Network Protocols and Algorithms
ISSN 1943-3581
2010, Vol. 2, No. 2

Cluster-based Energy-efficient k-Coverage for Wireless Sensor Networks

Mohamed LEHSAINI, Hervé GUYENNET

LIFC "Laboratory of Computer Science of Franche-Comte University",
16, Route de Gray, Besançon 25030, France.

Tel: +33 3 81 66 20 75   E-mail: mlehsaini@lifc.univ-fcomte.fr

Tel: +33 3 81 66 64 57   E-mail: hguyennet@lifc.univ-fcomte.fr

Mohammed FEHAM

Faculty of Technology, University of Tlemcen

B.P 230, Tlemcen 13 000, Algeria

Tel: +213 43 28 56 89   E-mail: m_feham@mail.univ-tlemcen.dz

Received: February 23, 2010   Accepted: June 12, 2010   DOI: 10.5296/npa.v2i2.325

Abstract

Coverage preservation and prolonging lifetime are the fundamental issues in wireless sensor networks. Due to the large variety of applications, coverage is subject to a wide range of interpretations. Some applications require that every point in the area is monitored by only one sensor while other applications may require that each point is covered by at least k sensors (k>1) to achieve fault tolerance. Hence, it is desirable to activate a minimum number of sensors that are able to ensure coverage area and turn off some redundant sensors to save energy and therefore extend network lifetime. Furthermore, determining a minimum number of active sensors is based on the degree coverage required and its level. In this paper, we propose a cluster-based efficient-energy coverage scheme called CSA_VS (Cluster-based Scheduling Algorithm–Virtual Sensor) to ensure the full coverage of a monitored area while saving energy. CSA_VS uses a novel sensor-scheduling scheme based on the k-density and the remaining energy of each sensor to determine the state of all the deployed sensors to be either active or sleep as well as the state durations. Simulation results showed that CSA_VS provides better performance in terms of the number and the percentage of active sensors to guarantee the area coverage compared to other algorithms.

Keywords: Cluster-based, Energy-efficient, k-coverage, Sensor-scheduling, Virtual sensor.

1. Introduction

A wireless sensor network (WSN) consists of a large number of sensor nodes deployed over a geographical area for monitoring physical phenomena like temperature, humidity, vibrations, seismic events, and so on [1]. These sensor nodes collaborate on a global sensing task and deliver required data to one or more remote sinks. Typically, a sensor node is a tiny device that includes three basic components: a sensing subsystem for data acquisition from the physical surrounding environment, a processing subsystem for local data processing and storage, and a wireless communication subsystem for data transmission. Sensor nodes are usually powered by lightweight batteries, and replacing or recharging these batteries is often not feasible because sensor nodes may be deployed in a hostile or unpractical environment. Moreover, WSN should have a lifetime long enough to fulfill the application requirements. Thus, low power consumption becomes a critical factor to be considered, especially in the design of algorithms and network protocols at all layers of the network architecture.

In WSNs, sensors are generally deployed in large number to observe an area; it results from that the emergence of points within the area monitored which are covered by several sensors. So, to save energy, it should be necessary to schedule the sensor activity such that to allow redundant sensors to enter the sleep mode as often and for as long as possible. To design such a sensor-scheduling scheme, one should answer the following questions: (1) Which rule should each sensor follow to determine whether to enter active mode or sleep mode? (2) How long should a sensor remain in the active mode? (3) How well an area is monitored or tracked by sensors.

Several works addressed the problem of active sensor selection, also known as sensor-scheduling, in WSNs [2,3]. However, at the best of our knowledge, our work is the first attempt that implies the position of a sensor represented by its k-density and its remaining energy to select the active sensors. Moreover, we tackled the problem of selecting a reduced set of active sensors among the deployed sensors so that these sensors ensure the full k-coverage of the monitored area where each position in the area is monitored by at least k ($k \geq 1$) sensors. The good choice of this set is essential because it could reduce energy consumption, and thus prolongs network lifetime.

In this paper, we considered the problem of sensor-scheduling activities to guarantee area coverage while maximizing network lifetime. For that, we proposed a cluster-based energy-efficient coverage algorithm called CSA_VS (Cluster-based Scheduling Algorithm – Virtual Sensor) to deal with the problem of preservation coverage and the problem of saving energy. We used clustering approach because it permits to save energy by avoiding frequent communication collisions and redundant messages in a sensor network since only the cluster-heads that are responsible for transmitting the collected data to the remote sink, directly or via multi-hop transmission mode. Moreover, CSA_VS determines whether every point on the monitored area of a wireless sensor network is covered by k ($k \geq 1$) sensors or not.

CSA_VS is performed as follows: first, it schedules sensor activities to maintain the full area coverage based on the algorithm presented in [4]. Then, it evaluates the coverage ratio of the area using virtual sensor approach. If this coverage ratio is lower than the ratio required, CSA_VS would improve it while activating other sensors. Moreover, when the coverage ratio is less than 100% then there are holes in the monitored area i.e. points not covered in the monitored area. The emergence of these holes may be due to the failure of sensors or there exist sensors in sleeping mode in these regions.

The simulation results showed that CSA_VS carries out the coverage degree required for various areas inside the monitored area, and that is more powerful than some centralized and located k-coverage algorithms as LPA, CKA and PKA [5].

The rest of this paper is organized as follows: in Section 2, we provide the necessary preliminary information for describing our scheme; Section 3 reviews several k-coverage algorithms that have been previously proposed; in Section 4, we present our novel sensor-scheduling scheme; and Section 5 presents a performance analysis of the proposed scheme. Finally, we conclude our paper and discuss future research work in Section 6.

2. Background of the coverage problem

The sensors are usually deployed with large number and their sensing areas are overlapped between the neighbors, which leads to a large number of redundant sensors. Hence, we can active a reduced number of sensors able to ensure full area coverage and turn off some redundant nodes to save energy. Furthermore, determining a minimum number of active sensors is based on the degree coverage required and its level. If each position in the area is monitored by at least k ($k \geq 1$) sensors, the sensor network is said to be a k-coverage sensor network where k is the coverage degree. Moreover, when the coverage is less than 100% then there are holes in the monitored area i.e. points not covered in the monitored area. The emergence of these holes may be due to failure of sensors or their disabling.

To facilitate the future description, we first define some terms and notations used by our approach. We view a wireless sensor network from a sensing perspective.

2.1. Notations and assumptions

We adopt the following notations and assumptions throughout the paper.

- Consider a set of sensors $S = \{s_1, s_2, \ldots, s_n\}$, distributed in a two-dimensional Euclidean plane P,
- Define A as the area where sensors are initially deployed,
- Each sensor is placed in *A* at coordinates (x_i, y_i) and it knows its own location,
- The sensing region of each sensor is a disk, centered at the sensor, with radius R_s, its sensing range,
- Assume that all sensors have the same sensing range R_s,
- Assume that each sensor has omnidirectional antenna i.e. it can do 360° observation.

2.2. Sensing model

2.2.1 Coverage area

The sensing region of a sensor s_i placed at coordinates (x_i, y_i) is represented by the surface: $A(s_i)=\{X \in P \mid d(X,X_i) \leq R_s\}$ where $d(X,X_i)$ is the Euclidean distance between sensor s_i and each point of the plane P. $A(s_i)$ represents the maximal circular area centered at a sensor s_i that can be covered by s_i. The radius of $A(s_i)$ is called sensor s_i's sensing range. We also define the coverage A(T) of a sensor set $T = \{s_1, s_2, ..., s_p\}$, as the union of sensing area covered by each sensor in T i.e. $A(T) = A(s_1) \cup A(s_2) \cup .. A(s_p)$. We said that T covers fully an area of targets A if and only if $A \subseteq A(T)$.

2.2.2 1-coverage point

A point $p \in A$ is covered by a sensor s_i if it is within the sensing range of s_i:

p is covered by $s_i \Leftrightarrow p \in A(s_i)$.

We define the set of sensors that covers a point $p \in A$ by:

$\text{Cover}(p) = \{s_i \in S \mid p \in A(s_i)\}$

If $\text{Cover}(p) \neq \emptyset$, p is covered by at least one sensor otherwise it is a hole in the monitored area.

2.2.3 k-coverage point

A point $p \in A$ is k-covered if it is in the sensing range of at least k sensors:

p is k-covered $\Leftrightarrow |\text{Cover}(p)| \geq k$
where $|\text{Cover}(p)|$ is the cardinality of the set Cover(p).

2.2.4 1-coverage area

A specific area A is said 1-covered by a set of sensors S if and only if each point *p* within *A* is covered by at least one sensor of *S*:

A is 1-covered $\Leftrightarrow \quad \forall p \in A \quad \text{Cover}(p) \neq \emptyset$

2.2.5 k-coverage area

A specific area A is said k-covered by a set of sensors S if and only if each point p within *A* is covered by at least k sensors of *S*:

A is k-covered $\Leftrightarrow \quad \forall p \in A \quad |\text{Cover}(p)| \geq k$

Considering a number *k*, k-coverage problem is a decision problem whose goal is to determine if each point in *A* is k-covered or not. We define k-coverage area where *k* represents the minimal number of sensors that cover each point p of the area A:

$k=\min \{|\text{cover}(p)| \; \forall p \in A\}$

3. Related work

Due to the large variety of applications area coverage could take several forms. For example, in the least sensitive applications such as the monitoring of the agricultural fields, we can conceive coverage protocols wherein each point in the monitored area is observed by only one sensor and in certain cases these protocols do not guarantee the full coverage area necessarily. However, in the sensitive applications like military applications or those related to the security, it should be necessary to ensure the coverage of each point within the monitored area by more than one sensor to achieve fault tolerance.

In this section, we present existing work related to each of the two forms of coverage: 1-coverage and k-coverage.

3.1. 1-coverage

Ye et al. [6] have proposed an area coverage algorithm called PEAS (Probing Environment and Adaptive Sleeping) for asynchronous sensor networks. In PEAS, the sensors use a simple rule to decide about their activities. If a sensor does not find another active sensor in its probing range, it will become active; otherwise it returns to the sleeping mode. The sensors pass to active mode based on a threshold distance *P* separating them from their neighbors and an active sensor remains waked up until it undergoes a failure or its battery is exhausted. Then, the sensors that are in sleeping mode, replace the failing sensors being in their vicinity, which it makes PEAS as a fault-tolerant protocol. This technique may not be desirable because the density of active sensors will degrade over time. Moreover, the failure of sensors could cause network division into unconnected sub-networks and create isolated sensors. Furthermore, in an environment where the main cause of sensors' failure is batteries depletion, it is desirable to balance energy consumption among all sensors of the network. For that, the selection of active sensors should be done periodically and based on several factors such as remaining energy and k-density. In addition, PEAS does not guarantee full coverage area unless a close relationship should be established between the distance threshold *P* and the sensing range R_s.

In [7], Gui and Mohapatra have proposed an extension of PEAS, called PECAS (Probing Environment and Adaptive Sleeping Collaborating) to overcome PEAS' limitations. In PECAS, a node remains in active mode only for a period given unlike PEAS where an active node remains in this state until it suffers a failure or it exhausts its battery.

Cai et al. [8] have developed an area coverage protocol for asynchronous sensor networks called ACOS (Area-based Collaborative Sleeping). This protocol improves PECAS performance and introduces the collaboration concept in order to balance energy consumption among sensors. In ACOS, a sensor can be in one of the following states: passive, active, pre-active or pre-passive and each sensor is able to calculate the portion of its surface which is not covered by any other sensor. A sensor *u* becomes active if the surface of its portion not covered by its 1-neighbors exceeds a certain threshold during a given period T_w. Therefore, u and the 1-neighbors of its neighbors may switch to active mode simultaneously. In addition, in ACOS, it is difficult to coordinate between the sensors that are in pre-passive mode and want to switch to passive mode.

In [9], the authors have presented a random sleep algorithm for area coverage. In this algorithm, a sensor can switch to passive mode if it obtains permission from its 1-neighbors. This permission includes the period during which the sensor will remain in the passive state. Furthermore, before sending permission, there are information exchanges between sensors about the energy level of their batteries. Thus, a sensor that is in the active state and with low energy level tends to become passive contrary to a sensor that has more energy. Although, this algorithm involves the energy criterion for selecting active sensors and that has a distributed aspect, it could generate a large number of active sensors that would have a negative impact on network lifetime.

In [10], Sheu et al. have proposed a localized algorithm based on priority sensors for selecting active sensors knowing the priorities of their neighbors. Initially, each sensor u sends its priority to its 1-neighbors within its sensing area. Then, it considers the perimeter portions of its neighbors with highest priority being in its sensing area. If these portions are covered by other neighbors with high priority then u could pass to passive mode. Moreover, the set of the active sensors is enough to construct a tree connected being used to transmit information from each sensor to the base station.

In [11], the authors proposed a technique for the problem of the area coverage. They assumed that all sensors have the same radius of sensing and communication and the sensing range is equal to the communication range. Initially, each sensor u sends a HELLO message to establish the list of its 1-neighbors. Then, it evaluates the areas covered by each one of its 1-neighbors using area-perimeter approach. If all surfaces covered by its neighbors cover its sensing area during a random period, u can become passive. The process is repeated until each sensor in the network decides its statute. However, in this technique, if an active sensor disappears without advising its neighborhood i.e. when its battery is exhausted or it fails, then its sensing area could be considered when evaluating areas covered by its active neighbors.

In [12], Zhang and Hu proposed an algorithm based on the geographical density control (Optimal Geographic Density Control) called OGDC. OGDC can configure a sensor network so that it provides full area coverage, network connectivity and energy conservation. Moreover, OGDC tries to optimize the number of active sensors by reducing the overlapped area between the active sensors. OGDC is carried out after each period during which each sensor decides to be active or passive. The starting node broadcasts a power-on message in a random direction along which active nodes are found. A sensor decides to turn off if it covers an intersection point between two active sensors and if it minimizes the overlapped area with active sensors. However, the authors have not shown how is chosen the sensor that should initiate the sensor-scheduling process? And, how could we solve the problem of conflict when several sensors simultaneously initiate the sensor-scheduling process? Moreover, the technique used to choose the active nodes do not take into account the ability of a sensor to perform area coverage, so it may choose nodes with low remaining energy as active nodes. In addition, selecting active nodes phase generates high latency since only one sensor is chosen to initiate the sensor-scheduling process.

All works presented above deal with the 1-coverage problem in WSNs. However, if the monitored area presents some important regions which require a permanent monitoring, it is thus essential to observe these regions by several sensors to achieve fault tolerance caused by sensors failure. In this case, it is the k-coverage problem.

3.2. k-coverage

In most works, k-coverage problem consists to find a minimal set of active nodes where every position within monitored area is covered by at least k sensors. Accordingly, several approaches were proposed in the literature. In [13,14], the authors choose a random subset of active nodes to maintain area coverage while reducing energy consumption. However, with this technique, it is difficult to be sure that the monitored area is fully covered. In [15,16], the authors proposed distributed cluster-based protocols to maintain area k-coverage. However, the overhead generated by these protocols is very high.

Huang and Tseng [17] have proposed solutions to deal with k-coverage problem. These solutions are based on checking the perimeter of each sensor’s sensing range where the sensing area of each sensor is modelled by a unit disk. The authors proved that the area monitored is fully k-covered if the perimeter of the sensing area of each sensor is k-covered. However, the running time of the algorithm is $\theta(n^2 * \log n)$ in the worst case for a set of n sensors. In [18], So and Ye proposed an improved modelling. They used the concept of order k Voronoï diagrams [19] to build a verifier algorithm. They proved that if all vertices of a bounded Voronoï diagram are sufficiently covered then the whole area is covered. The running time of the algorithm is bounded by the construction time of the Voronoï diagram which is $\theta(n * \log n + n * k^2)$ [20]. These solutions did not address the k-coverage problem and they did not propose distributed algorithms.

In [21], Tian and Georganas have proposed an extension of the algorithm presented in [11] dealing with preserving k-coverage problem in wireless sensor networks. This algorithm uses a sensor-scheduling scheme to guarantee that the level of coverage of the monitored area after turning off some redundant sensors remains the same. Hence, if there are more sensors than necessary, we may turn off some redundant nodes to save energy. These sensors may be turned on later when other sensors exhaust their energy. However, this extension requires a great number of control messages.

Zhou et al. [22] presented a greedy approach to solve the area coverage problem using connected sets. The greedy approach consists to find a set of sensors called *M* to ensure the area coverage so that the communication graph induced by *M* is connected. Each point in the monitored area is covered by at least *k* sensors in *M*. However, the complexity of this greedy algorithm is very high in terms of messages exchanged between sensors to construct the set *M*. Moreover, the messages exchanged may be corrupt and their sizes are potentially large.

In [23], Wang et al. have proposed a distributed configuration coverage protocol (CCP). CCP could provide various degrees of coverage required by some applications and maintain network connectivity at the same time. CCP is based on the *k*-coverage eligibility algorithm which is performed by each sensor to decide whether it should become active or not [23]. However, CCP requires that sensors known their positions and the sensor-scheduling mechanism used in CCP do not ensure that the number of active sensors is minimal. The complexity of the k-coverage eligibility algorithm is $\theta(d^3)$, where *d* is the number of nodes in the largest sensing neighbor set.

In [5], Yang et al. jointly addressed k-coverage problem (k-CS) and k-connected coverage problem (k-CCS) using the technique of integer programming (IP) and linear programming based on an approximation algorithm (LPA). The complexity of LPA is very high because it is dominated by the LP solver. The best performance of LPA is $\theta(d^3)$ using Ye's algorithm [24], where *n* is the number of variables. To overcome this limitation, Yang et al. have proposed a quasi-local approach called CKA (Cluster-Based k-CCS/k-CS Algorithm). CKA is a cluster-based algorithm which requires *k* iterations to select the active sensors. After each iteration, the cluster-heads selected are marked and removed from the network. These cluster-heads should be connected by gateway nodes which are also marked. However, the complexity of the proposed solution was still always high. It is $\theta(k*\log^3 n)$. To improve the performance of CKA in terms of complexity, Yang et al. have proposed a local solution based on local information in 2-neighborhood, called PKA (Pruning-based k-CS/k-CCS Algorithm). In this algorithm, the authors have involved a priority that has an abstract aspect for selecting active sensors that should guarantee the area coverage.

In [25], the authors have presented a protocol based on the dominating set to maintain area coverage, called ADS (Area Dominating Set Protocol). ADS is an improved version of DS and CDS that provide the node coverage. It consists to select a minimum number of sensors to ensure simultaneously the area coverage and network connectivity while saving energy. The complexity of the coverage algorithm used by ADS is $\theta(d^3)$ because this algorithm is based on the CDS construction algorithm whose complexity is $\theta(d^2)$.

4. Cluster-based Scheduling Algorithm

In the purpose to ensure the area coverage while prolonging network lifetime, we proposed a cluster-based distributed scheme called CSA_VS (Cluster-based Scheduling Algorithm – Virtual Sensor), which is used to allow each sensor to switch between active and sleep modes to save energy. Sensors are assumed that they have limited battery energy. Sensing, transmitting and receiving activities consume battery energy of a sensor, and thus limit the network lifetime. In our work, we determine the statute of all the deployed sensors to be either active or sleep based on their capabilities as well as the state durations, such that the network lifetime is maximized. In our context, the network lifetime is defined as the time duration starting from network set up to the time when the level of area coverage is lower than a certain threshold (90%).

CSA_VS selects the active sensors after the election of cluster-heads and the formation of clusters following the algorithm of self-organization presented in [4]. In this algorithm, we addressed the problem of node coverage while in CSA_VS we generalize this aspect and address the problem of area coverage. Furthermore, sensors that provide the area coverage would be chosen with a distributed manner and according to their weight by their corresponding cluster-heads. The weight of each sensor is a combination of the following parameters: k-density and residual energy, as presented in (Eq. 1). We used the k-density with k=2 not to weaken our algorithm of its performance and not to increase the overhead. The 2-density of a node *u* represents the ratio between the number of links in its 2-hop neighborhood (links between *u* and its neighbors and links between two 2-hop neighbors of *u*) and 2-degree of *u*. We used the 2-density as parameter instead of 2-degree to generate homogeneous clusters and to favour the node that has the most 2-neighbors related to become cluster-head. The coefficient of each parameter can be chosen depending on the application. For example, in an application where the energy is critical we choose the sensor that has more energy to be cluster-head. Therefore, we assign to β a big enough value relatively to α. In our contribution, we attribute adequate values to the various coefficients in the purpose to generate stable clusters (α=0.5, β=0.5).

$$Weight(u) = \alpha * 2\text{–}density(u) + \beta * Energy(u) \tag{1}$$

With $\alpha + \beta = 1$

CSA_VS is performed in two phases. The first phase is performed according to the following algorithmic scheme:

- C_i: the number of the cluster (i),
- $N_2(CH, C_i)$: the set of the cluster-head CH's 2-neighbors belonging to the cluster C_i,
- $N_1(CH, C_i)$: the set of the cluster-head CH's 1-neighbors belonging to the cluster C_i,
- $Cover(C_i)$: the set of sensors in the cluster C_i that are placed in active mode to maintain the coverage area.
- A sensor switches to active mode according to its weight.

Pseudo-code of the algorithm CSA_VS

For each cluster-head CH **do**
- $N_2'(CH, C_i) = N_2(CH, C_i)$
- $Cover(C_i) = \{CH\}$

End for

/* Switch to active mode the isolated 2-neighbors of CH and the CH's 1-neighbors that can cover these nodes

While $\exists v \mid v \in N_2'(CH, C_i) \wedge \exists u \in N_1(CH, C_i) \mid v \in N_1(u, C_i)$ **do**
- $Cover(C_i) = Cover(C_i) \cup \{u,v\}$
- $N_2'(CH, C_i) = N_2'(CH, C_i)/\{v\}$

End while

While $N_2'(CH, C_i) \neq \emptyset$ **do**
- **Choose** $u \in N_1(CH, C_i)$:
    - o $Weight(u) = Max(Weight(u_i): u_i \in N_1(CH, C_i))$

- $Cover(C_i) = Cover(C_i) \cup \{u\}$
- **Choose** $v \in N_1(u, C_i)$:
  - $Weight(v) = Max(Weight(v_i): v_i \in N_1(u, C_i) \wedge v_i \notin N_1(CH, C_i))$

For each $w \in N_1(u) \wedge w \in N_1(v)$ **do**

$Cover(C_i) = Cover(C_i) \cup \{w\}$

End for

- $N_2'(CH, C_i) = N_2'(CH, C_i) / N_1(u, C_i)$

End while

In the second phase, each active sensor classifies its 1-neighbors in ascending order of their weight. Then it checks the number of active sensors within its sensing area. Let *t* be the number of sensors in u's sensing area. If *t* is less than the coverage degree *k*, *u* would switch to active mode *(k-t)* sensors with the greatest weight in its 1-neighborhood. Then, *u* would pass in idle mode *k/2* sleeping sensors that have the greatest weight in its 1-neighborhood. These sensors could replace active sensors which may cease to operate before the expiry of the period T_{active} of their activities. Moreover, once clusters are formed and active sensors are selected, the data communication phase begins where the active sensors periodically collect data and send it to their corresponding cluster-head. The cluster-head nodes aggregate the data from the cluster memberships and route the aggregated data packets over the pre-determined multi-hop paths to the sink.

4.1. Virtual sensor and discretization of the monitored area

The k-coverage problem is considered as a NP-hard problem. A first solution consists to find all regions shared by a certain number of sensors and verify that each region is covered by at least *k* sensors. However, geometric verification of all regions covered by a number of sensors is a complex task because there may exist many regions shared by several sensors whose number may reach in the worst case $\theta(n^2)$. Moreover, it might be difficult to determine these regions.

Our solution consists to discretize the area monitored in several regions and to choose a random position within every region. To verify the k-coverage of every chosen position, we suppose that the latter is a virtual sensor that can exchange the beacons or Hello messages with its neighborhood to know its degree and therefore the number of sensors that covers it. This number represents the coverage degree of the position.

The algorithm associated to this approach is performed as follows:

1. Discretize the area monitored in several squared regions. Let *m* the number of these regions.
2. Generate a random point within every created region: by generation of random points using a uniform distribution function such as the generated points are uniformly distributed in the monitored area and their positions don't coincide with the already existing sensors or the already generated points.

Network Protocols and Algorithms
ISSN 1943-3581
2010, Vol. 2, No. 2

3. Calculate the coverage degree of every virtual sensor generated. For that, it is sufficient to calculate the Euclidean distance between this sensor and every sensor deployed within the monitored area.
4. Calculate the degree of area coverage: the virtual degree of the area is equal to the minimal degree of the generated virtual sensors.

Formally, let v_i, i=1,…, m the generated virtual sensors and $\delta_1(v_i)$ their corresponding degrees. The k-coverage of the area A, noted Cover(A) is equal to the minimal degree of all virtual sensors. If Cover(A) is equal to zero (Cover(A) =0), then some holes exist within the area monitored and in this case the area coverage ratio is less than 100%.

1. For each $v_i(x_i,y_i)$ i=1,…, m: virtual sensors
   - Calculate $\delta_1(v_i) = |N_1(v_i)|$ where $|N_1(v_i)|$ is the number of v_i's active neighbors.
2. Calculate the area k-coverage Cover(A)
   - $Cover(A) = Min\{\delta_1(v_i): i = 1, ..., m\}$
3. If Cover(A) is equal to zero, then some holes exist within the monitored area A, otherwise the degree of the area coverage is Cover(A).

Particularly, to check the area 1-coverage, it suffices that there is at least one active sensor in the 1-neighborhood of each virtual sensor.

4.2. *Evaluation of the coverage ratio*

To calculate the coverage ratio θ(Cover(A)) of a specific area A, we decompose the area A in several regions and we generate random points within each region. Let m the number of points generated and $Ref(v_i)$ the set of these points. These generated points represent virtual sensors. Then, we determine the number of virtual sensors that have at least one active sensor in its neighborhood $\{v_i: \delta_1(v_i) > 0\}$. Thus, the coverage ratio of the area A is the ratio of the number of virtual sensors that have at least one active neighbor and the number of virtual sensors $(Ref(v_i))$ (Eq. 2):

$$\theta(Cover(A)) = \frac{|\{v_i : i=1,\ldots,m / \delta_1(v_i) > 0\}|}{|Ref(v_i)|} \qquad (2)$$

The precision of coverage ratio θ(Cover(A)) depends on the number of generated points $(Ref(v_i))$ and of their positions.

Moreover, the evaluation of the area coverage ratio permits to improve CSA_VS performance and to overcome its limitations. Thus, if a virtual sensor doesn't have any active neighbor, it performs its maintenance phase to improve the coverage ratio of the monitored area. For that, it activates the sensor that has the greatest weight among its neighbors which are in idle mode. We can generalize this process to ensure area k-coverage.

5. Performance evaluation

In this section, we evaluated the ratio of the area coverage by CSA_VS for two degrees of coverage: 1-coverage and 2-coverage. Then, we compared the performance of CSA_VS to those algorithms LPA, PKA, and CKA presented in [5] in terms of number and percentage of active sensors.

Table 1. Simulation parameters

Parameters	Values
Surface of the monitored region	100m x 100m
Number of sensors	100, 200, 300, 400, 500
Sensing range (R_s)	20m, 40m

To evaluate the coverage ratio and the percentage of active sensors, we used a simulator written in C++, that we developed to avoid the noise caused by the other simulators because we assumed that communications are reliable and there are no corrupted messages.

5.1. Context of the simulation evaluation

To illustrate the impact of the density on the number and percentage of active sensors, we considered several network topologies. Each one includes n sensors having the same communication range R_c and the same sensing range R_S as $R_c = 2 * R_S$. Sensors are placed randomly in a 100m × 100m square area following a uniform distribution function, and the base station is placed outside the monitored area. Moreover, to show the effect of the link density on CSA_VS's performance, we used two distinct sensing ranges 20m and 40m. For each configuration, the simulations are repeated 100 times to calculate the average value of each performance criterion. Table 1 summarizes the simulation parameters used.

5.2. Results analysis

In this section, we present and we analyze the performance of the algorithm CSA_VS.

5.2.1. Coverage ratio versus number of deployed sensors

To determine the ratio of the area coverage guaranteed by CSA_VS, we have randomly generated 100 reference positions within the 100 created regions and we checked if they are covered or not. Therefore, we used several network topologies with distinct densities to illustrate the impact of density on the ratio coverage. Moreover, we used two distinct sensing ranges to illustrate the effect of the link density.

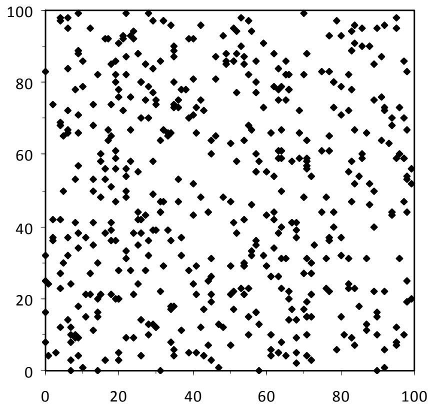


Figure 1. An example of random deployment of 500 sensors

Figure 1 shows an example of deployment of 500 random sensors in a *100m* × *100m* square area.

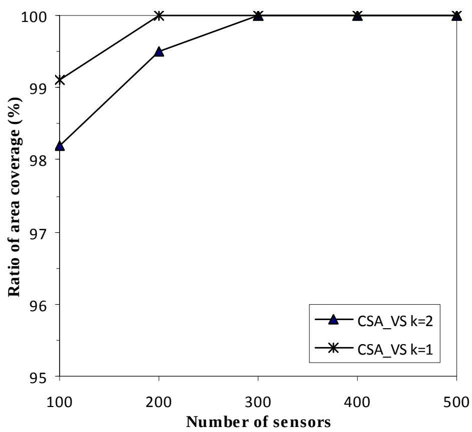
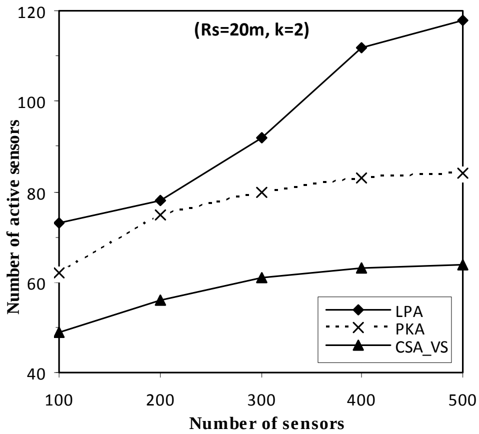


Figure 2. Coverage ratio

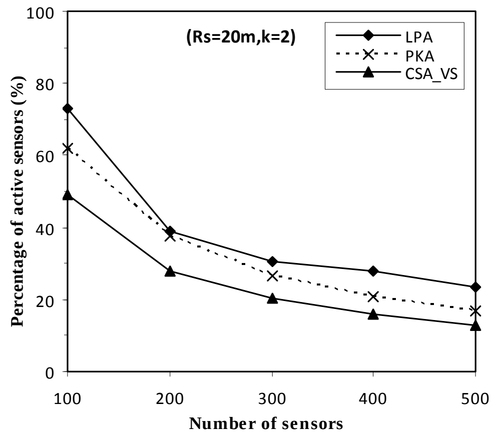
Figure 2 shows that the 1-coverage ratio provided by CSA_VS is 99, 1% when the number of sensors is 100 if the sensing range is 20m, and it reaches 100% (full coverage of the monitored area) when the number of sensors exceeds 100. However, the 2-coverage ratio provided by CSA_VS is 98, 2% when the number of sensors is 100 and it reaches 100% when the number of sensors exceeds 300. Besides, we noticed that the 1-coverage ratio and the 2-coverage ratio ensured by CSA_VS is always 100% when the sensing range is 40m.

5.2.2. Evaluation of the coverage degree

To determine the number of active sensors and the coverage degree k versus the sensing range, we have even used two distinct sensing ranges R_s = 20m and R_s = 40m to illustrate the impact of the link density on the performance of CSA_VS. We have evaluated the coverage degrees for k = 2 and k = 3.

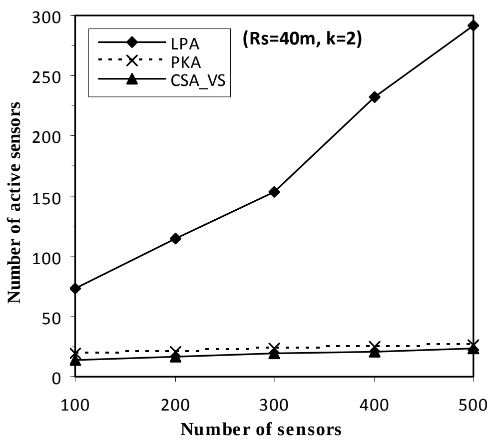


(a) Figure 3. Evaluation of the number of active sensors

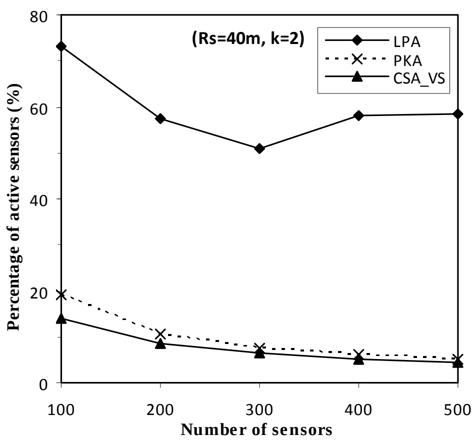


(b) Figure 3. Evaluation of the percentage of active sensors

Figures 3(a) and 3(b) compare respectively the number and the percentage of active sensors to ensure the full 2-coverage of the monitored area when the sensing range is 20m. We noted that CSA_VS implies a minimal number of sensors to ensure the full area 2-coverage compared to PKA and LPA. Moreover, CSA_VS provides better results in terms of percentage of active sensors. Indeed, when the number of sensors deployed increases, the percentage of active sensors greatly decreases. On the other hand, LPA implies more that 50% of deployed sensors for area 2-coverage.

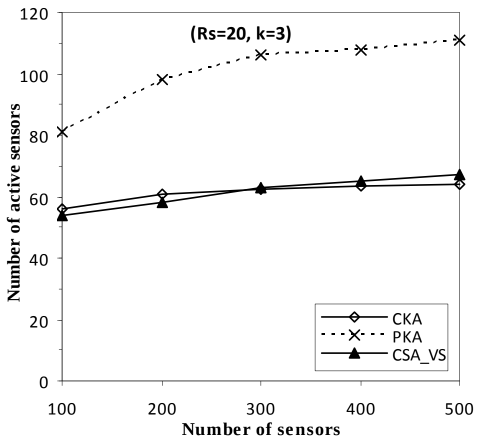


(a) Figure 4. Evaluation of the number of active sensors

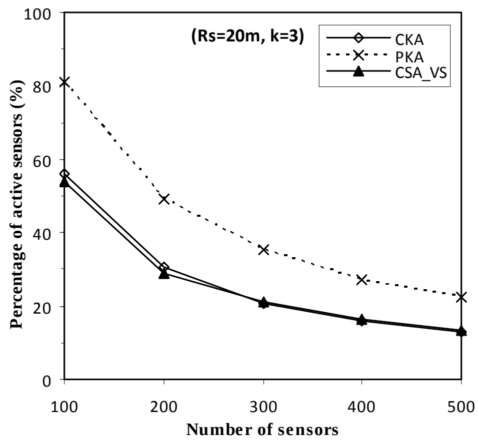


(b) Figure 4. Evaluation of the percentage of active sensors

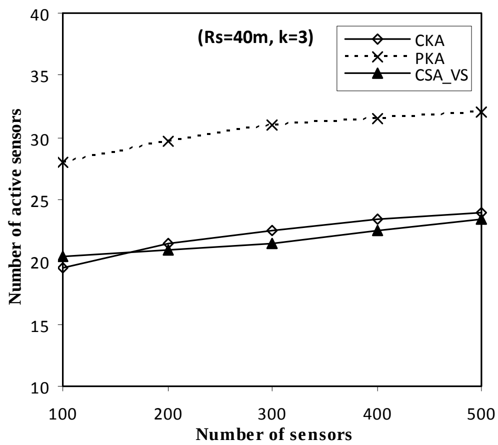
Moreover, Figures 3(a), 3(b), 4(a) and 4(b) show the effect of the link density on the number of active sensors. Thus, when the sensing range of sensors increases the number of active sensors decreases except in LPA which is centralized. These results also show that the quasi-local aspect of CSA_VS has good effects on its performance. On the other hand, the high network density has a negative impact on the performance of LPA because when the density increases the maximum degree of a node increases which will affect the ratio $1/(1+\Delta)$ where Δ is the maximum network degree and therefore the cardinality of the set containing the active nodes increases.



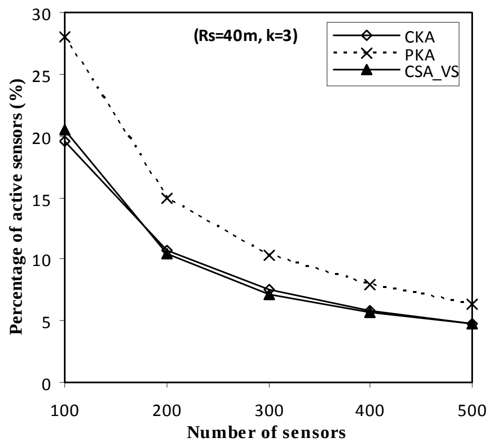
(a) Figure 5. Evaluation of the number of active sensors



(b) Figure 5. Evaluation of the percentage of active sensors



(a) Figure 6. Evaluation of the number of active sensors



(b) Figure 6. Evaluation of the percentage of active sensors

In the figures 5(a), 5(b), 6(a) and 6(b), we compared CSA_VS to PKA and CKA which has the same aspect as CSA_VS. The results showed that CSA_VS provides better results than CKA for a high degree of coverage in terms of number of active nodes when the number of nodes in the network is less than 300 while CKA slightly exceeds CSA_SV when the network becomes dense. However, the authors in [5] have not checked that the coverage ratio is 100%, unlike CSA_VS that provides the full coverage of the monitored area with the results presented in these various figures.

6. Conclusion and future work

In this paper, we addressed the k-coverage problem because in some applications, it is possible that some locations called sensitive regions in the monitored area are more important than others and need to be covered by more sensors to achieve fault tolerance and to deal with erroneous measurements collected by the sensors. The solution proposed can test whether a point within the monitored area is k-covered or not. To check k-coverage of this point, we apply the algorithm CSA_VS and verify if each virtual sensor has at least k active sensors in its neighborhood.

The work presented in this paper has helped to ensure full coverage of the monitored area, involving a minimum number of sensors. As a result, energy consumption is minimized and therefore network lifetime will be extended. To demonstrate the performance of the algorithm, we have compared the results obtained by this algorithm to the results provided by other efficient algorithms described in the literature. We have shown that CSA_VS implies a lower number of active sensors for area coverage compared to LPA, PKA, and CKA. Thus, the quasi-local aspect of CSA_VS and the periodic selection of active sensors according to the k-density and the remaining energy of sensors enabled CSA_VS to provide good results.

In future work, we propose to deal with the coverage problem in a mobile environment and to study the complexity of our contributions and compare it to those of the other protocols. Besides, we quantify the energy consumption and compare it to those of the other protocols.

References

[1] Akyildiz I.F., Su W., Sankarasubramaniam Y., Cayirci E.,"Wireless sensor networks: a survey". Computer Networks (Elsevier). Vol. 38, Issue 4. Pp 393-422. March 2002.

[2] Cardei M., Thai M.T., Li Y. and Wu W., "Energy-efficient target coverage in wireless sensor networks". International IEEE Conference on INFOCOM, Pp 1976-1984, 2005.

[3] Cardei M., Wu J., Lu M. and Pervaiz M.O., "Maximum network lifetime in wireless sensor networks with adjustable sensing ranges", International IEEE Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2005.

[4] Lehsaini M., Guyennet H., Feham M., "α-coverage Scheme for Wireless Sensor Networks", The Fourth IEEE International Conference on Wireless and Mobile Communications (ICWNC'08), Pp 91-96, Athens, Greece, July 2008

[5] Yang S., Dai F., Cardei M., Wu J., "On Connected Multiple Point Coverage in Wireless Sensor Networks". International Journal of Wireless Information Networks. Vol. 13, Issue 4. Pp 286-301. October 2006.

[6] Ye F., Zhong G., Cheng J., Lu S. and Zhang L., "Peas : A robust energy conserving protocol for long-lived sensor networks", International Conference on Distributed Computing Systems (ICDCS03). Pp 28-37, Rhode Island, USA, 2003.

[7] Gui C. and Mohapatra P., "Power conservation and quality of surveillance in target tracking sensor networks", International Conference of ACM Mobile Computing and networking (Mobicom). Pp 129-143, Philadelphia, PA, USA, 2004.

[8] Cai Y., Li M., Shu W. and Wu M.Y., "Acos : A precise energy-aware coverage control protocol for wireless sensor networks". International Journal Ad Hoc Sensor Wireless Networks. Vol. 3, Issue 1. Pp 77-98. 2007.

[9] Hsin C.F. and Liu M., "Network coverage using low duty-cycled sensors: random

coordinated sleep algorithms". The 3rd International Symposium on Information Processing in Sensor Networks (IPSN04), Pp 433-442, Berkeley, CA, USA, 2004.

[10] Sheu J.P., Yu C.H. and Tu S.C., "A distributed protocol for query execution in sensor networks". International IEEE Conference on Wireless Communications and Networking (WCN05), Pp 1824-1829, New Orleans, LA, USA, 2005.

[11] Tian D. and Georganas N., "A coverage-preserving node scheduling scheme for large wireless sensor networks". International ACM Conference on Wireless Sensor Networks and Applications (WSNA02), Pp 32-41, Atlanta, GA, USA, 2002.

[12] Zhang H. and Hou J., "Maintaining sensing coverage and connectivity in large sensor networks". International Journal Wireless Ad Hoc and Sensor Networks. Vol. 1, Issue 1-2. Pp 89-123. 2005.

[13] Cardei M., MacCallum D., Cheng X., Min M., Jia X., Li D., Du D.Z.. Wireless Sensor Networks with Energy Efficient Organization. International Journal of Interconnection Networks. Vol. 3, Issue 3-4. Pp 213-229. 2002.

[14] Slijepcevic S. and Potkonjak M., "Power Efficient Organization of Wireless Sensor Networks". International IEEE Conference on Communications. Vol. 2, Pp 472-476, 2001.

[15] Abrams Z., Goel A. and Plotkin S., "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks". International Symposium on Information Processing in Sensor Networks, Pp 424-432, New York, NY, USA, 2004.

[16] Yang S., Dai F., Cardei M. and Wu J., "On multiple point coverage in wireless sensor networks". International IEEE Conference on Mobile Ad hoc and Sensor Systems, 2005.

[17] Huang Y. and Tseng Y., "The coverage problem in a wireless sensor network". International ACM Conference on Wireless Sensor Networks and Applications, San Diego, CA, September 2003.

[18] So A.M.C. and Ye Y., "On Solving Coverage Problems in a Wireless Sensor Network Using Voronoi Diagrams". Workshop on Internet and Network Economics (WINE 2005), Pp 584-593, Hong Kong, December 2005.

[19] Gosnell T.B., Hall J.M., Ham C.L., Knapp D.A., Koenig Z.M., Luke S.J., Pohl B.A., Schach von Wttenau A. and Wolford J.K., "Gamma-Ray Identification of Nuclear Weapon Materials". Technical Report DE97053424. Lawrence Livermore National Lab., Livermore, CA (USA). February 1997.

[20] Dai F. and Wu J., "On Constructing k-Connected k-Dominating Set in Wireless Ad Hoc and Sensor Networks". International Journal of Parallel and Distributed Computing. Vol. 66, Issue 7. Pp 947-958, July 2006.

[21] Tian D. and Georganas N.D., "Connectivity maintenance and coverage preservation in wireless sensor networks". AdHoc Networks Journal (Elsevier Science). Pp 744-761, 2005.

[22] Zhou Z., Das S. and Gupta H. "Connected k-coverage problem in sensor networks". The 13th International Conference on Computer Communications and Networks (ICCCN04), Pp 373-378, Chicago, USA, 2004.

[23] Wang X., Xing G., Zhang Y., Lu C., Pless R. and Gill C., "Integrated coverage and connectivity configuration in wireless sensor networks". The first ACM International Conference on Embedded Networked Sensor Systems, Pp 28-39, Los Angeles, USA, 2003.

[24] Ye Y., "o(n3l) potential reduction algorithm for linear programming". Mathematical

Programming. Vol. 50, Issue 2. Pp 239-258, 1991.
[25] Carle J. and Simplot-Ryl D., "Energy efficient area monitoring for sensor networks". IEEE Computer Society Magazine. Vol. 37, Issue 2. Pp 40-46, February 2004.