{\alpha}$ is adjusted according to the following lemma:

Lemma 1. *A node's wakeup rate is monotonically increasing.*

Proof. the probing rate is the conditional wakeup probability of a node during the interval t and $t + \Delta t$, given it was in a sleeping state at time t. We denote it $\lambda(t)$, a function of time t, named probing rate function:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} P(t < T \leqslant t + \Delta t | T > t) = \frac{f(t)}{1 - F(t)}$$
$$= \frac{f(t)}{\mathcal{R}(t)} = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1} \tag{1}$$

Please cite this article in press as: J. Bahi et al., Efficient distributed lifetime optimization algorithm for sensor networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.11.010

where $f(t)$ is the probability density function and $F(t)$ is the cumulative distribution function. For a given sensor node i, at time t and t^*, $t^* > t$, its probing rates are $\lambda(t)$ and $\lambda(t^*)$ respectively. Obviously, according to formula 1 and $\beta > 1$, we can get the result $\lambda(t^*) > \lambda(t)$, *i.e.*, the probing rate of a node goes up with time. □

The advantage of self-adjusting the probing rate of each sleeping sensor node is that this avoids two main problems:

(i) A sleeping sensor node does not need to know the number of its direct neighbors to adjust its probing time; this avoids maintaining a current state of the neighborhood, which is especially important in a harsh environment where sensor nodes can fail suddenly.
(ii) No complicated coordination mechanism is needed to perform sending messages between nodes.

4.2. Problem II – resolution

To the best of our knowledge, no realistic solution has been proposed in the literature to ensure the fact that only one sensor node must be in the *active* state for each monitoring zone. The reason why there exists no realistic solution algorithm is that two neighboring sensor nodes may be elected at the same time step, and decision of two neighboring nodes may be the same. This paper aims at filling this gap by proposing an efficient self-stabilizing scheme to improve network's lifetime. Indeed, the use of such a scheme allow sensor network to gracefully degrade in performance instead of failing unpredictably.

We need the following notations and definitions:

Definition 1. Let *Id* be a naming function of sensor nodes. By $Id(i)$, we denote the node identifier of i for each sensor node i. Sensor node identifier [22] is unique if and only if $Id(i) \neq Id(j)$ holds for each $i,j \in V (i \neq j)$. Sensor node identifier is chromatic if and only if $Id(i) \neq Id(j)$ holds for each $(i,j) \in E$. Sensor node identifier is anonymous if and only if $Id(i) = Id(j)$ holds for each $i,j \in V$.

In the present work, we do not consider anonymous sensor networks.

Definition 2. A sensor node can be in one of these three states: *active*, *wakeup* or *sleeping*. The state of a node i is denoted by $\mathcal{S}(i)$.

Definition 3. We say that a sensor node i is independent if $\mathcal{S}(i) = active \wedge (\forall j \in N_i : \mathcal{S}(j) = sleeping \vee wakeup)$ and that i is dominated if $(\mathcal{S}(i) = sleeping \vee wakeup) \wedge (\exists j \in N_i)(\mathcal{S}(j) = active)$

In the following section, we give a fully distributed self-stabilizing algorithm for lifetime optimization in a wireless sensor network. We first focus on the legitimate state formulation and next, we present the algorithm, which consists in only two rules, and give the correctness proofs.

4.2.1. Problem formalization

Let $G = (V; E)$ the graph modeling the sensor network. Let $\mathcal{Z} = \{z_1, z_2, \ldots, z_k\}$ be the set of monitoring zones to be covered and $S = \{1, 2, \ldots, n\}$ the set of sensor nodes. Each zone in $\mathcal{Z}$ has to be covered by at least one sensor node in S. We call ℓ_u the set of *neighbor-sensors* of zone $z_u, 1 \leqslant u \leqslant k$. Each *neighbor-sensor* $j \in \ell_u$ is capable of monitoring the zone z_u (monitoring every point q in z_u), formally:

$$\forall q \in z_u, \quad \forall j \in \ell_u : d(q,j) \leqslant R_s, \quad \ell_u \subseteq S, \quad z_u \in \mathcal{Z},$$

where $d(q,j)$ denotes the distance between point $q \in z_u$ and sensor j.

With respect to Problem I, the legitimate state (let denote it Λ) of the network is then expressed as follows:

$$\forall z_u \in \mathcal{Z} : \exists i,j \in \ell_u | \mathcal{S}(i) = \mathcal{S}(j) = active \Rightarrow i = j$$

In other words, each monitoring zone is covered by at most one sensor node.

4.2.2. Self-stabilizing algorithm

In the following we present the algorithm. We assume distributed/asynchronous scheduler under composite read/write atomicity [13]. We also assume sensor node identifiers to be unique.

The following notations are also given for the predicates of node i

– $\mathcal{A}(i)$: Active neighbor: $\exists j \in \ell_u(i),\ \mathcal{S}(j) = active$.
– $\mathcal{A}^*(i)$: Active neighbor with lower Id: $\exists j \in \ell_u(i)$, $\mathcal{S}(j) = active \wedge Id(i) > Id(j)$.
– $\mathcal{W}^*(i)$: Wakeup neighbor with lower Id: $\exists j \in \ell_u(i)$, $\mathcal{S}(j) = wakeup \wedge Id(i) > Id(j)$.

The self-stabilizing algorithm uses the following two rules:

r_1: **If** $(\mathcal{S}(i) = wakeup \wedge \mathcal{A}(i)) \vee (\mathcal{S}(i) = active \wedge \mathcal{A}^*(i))$ **then** $\mathcal{S}(i) \leftarrow sleeping$

r_2: **If** $\mathcal{S}(i) = wakeup \wedge \neg\mathcal{A}(i) \wedge \neg\mathcal{W}^*(i)$ **then** $\mathcal{S}(i) \leftarrow active$

4.2.3. Self-stabilization proofs

Lemma 2. *Every monitoring zone z_u is eventually covered by at most one sensor node.*

Proof. This is ensured by the rule r_1 *i.e.* if there are two or more active nodes in the same zone, only the one with the lowest Id will remain active. □

Lemma 3. *If a node has executed r_2, then it and each one of its neighbors will execute at most one more rule until their next wakeup, and this rule will be r_1.*

Proof. Let i be a node that executed r_2. When node i passes to active state, all its neighbors are either in *sleeping* or *wakeup* state. So we have two possible scenarios: (i) neighbors in sleeping state: there is no conflict in this case. (ii) neighbors with wakeup state: those neighbors have a higher *Id* than i. □

Please cite this article in press as: J. Bahi et al., Efficient distributed lifetime optimization algorithm for sensor networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.11.010

Lemma 4. *When a node is not sleeping, it can make at most 2 moves.*

Proof. It is easy to see that each rule can be executed at most once by a node. Hence, the only case when a node makes two moves is when it executes r_2 then r_1 with an *active* state. □

Theorem 1. *The proposed algorithm is self-stabilizing with respect to Λ within 2n moves.*

Proof. This follows from Lemma 2 to Lemma 4. □

Corollary 1. *For sensor networks with chromatic node identifier, the proposed algorithm is self-stabilizing with respect to Λ.*

Proof. When a sleeping node wakes up, it sends a probe request message to check if there exist working neighbor nodes in its vicinity. If more than one sensor is candidate to become active at the same time, node's decisions for the tie break resolution are taken by comparing its identifier only with the identifiers of its neighbors even if two or more neighbors of a node have the same identifier. Hence a result. □

5. Theoretical analysis

5.1. Message complexity analysis

When we deal with unreliable communications, the asynchronous mode of the proposed algorithm presents the major advantages of allowing more flexible communication schemes. They are less sensitive to the communication delays and to their variations.

As our scheme do not rely to any node synchronization, and according to rules r_1 and r_2, losses do not prevent the progression of the self-stabilizing process on both the sender (wakeup) and receiver (working) nodes, i.e., the probe/reply messages loss is absorbed/supported by the two rules of the algorithm.

In the following, we give an Upper-Bound of the actual number of *probe/reply* messages exchanged during the network's lifetime optimization task.

Theorem 2. *The number of probe/reply messages involved by the algorithm is at most:*

$$O\left(nm \times \frac{\max_i \, t_i^{\mathcal{R}_i}}{\min_i \, \min_j \Delta_{ji}}\right), \quad 1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant k$$

where n is the number of nodes, m is the number of virtual communication links, $t_i^{\mathcal{R}_i}$ is the reliable life of node i and Δ_{ji} is the jth sleeping period time of node i.

Proof. The reliable life, $t_i^{\mathcal{R}_i}$, of the node i, $1 \leqslant i \leqslant n$ for a specified reliability $\mathcal{R}_i$, starting the mission at age zero, is computed as follows:

$$\mathcal{R}_i = 1 - F(t_i^{\mathcal{R}_i}) = e^{-\left(\frac{t_i^{\mathcal{R}_i}}{\alpha}\right)^{\beta}} \Rightarrow \ln \mathcal{R}_i = -\left(\frac{t_i^{\mathcal{R}_i}}{\alpha}\right)^{\beta} \Rightarrow \ln \frac{1}{\mathcal{R}_i} = \left(\frac{t_i^{\mathcal{R}_i}}{\alpha}\right)^{\beta} \Rightarrow \left(\ln \frac{1}{\mathcal{R}_i}\right)^{\frac{1}{\beta}} = \frac{t_i^{\mathcal{R}_i}}{\alpha} \Rightarrow t_i^{\mathcal{R}_i} = \alpha\left(\ln \frac{1}{\mathcal{R}_i}\right)^{\frac{1}{\beta}}.$$

This is the life for which the sensor node i will be functioning successfully with a reliability of $\mathcal{R}_i$.

According to node's sleeping periods subdivisions of the time, we have:

$0 = t_0 < t_1 < t_2 < \cdots < t_k = t$

Let $\Delta_p = [t_{p-1}, t_p[, 1 \leqslant p \leqslant k$ denote the pth sleeping period time. Thanks to Lemma 1, a node's wakeup rate is monotonically increasing, i.e., the sleeping time period decreases with time. This implies that the probing process of node i costs at most $O\left(\frac{t_i^{\mathcal{R}_i}}{\min_j \Delta_{ji}}\right), 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant k$ In addition, for each probing message issued from node i we may have the corresponding reply messages from its working neighbors. This cost is at most $O(|N_i|)$. Therefore, from the point of view of node i, the number of probe/reply message is at most $O\left(|N_i| \times \frac{t_i^{\mathcal{R}_i}}{\min_j \Delta_{ji}}\right)$.

Finally, summing up for the whole n sensor nodes, the algorithm's message cost is at most

$$O\left(\sum_{i=1}^{n} |N_i| \times \frac{t_i^{\mathcal{R}_i}}{\min_j \Delta_{ji}}\right) \leqslant O\left(nm \times \frac{\max_i t_i^{\mathcal{R}_i}}{\min_i \min_j \Delta_{ji}}\right), \quad 1 \leqslant i \leqslant n, \quad 1 \leqslant j \leqslant k$$ □

Theorem 3. *The upper bound* $O\left(nm \times \frac{\max_i \, t_i^{\mathcal{R}}}{\min_i \, \min_j \Delta_{ji}}\right)$ *is attainable.*

Proof. To see that this bound is really attainable, consider a linear chain graph of only two sensor nodes s_1 and s_2 $(n = 2)$. We need to orchestrate the involved communications between these nodes in time. Let $\beta = 1$ and assume that s_1 is working and s_2 is in the passive state. If $t_1^{\mathcal{R}_1} = t_2^{\mathcal{R}_2}$ (s_1 and s_2 start functioning and will fail at the same time), then the whole number of probe-message issued from s_2 is $\frac{t_2^{\mathcal{R}_2}}{\Delta_2}$, where Δ_2 is the constant sleeping time period of s_2. Since, both s_1 and s_2 have the same life for which nodes will be functioning successfully, node s_1 will reply for each probing message issued from s_2. As a result, the whole number of involved probe-request/reply message before the failure of s_1 and s_2 is $nm \times \frac{\max_i t_i^{\mathcal{R}}}{\min_i \min_j \Delta_{ji}} = 2 \times \frac{t_2^{\mathcal{R}_2}}{\Delta_2}$. □

Corollary 2. *If $\beta = 1$, then the number of probe/reply messages involved by the algorithm is at most:*

$$O\left(nm \times \max_i \frac{t_i^{\mathcal{R}_i}}{\Delta_i}\right), \quad 1 \leqslant i \leqslant n$$

Please cite this article in press as: J. Bahi et al., Efficient distributed lifetime optimization algorithm for sensor networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.11.010

Proof. the proof is straightforward since the sleeping time period Δ_i of the sensor node i is constant in time (exponential case). □

5.2. Reliability analysis

The reliability of the sensor network system ($\mathcal{R}_{SN}$) reflects its tolerance to sensor node failures. It is the dominant factor of quality of services of sensor networks because it gives us the information about the system's survival over time.

In the following, we consider the case of mutually exclusive sets of sensor nodes, where the members of each of those sets together completely cover the monitoring zones. The intervals of activity are the same for all sets, and only one of the sets is active at any time to provide continuous service while the remaining sets are scheduled to sleep.

We need to orchestrate the tradeoffs relation between sensor network's reliability and the required number of cover sets, i.e., we would like to determine, for a fixed reliability threshold, what is the minimum number of cover sets that can be used in the system's coverage while achieving the prescribed reliability of the sensor network?

Theorem 4. *Let f_s be the failure probability of a node $s \in S$. The minimum number of cover sets to meet the prescribed reliability threshold $\mathcal{R}^*_{SN}$ of the sensor network system is*

$$\eta^* = \left\lceil \frac{\log(\mathcal{R}^*_{SN})}{\log\left(\min\limits_{i=1..\eta}\prod_j(1-f_j)\right)} \right\rceil, \quad 1 \leqslant j \leqslant |C_i|$$

where η is the maximum number of mutually exclusive cover sets C_i.

Proof. A cover set is considered "complete" if and only if it contains the necessary sensor nodes to cover all zones. This is logically equivalent to a serial connection of components which turn up in many different systems. Thus, the overall reliability of a given cover set C_i can be calculated as the probability of all nodes executing successfully. In other words, it is the product of the individual reliabilities:

$$\prod_{1 \leqslant j \leqslant |C_i|} \mathcal{R}_j = \prod_{1 \leqslant j \leqslant |C_i|} (1-f_j)$$

Similarly, the algorithm's coverage is successful if and only if each cover set is operational (no failure) while it is covering its monitoring zones.

Intuitively, the maximum number of mutually exclusive cover sets η is given/defined by the size of the smallest set of sensors covering a monitoring zone. Formally:

$$\eta = \min_{u=1\ldots k} |\ell_u|$$

As a result, the reliability of the sensor networks (SN) is the probability that the algorithm coverage can run successfully during the mission. Formally:

$$\mathcal{R}_{SN} = \prod_i \prod_j \mathcal{R}_j = \prod_i \prod_j (1-f_j), \quad 1 \leqslant i \leqslant \eta, \quad 1 \leqslant j \leqslant |C_i|$$

The computation of the number of cover sets is based on the following condition:

$$\mathcal{R}_{SN} \geqslant \mathcal{R}^*_{SN} \tag{2}$$

We have,

$$\mathcal{R}_{SN} = \prod_i \prod_j (1-f_j) \geqslant \prod_i \min_{i=1\ldots\eta} \prod_j (1-f_j)$$

From (2), we get

$$\mathcal{R}^*_{SN} \leqslant \prod_i \min_{i=1..\eta} \prod_j (1-f_j) = \left(\min_{i=1..\eta} \prod_j (1-f_j)\right)^{\eta} \Rightarrow \log(\mathcal{R}^*_{SN}) \leqslant \eta \times \log\left(\min_{i=1\ldots\eta} \prod_j (1-f_j)\right)$$

We obtain

$$\eta \geqslant \frac{\log(\mathcal{R}^*_{SN})}{\log\left(\min_{i=1\ldots\eta}\prod_j(1-f_j)\right)}$$

Finally, we derive the minimum number η^* of cover sets

$$\eta^* = \left\lceil \frac{\log(\mathcal{R}^*_{SN})}{\log\left(\min_{i=1\ldots\eta}\prod_j(1-f_j)\right)} \right\rceil, \quad 1 \leqslant j \leqslant |C_i|$$ □

6. A brief description of PEAS and PCP protocols

In order to compare our algorithm to PEAS [36] and PCP [19] protocols, which are the closest works to the one presented in this paper, we briefly outline here the main features of these algorithms.

6.1. PEAS algorithm

Initially, nodes are in the sleeping mode. Each node sleeps for an exponentially distributed time generated according to a Probability Density Function (PDF) $f(t) = \lambda e^{-\lambda t}$, where λ is the probing rate of the sensor node and t denotes its sleeping time duration.

The inter-wakeup time of each sleeping node i, $1 \leqslant i \leqslant n$ is related to an exponential law with parameter λ_i and probings from different sleeping neighbors construct a Poisson process with parameter λ, the sum of all sleeping nodes' rate $\lambda_i : \lambda = \sum_{i=1}^{n} \lambda_i$. When a sleeping node wake-ups, it sends/broadcasts a *ping-req* message. Thus, the working sensor node can perform the enumeration of the whole *ping-req* messages received from the passive sensors and therefore the total number of probing messages.

Each working sensor node maintains two parameters: (i) $\mathcal{K}$ a counter that records how many *ping-req* messages have been received, and (ii) t_0 the most recent time when $\mathcal{K}$ is set to 0. When the working node hears the first *ping-req* message, it sets the counter $\mathcal{K}$ to 0, and t_0 to the current time t. After that, each time it receives a new *ping-req* message, the counter is incremented by one until a threshold value is reached. The threshold $\mathcal{K}$, determines the accuracy of the estimation. In [35], it is shown that a value $\mathcal{K} > 16$ allows an approximation with an error of 1%. Once $\mathcal{K}$ is reached, a measurement $\lambda_{\text{Estimated}}$ of the actual probing rate λ is computed as follows: $\lambda_{\text{Estimated}} = \frac{\mathcal{K}}{t - t_0}$, where t is the current time. The above process is repeated after setting t_0 to t and reseting the counter to 0.

Upon receiving a *ping-ack* reply message from the working node, a probing node i updates its actual probing rate λ_i by taking into account the received $\lambda_{\text{Estimated}}$: $\lambda_i^{new} \leftarrow \lambda_i \cdot \frac{\lambda_D}{\lambda_{\text{Estimated}}}$. Then, a new sleeping period is generated by using the new computed parameter λ_i^{new} according to the PDF function: $f(t) = \lambda^{new} e^{-\lambda^{new} t}$.

Unlike PEAS protocol, the main advantage of using the Weibull distribution instead of the exponential model is that Weibull law natively takes into account the wear-out of sensors, thus progressively increasing the rate of wake ups over time in an implicit manner. The exponential model on the other hand provides a constant average of wake ups over time and the update of this rate can only be done explicitly by exchanging messages between nodes and hence consuming more energy we want to save.

6.2. PCP algorithm

The idea of PCP is to activate a subset of deployed sensors to form an approximate triangular lattice over the area to be covered. PCP works in rounds of R seconds each. R is chosen to be much smaller than the average lifetime of sensors. In the beginning of each round, all nodes start running PCP independent of each other, then a number of messages will be exchanged between nodes to determine active/sleep nodes. PCP refers to the distance between the vertices of the triangular lattice as the maximum separation between working nodes, and it is denoted by s. The value of s is computed from the sensing range r_s of sensor nodes. In the disk sensing model, the maximum separation is set to $\sqrt{3} r_s$ as it has been shown in [4].

Under the exponential sensing model, to ensure that the probability of sensing at the least-covered point is at least the coverage threshold parameter θ, the authors compute the maximum separation, s, as: $\sqrt{3}\left(r_s - \frac{\ln\left(1-\sqrt[3]{1-\theta}\right)}{\alpha}\right)$, where α is a factor that describes how fast the sensing capacity decays with distance. From this equation, it is clear that if we set $\alpha = \infty$, then the exponential sensing model reduces to the disk sensing model.

Unlike our algorithm, PCP protocol assumes nodes to have positioning informations. This helps to construct a triangular lattice allowing high performance for the coverage issue. The algorithm we develop is completely distributed and does not need to know the positions of any sensor node in the network.

7. Simulation results

In this section, we discuss some results through simulations. We consider a flat grid topology of 10 by 10 *i.e.* 100 monitoring zones. We vary the number of sensors between 200 and 1600 nodes. Since sensors are uniformly distributed on the monitoring area, the density of sensors at each zone varies between 2 and 16.

The performance evaluation considers four aspects: (i) Network lifetime evolution (see Fig. 1); (ii) Effective monitoring time (see Fig. 2): this measure is related to the time between the death of the active node in a monitoring zone and its replacement; it is expressed in (%); (iii) Total number of messages (see Fig. 3)); and (iv) Number of awakenings per inactive sensors (see Fig. 4). Obtained results are compared with PEAS protocol [36] (to the best of our knowledge, PEAS protocol is the closest work to the one presented in this paper) and to PCP protocol [19]. Here, we use an optimistic implementation configuration for both PEAS and PCP, i.e, we assume that links are faithful/faultless and loss-less (recall that our algorithm does not need such assumptions, see Section 5.1).

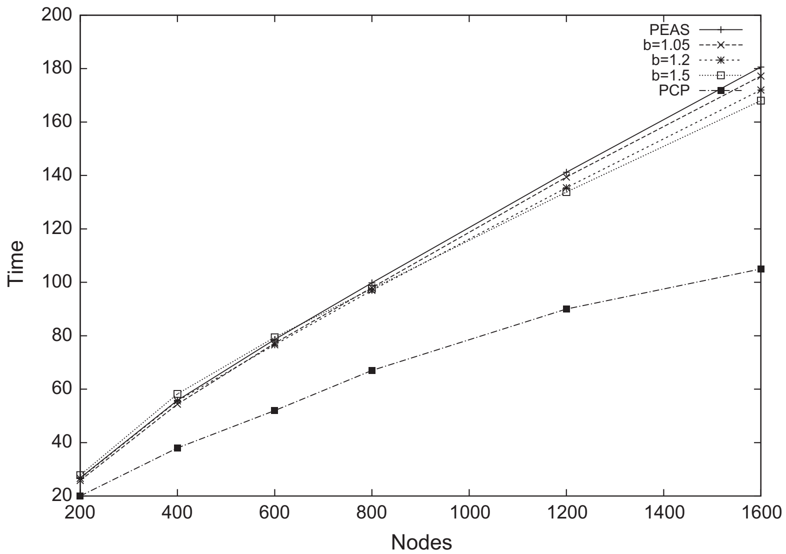

Fig. 1. Lifetime evolution.

Please cite this article in press as: J. Bahi et al., Efficient distributed lifetime optimization algorithm for sensor networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.11.010

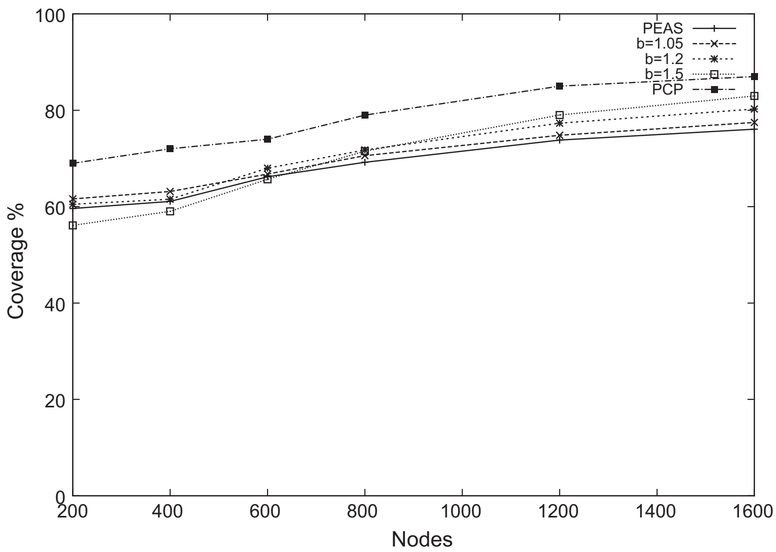

Fig. 2. Effective monitoring time.

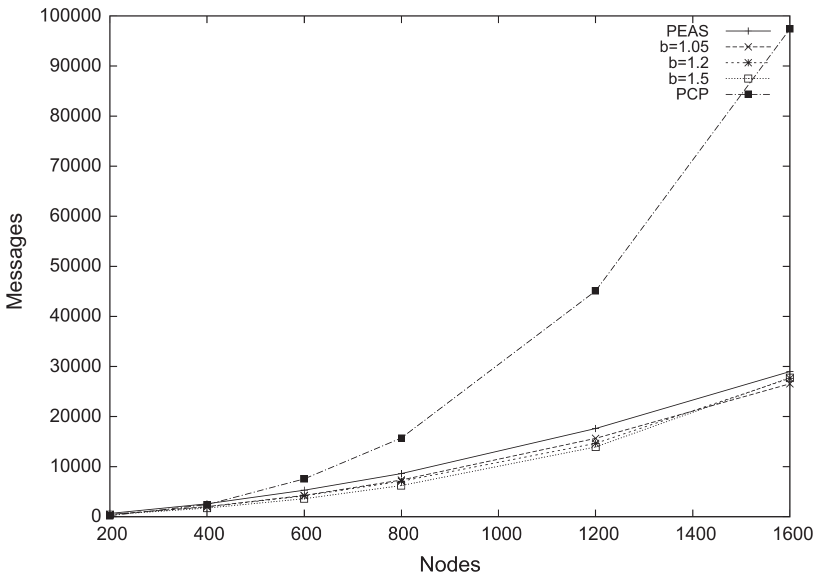

Fig. 3. Total number of messages.

All the four aspects are studied according to different network sizes (densities) and the factor β which is fixed respectively to 1.05, 1.2 and 1.5. Moreover, the parameter α is aligned on the average lifetime of a sensor (which is calculated according to the battery capacity). In our tests, it is fixed to 10 time units. The referential case is when there is no inactive sensors at the initialization. Hence, each monitoring zone contains exactly 1 sensor (100 sensors in the network). In this case, the obtained network lifetime is 10, the number of awakenings is 0 (since there is no inactive sensors) and the effective monitoring time is 100%.

First, we compare with PEAS. We can observe that for the effective monitoring time, the choice of the value of β is important. In fact, if $\beta = 1.05$, the behavior is similar to PEAS but slightly improved. Moreover, according to the different values of the network's density, the parameter β behaves differently. From the obtained curves, we can see that for values between 200 and 500 nodes, $\beta = 1.05$ is the best choice, from 500 to 800 nodes, $\beta = 1.2$ gives the best performance, and finally, from 800 to 1600, it is $\beta = 1.5$ that outperforms the others. The global network lifetime evolves similarly for all considered configurations until 800 nodes, then we can observe that PEAS protocol

Please cite this article in press as: J. Bahi et al., Efficient distributed lifetime optimization algorithm for sensor networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.11.010

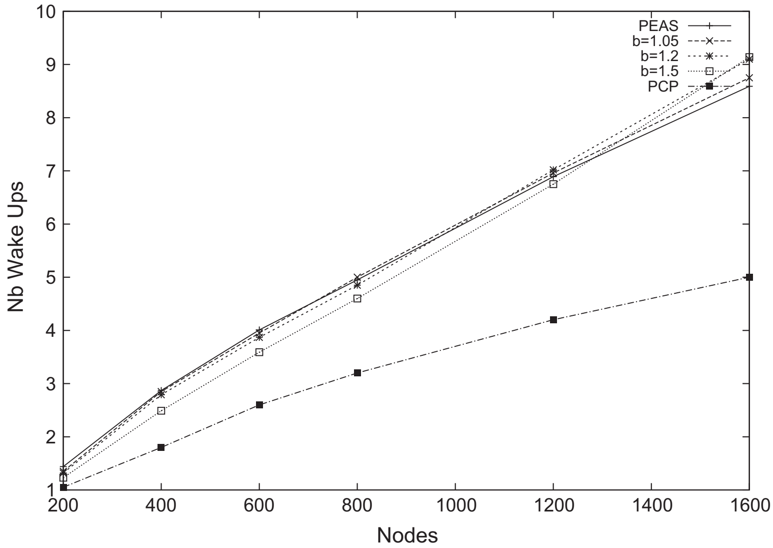

Fig. 4. Awakenings per node.

gives the longest lifetime. This is due to the fact that there are more gaps in the effective monitoring time, hence the global time will be improved but with less effective monitoring time. Moreover, the difference is at most about 5%. Our protocol benefits from its self-stabilization mechanism to optimize the number of exchanged messages. Thus, the protocol we proposed consumes less messages than PEAS. In addition, the number of wake-ups is quite similar for all configurations.

Let us now consider PCP. As it was predictable, PCP gives the highest performance for the covering metric. This is due to the triangular lattice structure of activated nodes that guarantee high covering quality. However, our protocol remains close to PCP performance especially for higher density configurations, from 800 to 1600, where the difference reaches only 5% as shown by Fig. 1. However, in global network lifetime and message consumption, our proposition behaves much better as illustrated by Figs. 2 and 3. In addition, the number of wake-ups is closely related to global network lifetime as shown by Fig. 4

8. Conclusion

In this paper, we have addressed the problem of lifetime optimization in wireless sensor networks. This is a very natural and important problem, as several unexpected node failures may occur during the network's service. To cope with this problem, a distributed self-stabilizing and wear-out-aware algorithm is presented and theoretically/experimentally analyzed. Our algorithm seeks to build resiliency by maintaining a necessary set of working nodes and replacing failed ones when needed. The proposed algorithm is able to increase the lifetime of wireless sensor networks, especially when the reliabilities of sensor nodes are expected to decrease due to use and wear-out effects. Thus, we conclude that the use of such a scheme offers the potential to extend system life and allow sensor network platforms to gracefully degrade in performance instead of failing unpredictably.

Acknowledgments

We thank the anonymous referees for all the valuable comments that helped us to improve the quality of the paper.

References

[1] Zoë Abrams, Ashish Goel, Serge A. Plotkin, Set k-cover algorithms for energy efficient monitoring in wireless sensor networks, in: IPSN, 2004, pp. 424–432.

[2] I. Akyildiz, W. Su, Y. Sankarasubramniam, E. Cayirci, A survey on sensor networks, IEEE Comm. Mag. (2002) 102–114.

[3] Thara Angskun, Graham E. Fagg, George Bosilca, Jelena Pjesivac-Grbovic, Jack Dongarra, Self-healing network for scalable fault-tolerant runtime environments, Future Gener. Comp. Syst. 26 (3) (2010) 479–485.

[4] Xiaole Bai, Santosh Kumar, Dong Xuan, Ziqiu Yun, Ten-Hwang Lai, Deploying wireless sensors to achieve both coverage and connectivity, in: MobiHoc, 2006, pp. 131–142.

[5] Joffroy Beauquier, Julien Clement, Stephane Messika, Laurent Rosaz, Brigitte Rozoy, Self-stabilizing counting in mobile sensor networks, in: PODC '07: Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing, ACM, New York, NY, USA, 2007, pp. 396–397.

[6] P. Berman, G.Calinescu, Power efficient monitoring management in sensor networks, in: Proceedings of IEEE Wireless Communication and Networking Conference (WCNC'04), 2004, pp. 2329–2334.

[7] Carlo Caione, Davide Brunelli, Luca Benini, Distributed compressive sampling for lifetime optimization in dense wireless sensor networks, IEEE Trans. Ind. Inform. 8 (1) (2012) 30–40.

[8] Mihaela Cardei, Ding-Zhu Du, Improving wireless sensor network lifetime through power aware organization, Wirel. Netw. 11 (3) (2005) 333–340.

[9] Mihaela Cardei, David MacCallum, Maggie Xiaoyan Cheng, Manki Min, Xiaohua Jia, Deying Li, Ding-Zhu Du, Wireless sensor networks with energy efficient organization, J. Interconnect. Netw. 3 (3–4) (2002) 213–229.

[10] Mihaela Cardei, My T. Thai, Yingshu Li, Weili Wu, Energy-efficient target coverage in wireless sensor networks, in: INFOCOM, 2005, pp. 1976–1984.

[11] Manju Chaudhary, Arun K. Pujari, High-energy-first (hef) heuristic for energy-efficient target coverage problem, Int. J. Ad hoc, Sens. Uniquitous Comput. (IJASUC) 2 (1) (2011).
[12] Edsger W. Dijkstra, Self-stabilizing systems in spite of distributed control, Commun. ACM 17 (11) (1974) 643–644.
[13] Shlomi Dolev, Self-Stabilization, MIT Press, 2000.
[14] Antoine Gallais, Jean Carle, David Simplot-Ryl, Ivan Stojmenovic, Localized sensor area coverage with low communication overhead, in: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications, 2006, pp. 328–337.
[15] Naveen Garg, Jochen Koenemann, Faster and simpler algorithms for multicommodity flow and other fractional packing problems, in: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, FOCS '98, 1998, pp. 300–309.
[16] Sandeep K.S. Gupta, Pradip K. Srimani, Self-stabilizing multicast protocols for ad hoc networks, J. Parall. Distrib. Comput. 63 (1) (2003) 87–96 (Wireless and Mobile Ad Hoc Networking and Computing).
[17] Shibo He, Jiming Chen, Xu Li, Xuemin (Sherman) Shen, Youxian Sun, Leveraging prediction to improve the coverage of wireless sensor networks, IEEE Trans. Parall. Distrib. Syst. 23 (4) (2012) 701–712.
[18] Shibo He, Jiming Chen, David K.Y. Yau, Huanyu Shao, Youxian Sun, Energy-efficient capture of stochastic events under periodic network coverage and coordinated sleep, IEEE Trans. Parall. Distrib. Syst. 23 (6) (2012) 1090–1102.
[19] Mohamed Hefeeda, Hossein Ahmadi, Energy-efficient protocol for deterministic and probabilistic coverage in sensor networks, IEEE Trans. Parall. Distrib. Syst. 21 (5) (2010) 579–593.
[20] Wendi B. Heinzelman, Anantha P. Chandrakasan, Hari Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Trans. Wirel. Commun. 1 (4) (2002) 660–670.
[21] Jaap-Henk Hoepman, Andreas Larsson, Elad M. Schiller, Philippas Tsigas, Secure and self-stabilizing clock synchronization in sensor networks, in: SSS'07: Proceedings of the 9h international conference on stabilization, safety, and security of distributed systems, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 340–356.
[22] Michiyo Ikeda, Sayaka Kamei, Hirotsugu Kakugawa, A space-optimal self-stabilizing algorithm for the maximal independent set problem, in: PDCAT, 2002, pp. 70–74.
[23] Jayakrishnan V. Iyer, Heeyeol Yu, Hogil Kim, Eun Jung Kim, Ki Hwan Yum, Pyeong Soo Mah, Assuring k-coverage in the presence of mobility and wear-out failures in wireless sensor networks, IJSNet 5 (1) (2009) 58–65.
[24] Gaurav S. Kasbekar, Yigal Bejerano, Saswati Sarkar, Lifetime and coverage guarantees through distributed coordinate-free sensor activation, IEEE/ACM Trans. Netw. 19 (2) (2011) 470–483.
[25] Jae-Joon Lee, Bhaskar Krishnamachari, C.-C.Jay Kuo, Impact of energy depletion and reliability on wireless sensor network connectivity, in Proceedings of the SPIE Defense and Security, 2004.
[26] Jae-Joon Lee, Bhaskar Krishnamachari, C.-C. Jay Kuo, Node aging effect on connectivity of data gathering trees in sensor networks. in: Proceedings of the 60th IEEE Vehicular Technology Conference, 2004, pp. 4630–4634.
[27] Daniel Nurmi, Rich Wolski, John Brevik, Model-based checkpoint scheduling for volatile resource environments, in: Proceedings of Cluster 2005, 2005.
[28] S. Pansrisu, S. Premrudeepreechacharn, Preventive maintenance optimization for excitation system of bhumibol hydro power plant, in: International Conference on Energy and Sustainable Development: Issues and Strategies (ESD'10), 2010.
[29] Sasa Slijepcevic, Miodrag Potkonjak, Power efficient organization of wireless sensor networks, in: International Conference on Communications (ICC'01), 2001, pp. 472–476.
[30] Di Tian, Nicolas D. Georganas, A coverage-preserving node scheduling scheme for large wireless sensor networks, in: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, WSNA '02, ACM, 2002, pp. 32–41.
[31] Di Tian, Nicolas D. Georganas, Connectivity maintenance and coverage preservation in wireless sensor networks, Ad Hoc Netw. 3 (6) (2005) 744–761.
[32] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, Christopher Gill, Integrated coverage and connectivity configuration in wireless sensor networks, in: SenSys: Proceedings of the 1st international on Embedded networked sensor systems, 2003, pp. 28–39.
[33] Roger Wattenhofer, Li Li, Paramvir Bahl, Yimin Wang, Distributed topology control for power efficient operation in multihop wireless ad hoc networks, in: INFOCOM, 2001, pp. 1388–1397.
[34] Ya Xu, John S. Heidemann, Deborah Estrin, Geography-informed energy conservation for ad hoc routing, in: MOBICOM, 2001, pp. 70–84.
[35] Fan Ye, Honghai Zhang, Songwu Lu, Lixia Zhang, Jennifer C. Hou, A randomized energy-conservation protocol for resilient sensor networks, Wirel. Netw. 12 (5) (2006) 637–652.
[36] Fan Ye, Gary Zhong, Jesse Cheng, Songwu Lu, Lixia Zhang, Peas: a robust energy conserving protocol for long-lived sensor networks, in: Proceedings of the 23rd International Conference on Distributed Computing Systems, ICDCS'03, 2003, pp. 28–37.
[37] Honghai Zhang, Jennifer C. Hou, Maintaining sensing coverage and connectivity in large sensor networks, Ad Hoc Sens. Wirel. Netw. 1 (1–2) (2005).
[38] D. Zorbas, D. Glynos, P. Kotzanikolaou, C. Douligeris. B{GOP}: an adaptive coverage algorithm for wireless sensor networks, in: Proceedings of the 13th European Wireless Conference, EW'07, 2007.

Jacques Bahi was born in July, 25th 1961, after a Master of Science in Applied Mathematics, he received a PhD in Applied Mathematics from the university of Franche-Comte in 1991. From 1992 to 1999, he was a associate professor of applied mathematics at the Mathematical Laboratory of Besanon, his research interests were focused on parallel synchronous and asynchronous algorithms for differential-algebraic equations and singular systems. Since september 1999, he became a full professor of computer science at the University of Franche-Comte. He published about 50 articles in peer reviewed journal and more than 65 articles in peer reviewed international conferences and 2 scientific books. Jacques Bahi is a IEEE senior member since 2009. He is the head of the Distributed Numerical Algorithms team of the Computer Science Laboratory of Besanon, he supervised 15 PhD students. He is a member of the editorial board of 2 international journals and is regularly a member of the scientific comitees of many international conferences. Currently, he is interested in (1) high performance computing, (2) distributed numerical algorithms for ad hoc and sensor networks and (3) dynamical systems with application to data hiding and privacy.

Mohammed Haddad is currently an associate professor in computer science at Claude Bernard Lyon 1 University (France). He received his PhD degree and MS degree in computer science from Claude Bernard Lyon 1 University. His research interests include graph theory fundamentals and their applications in distributed systems (ad hoc and sensor networks, social networks, etc).

Mourad Hakem received the PhD degree from Paris Nord University in 2006. He worked as a lecturer in the LIPN Laboratory at Paris Nord University from 2006 to 2007 and joined the LIP Laboratory at ENS Lyon where he spent one year as a postdoctoral researcher. Currently, he is an associate professor at the University of Franche Comté. His main research interests are scheduling, fault tolerance techniques and parallel and distributed algorithm design for sensor networks, clusters and grids.

Please cite this article in press as: J. Bahi et al., Efficient distributed lifetime optimization algorithm for sensor networks, Ad Hoc Netw. (2013), http://dx.doi.org/10.1016/j.adhoc.2013.11.010

Hamamache Kheddouci is currently a professor in computer science at Claude Bernard Lyon 1 University, France. He is the Head of GAMA laboratory. He passed 5 years at Burgundy University, Dijon as associate professor in computer science. He received his PhD degree from Paris-XI University, Orsay in January 1999. His research focuses Graphs, Algorithms and Applications. (see http://www710.univ-lyon1.fr/hkheddou/)