Data gathering architecture for temporary worksites based on a uniform deployment of wireless sensors

Ines Khoufi, Saoucene Mahfoudh, Pascale Minet, Anis Laouiti

Data gathering architecture for temporary worksites based on a uniform deployment of wireless sensors

Ines Khoufi

INRIA, Rocquencourt,
78153 Le Chesnay Cedex, France,
Email: ines.khoufi@inria.fr

Saoucene Mahfoudh

INRIA, Rocquencourt,
78153 Le Chesnay Cedex, France,
Email: saoucene.mahfoudh@inria.fr

Pascale Minet

INRIA, Rocquencourt,
78153 Le Chesnay Cedex, France,
Email:pascale.minet@inria.fr

Anis Laouiti

TELECOM SudParis,
CNRS Samovar UMR 5157,
91011 Evry Cedex, France,
Email: anis.laouiti@telecom-sudparis.eu

Abstract: Data management is an important issue in wireless sensor networks. However, the data gathering process may be unsuccessful in disconnected or random networks: some data gathered cannot be delivered to the sink, coverage holes may occur and result in missing data. To cope with this problem, we propose two distributed redeployment algorithms (DVFA and ADVFA), based on virtual forces, that provide a uniform deployment of sensor nodes. Both ensure a full area coverage and network connectivity. Hence, an accurate data gathering can be done. For that purpose, we define a three-tier architecture where after the execution of a redeployment algorithm, sensor nodes are uniformly deployed to monitor a temporary worksite. A uniform deployment provides a better balancing of the routing tree (used to report the data gathered) leading to smaller data gathering delays. We also show how to save energy during the data gathering phase. We compute the optimal number of sensors needed to cover an area. Using this result, we parameterize DVFA and evaluate its performances. We then propose ADVFA to cope with DVFA drawbacks (e.g. node oscillations greedy in energy). ADVFA adapts the target distance between two neighbors to the number of operational nodes discovered. ADVFA outperforms DVFA, considerably reducing the distance traveled by nodes and then maximizing network lifetime by saving energy.

Keywords: WSN; wireless sensor network; data gathering; mobile sensors; deployment; three-tier architecture; coverage; connectivity; adaptivity; convergecast.

Biographical notes: Ines Khoufi received her Computer Science Engineering and Master degrees from the National College of Computer Science (ENSI) in 2010 and 2011 respectively. Currently, she is a Ph.D. student in HIPERCOM2 research team at Inria. Her current research interest is on wireless sensor networks deployment and progressive discovery of an unfriendly environment.

Saoucene MAHFOUDH received her Computer Science Engineering degree from ENSI in 2005 and her Master degree from the University of Paris 6 in 2006. She obtained her Ph D diploma in 2010 from the University of Paris 6. Her research topics deal with energy efficiency, cross-layering, routing and redeployment in wireless sensor networks. After a post-doctoral fellow at Inria, she has been working at King AbdelAziz University since 2013.

Pascale MINET works at the Inria research center of Rocquencourt, near Versailles. She is head of the HIPERCOM (High Performance Communication) team. She got her qualification in advising PhD students in 1998 from the University of Versailles. Previously she got her PhD diploma in Computer Science, in 1982 from the University of Toulouse and her Engineer diploma in Computer Science in 1980 from ENSEEIHT (Engineering school of Toulouse). Her research topics relate to wireless sensor networks and mobile ad hoc networks and more particularly energy

efficiency, routing, node activity scheduling, multichannel communication, redeployment and quality of service in these networks. She is co-author of the OLSR routing protocol standardized at IETF.

Anis Laouiti is an associate professor at Telecom SudParis since 2006. Before, he did his Phd research work and worked as a research engineer within Hipercom team at Inria-Rocquencourt where he participated to the OLSR routing protocol design (RFC3626). His research covers different aspects in wireless ad hoc and mesh networks including protocol design, performance evaluation and implementation testbed.

1 Introduction

Wireless Sensor Networks (WSNs) constitute an emergent technology that has caught the interest of many researchers in the last few years. WSN has a wide range of application domains e.g. environmental applications, reasonable use of fertilizers and pesticides in precision agriculture, pollutant detection in temporary industrial worksites, mine clearance of an area, exploration of natural resources (e.g. oil) and geological scanning (1), to name a few. A WSN is a wireless network consisting of a set of static or mobile sensors scattered in an area of interest to monitor physical or environmental conditions. The applications given previously, usually require not only to collect data related to the area studied but also to ensure the time and space consistency of these data. Indeed, only periodic measurements taken simultaneously in different points uniformly distributed in the area studied allow an adequate monitoring of the phenomenon observed.

In many applications, sensors are deployed randomly in a specific area without any strategy with regard to their initial positions. This random deployment results in an area where some regions are highly covered while others have just few scattered sensors. As a result, many regions of the deployed area cannot be monitored. Consequently, a redeployment algorithm is necessary to place sensors in appropriate positions to ensure the full area coverage in order to detect each event occurring in this area.

However, ensuring the full area coverage is not sufficient: each event detected should be reported to the sink. As a consequence, network connectivity is required. It has been proved that the full area coverage implies network connectivity under certain conditions binding the transmission range and the sensing range (7). In this paper, we focus on uniform deployment that ensures full area coverage with the minimum number of sensors, taking advantage of this property.

During the initial deployment, some nodes may be damaged and fail. Hence, the number of operational nodes may differ from the total number of nodes that are for instance dropped from a helicopter. If only 70% of nodes are operational, the deployment computed with 100% of nodes will fail to achieve a good coverage: data corresponding to coverage holes will be missing. Furthermore, in a real environment, the direct communication between two nodes may be impossible, even if their distance is smaller than the radio transmission range. This can be due for instance to obstacles limiting the radio propagation. To meet the requirements of applications requiring full and accurate data gathering, these problems must be solved. That is the purpose of the two redeployment protocols presented in this paper. They have the advantage of checking radio links between nodes without relying on the implicit assumption of the unit disk graph model. Both discover dynamically the operational nodes in the network. One of them is also able to adapt its parameters (i.e. the target distance between nodes) to the real number of operational nodes discovered.

In this paper, we consider a network of mobile sensor nodes that must achieve firstly a uniform deployment in a given area during the redeployment phase because of the above explained reasons. Once they are deployed, they start a second phase of data gathering.

Data gathering may also fail because of energy depleted nodes. During the redeployment, we notice that the main source of energy consumption is due to the node moves. Radio communication represents only a small part. That is why we want to optimize nodes moves during the redeployment, by minimizing the total distance traveled by mobile nodes. During the data gathering, nodes are stopped and the main energy consumption is due to communication. Energy efficient techniques are used to save energy. Furthermore, nodes with low residual energy should be replaced by redundant nodes before failure. Otherwise, a global redeployment should be performed.

The focus will be put on the study of a uniform deployment to ensure the full area coverage, which is the key for an accurate data gathering. This paper is organized as follows: In Section 2, we first propose a three-tier architecture to collect information produced by sensor nodes uniformly deployed. Then, we give a state of the art on the redeployment, we position our contribution and define the redeployment problem. In Section 3, we compute the optimal number of sensors required to fully cover a 2-dimension area. In Section 4, we give the principles and parameters used in our performance evaluation. In Section 5, we apply the results found in Section 3 to a distributed uniform redeployment algorithm, DVFA, based on virtual forces and evaluate its performances. In Section 6, we show how to reduce node oscillations, with the adaptive algorithm called ADVFA. A comparative performance evaluation between DVFA and ADVFA is given. We also show how ADVFA copes with positioning errors. Then, the global system functioning is presented in Section 8. Finally, we conclude in Section 9 and list some perspectives.

2 Overview

In this section we first propose an architecture for data gathering in temporary worksites. We then present different

WSNs architectures existing in the state of the art. We list the main constraints that must be taken into account in data gathering and discuss the adequacy of the architectures previously described. We give a state of the art related to redeployment strategies ensuring full area coverage. We position our contribution and present our problem statement.

2.1 The architecture proposed

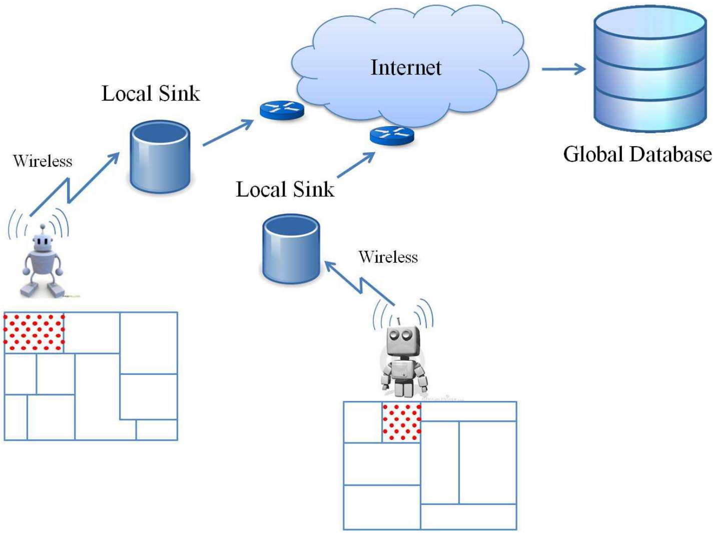


Figure 1: the three-tier architecture proposed.

Figure 1 depicts a three-tier architecture for data gathering supported by mobile wireless sensor nodes. Such an architecture can be used to gather data in temporary worksites (represented by small rectangles in Figure 1). Measurements should be done simultaneously in different points uniformly distributed in the worksite monitored (see red points in Figure 1). The next day, the worksite is moved and a new deployment of the sensor nodes is done in the new area to explore (e.g neighboring rectangles in Figure 1). The goal is to build a precise cartography of a given zone. Examples of such applications are natural resource exploration (e.g. oil), geological study, damage assessment after a disaster, to name a few. In this architecture, we distinguish:

- Mobile sensors which are illustrated by small red points in the temporary worksite (see first rectangle in the left side of Figure 1). These sensor nodes are uniformly deployed using a redeployment algorithm (e.g. ADVFA/DVFA in our work). Mobile sensors are in charge of sensing each event occurring in the worksite considered. They communicate using a wireless medium.

- A mobile controller robot (see the robot in Figure 1) has the task of deciding which worksite will be monitored and notifying the sensor nodes. This controller robot is also in charge of collecting the information produced by sensor nodes in the monitored worksite via a WSN and reporting it to the local sink via a wireless technology (e.g. WiFi, 3G, 4G...). Notice that the controller robot is within the worksite being monitored and it should move to each new monitored worksite.
  A local sink receives the data gathered by the controller robot. Then, it processes this information and sends it to the global database via Internet for example. Notice that the local sink and the controller robot closely interact. That is why they are considered as one tier of the architecture.

- The global database receives the information from each local sink. The global database is in charge of storing, processing and exploiting the data corresponding to the whole area consisting of several worksites.

This architecture supports the following steps:

- (Re)deployment of mobile sensor nodes in a temporary worksite with DVFA or ADVFA (see Section 6) to obtain a uniform deployment. Nodes stop moving when this uniform deployment is achieved.

- Building of the routing tree rooted at the controller robot (see Section 8). As a consequence, each sensor node has a path to reach the controller robot. Notice that the uniform deployment of sensor nodes and the central position of the controller robot will produce a better balanced tree, leading to shorter data gathering delays (see (32) for a proof).

- During the data gathering, we use optimized techniques to minimize the energy consumed by communication (see Section 8). Data collected by the sensor nodes, from the worksite, are delivered to the controller robot, transferred to the local sink via a wireless network and then to the global database via Internet, where they are exploited.

2.2 State of the art of data gathering

The data gathering is an important issue in wireless sensor networks to support monitoring applications. It consists in collecting information (e.g. physical measures) from sensor nodes deployed in the area considered.

2.2.1 Classification of WSN architectures

Various types of wireless sensor network architectures are proposed in the literature in order to perform the data gathering task and meet the monitoring application requirements. We classify these architectures based on the terminology proposed in (2) which distinguishes three types of nodes:

- **Sensor nodes** (mobile/static) in charge of sensing events, reporting data and possibly forwarding data received from other nodes.

- **Sinks** in charge of collecting data from sensor nodes or from special support nodes. A sink can be static or mobile.

- **Special support nodes** that relay data from sensor nodes to the sink. They can be static (e.g. throwboxes, gateways) or can take advantage of their mobility to perform their task (e.g. mobile collectors, data mules, ferries).

We distinguish between the following types of WSN architecture to perform data gathering; architectures 1 and 2 consist of only two types of nodes whereas architectures 3 and 4 use three types of nodes and architecture 5 is based on four types of nodes.

- **Architecture 1**, the simplest one, is only composed of a static sink and sensor nodes. The sink is able to communicate with all the sensor nodes either directly or via multi-hop. See Figure 2a.
- **Architecture 2**, composed of a mobile sink and sensor nodes. The mobile sink visits each node to collect its data. See Figure 2b.
- **Architecture 3**, composed of a static sink, one or multiple mobile collectors or ferries, and sensor nodes. Each mobile collector (See Figure 2c) is in charge of collecting data from some sensor nodes and reporting them to the sink. A ferry is like a mobile collector but it has a fixed trajectory (e.g. (4) and (5)). When sensor nodes are connected to the ferry, they transmit their data. If multiple ferries exist, when two ferries are connected, the nearest one to the sink receives the collected data from the other ferry and forward them to the static sink: see Figure 2d.
- **Architecture 4**, composed of a mobile sink, throwboxes and sensor nodes. The throwbox acts as a cluster head (6). It is connected to a group of nodes to collect their data. The mobile sink is in charge of visiting each throwbox to achieve the data gathering task. See Figure 2e.
- **Architecture 5**, the most complex architecture, is composed of a static sink, ferries or mobile collectors, throwboxes and sensor nodes. The throwboxes perform like in architecture 4, but a mobile collector is in charge of collecting data from throwboxes and forwarding them to the sink. See Figure 2f.

2.2.2 *Data gathering constraints*

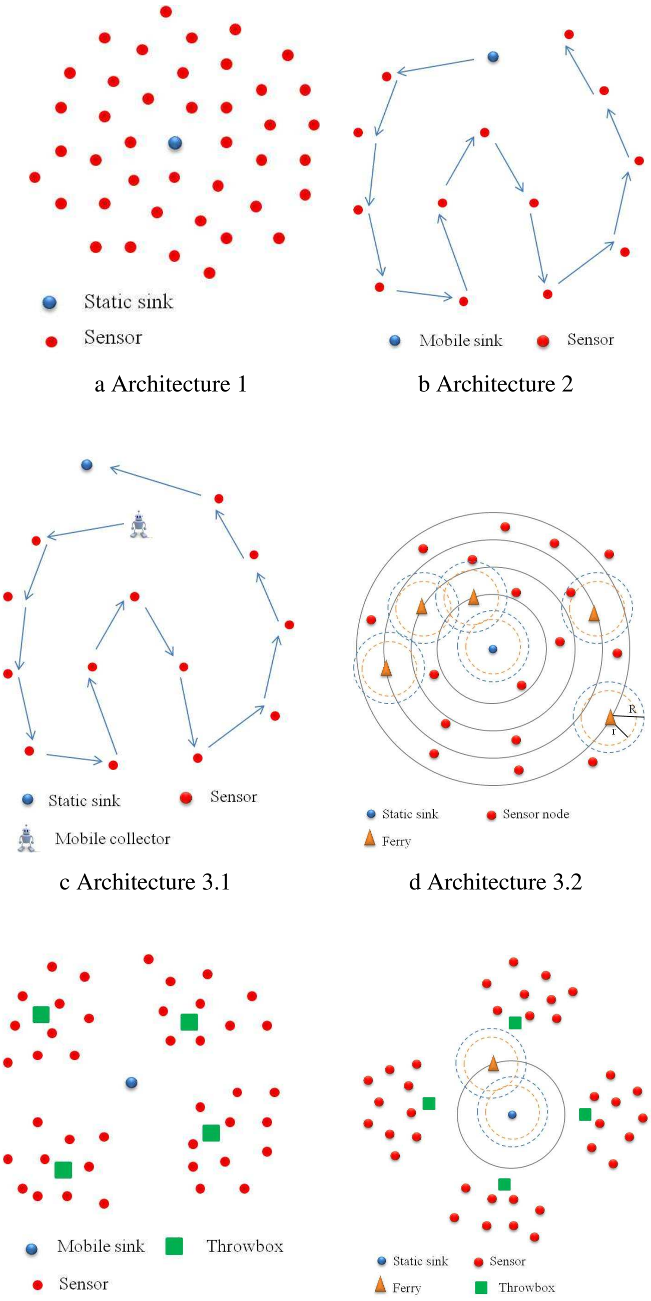
Depending on the application requirements, data gathering must meet different constraints related to:

- Types of nodes

Depending on the budget and constraints of the applications, different types of nodes are available to collect data. They belong to the three types identified in Section 2.2.1: sink, sensor nodes, special support nodes. They can be static (e.g. throwboxes, gateways) or can take advantage of their mobility to perform their task (e.g. mobile collectors, data mules, ferries).

- Sensor deployment and network connectivity

The sensor deployment has an important impact on the process of data gathering. Depending on the application requirements, sensor deployment can be random, uniform, regular, etc. Also, it can provide full or partial coverage. In addition, network connectivity may be maintained or not. If network connectivity is maintained, then there is at least one path from each sensor node to the sink. Hence, all the sensor nodes are able to forward their data to the sink. In this case, the position of the sink may have an impact on the energy consumed by sensor nodes during the data gathering. It is preferred that the sink position be in the center of the area considered to have a tree balanced for data collection. In this case, the load between nodes is more balanced and the data gathering delay is small. On the contrary, when the network is disconnected, the data gathering task can be performed using a mobile sink or mobile collectors to collect data from sensor nodes or from throwboxes which have already collected data from sensor nodes. However, if the sink is static, then mobile collectors, ferries or data mules can be used to report information to the sink.



a Architecture 1 b Architecture 2

c Architecture 3.1 d Architecture 3.2

e Architecture 4 f Architecture 5

Figure 2: WSN Architectures.

- Node's energy

The energy is a constraint in data gathering. Sensor nodes may consume all their energy in reporting and forwarding data to the sink. In some applications, the mobile sink should visit each sensor node to collect data. Then, its energy can be totally consumed due to the great distance traveled. To avoid this problem, the special support nodes (e.g gateway, throwbox, ferry, etc) can be used. Some throwboxes can be deployed to collect data from sensor nodes and then the mobile sink will visit only these throwboxes. The use of ferries can also reduce the energy consumption. The ferry collects data from sensor nodes when there are connected and transmits them to another ferry until reaching the sink. In addition, the aggregation technique can be used to save energy and bandwidth.

- Network bandwidth

The amount of data and control messages must meet the limited bandwidth available in the WSN considered.

- Application requirements

The uniform deployment can be needed in precision agriculture applications where it provides an accurate information of the phenomenon measured and allows the easy determination of the disease propagation in the field. Furthermore, some applications require the time and space consistency of the data collected. In this case, an appropriate deployment should be ensured before performing data gathering. In other words, collecting data with time and space consistency may not be possible without a uniform or regular sensor deployment.

2.2.3 Discussion about architectures with regard to data gathering constraints

Architecture 1 is chosen when all the nodes are static. This architecture requires a sensor deployment guaranteeing connectivity with the sink and area coverage meeting the application requirement. To save energy and to meet short data gathering delays, the sink must be located at the center of the area monitored. The main advantage of this architecture, is its short data gathering delays. However, sensor nodes close to the sink will exhaust their battery quicker than further nodes (3). To alleviate this problem, additional sensor nodes may be deployed around the sink to balance the data forwarding to the sink. Another solution is given by architecture 2.
Architecture 2 is selected when the sink is mobile to maximize network lifetime by saving energy of sensor nodes. The sink mobility ensures an intermittent connectivity. Unlike in the previous architecture, sensor nodes do not forward data received from other sensor nodes. They wait the visit of the mobile sink to send their data. The trajectory of the mobile sink can be optimized with regard to the application requirements. The drawback of this architecture is the time needed by the mobile sink to collect the data. In addition, this architecture is not scalable.
Architecture 3 is adopted when the sink is static and the connectivity is not ensured. Then, mobile collectors are in charge of collecting data from sensor nodes and bringing them to the sink. Hence they provide intermittent connectivity. If there is only one mobile collector, we have the same drawbacks as in architecture 2. However, when several mobile collectors exist, this architecture is more efficient and scalable.
Architecture 4 is an optimization of the architectures 1, 2 and 3 to reduce the number of nodes visited. The trajectory of the mobile sink is smaller than in architecture 2. The energy consumed by sensor nodes is less than in architecture 1 and the data gathering delay is smaller than in architectures 2 and 3. The challenge of this architecture is in the optimization of the location of throwboxes.
Architecture 5 is used when the sink is static. This architecture takes advantage of several mobile collectors or ferries to reduce the data gathering delays compared to architecture 3. It can be seen as a generalization of architecture 4 when the sink is static.

In this paper, we focus on applications that require full area coverage of an industrial temporary worksite and network connectivity. In addition, time and space consistency of the data collected is needed. For this purpose, we are interested in uniform deployments where the sink is located at the center of the worksite monitored. The architecture proposed in Section 2.1 corresponds to architecture 1 in this state of the art. The mobile robot, static during the data gathering related to a given worksite, plays the role of the static sink in architecture 1. To ensure full area coverage of a temporary worksite, the number of sensors may be too high to have a mobile sink visiting each sensor node.

2.3 State of the art of redeployment algorithms

We now present a state of the art related to redeployment strategies ensuring full area coverage.

2.3.1 Redeployment algorithms: distributed versus centralized

Sensor deployment algorithms have been proposed in WSN to determine the efficient sensor positions in order to detect each event occurring in the monitored area and report it to a sink. They can be either centralized or distributed.

- Centralized algorithms

In a centralized algorithm, the computation of node positions is done by a central entity. This central entity collects information (e.g. energy, position) from all sensor nodes and computes their final positions accordingly. Sensor nodes physically move only once at the end of the algorithm when the central entity assigns them their final positions. Some are based on virtual forces such as (25) and (14), others on particle swarm optimization (24), or are computational geometry based (9).

- Distributed algorithms

In a distributed algorithm, all sensor nodes run the same algorithm. They are dynamic and autonomous. They compute their new positions according to local information gathered from their surroundings. Then, distributed algorithms involve all nodes that cooperate to compute their appropriate positions to ensure the full coverage in the considered area. Examples are given by the distributed self-spreading algorithm (26) inspired by the equilibrium of molecules, force based genetic algorithm (22) or mass-spring-relaxation algorithm (27).

2.3.2 Redeployment strategies

In order to solve coverage problems in WSNs, several sensor redeployment strategies have been proposed in the literature: Computational geometry based approach, Grid based approach, Force based approach and others.

- Geometry based approach

is generally used in WSNs with the goal of optimizing the coverage rate. Voronoi diagram and Delaunay triangulation are commonly used in computational geometry approach. The Voronoi diagram is a method of partitioning the area into a number of polygons based on distances to a specific discrete set of nodes as shown in Figure 3a. Each node occupies only one polygon and is closer to any point in this polygon rather than any other node in the neighboring polygons. These polygons can be obtained by drawing the mediator of each two neighbor nodes. Consequently, the edges of polygons are equidistant from neighboring nodes. Voronoi diagram is dual to Delaunay triangulation. The Delaunay triangulation can be obtained by connecting the nodes in the Voronoi diagram whose polygons share a common edge as shown in Figure 3b. Sensor nodes can construct the Voronoi diagram according to the location information exchanged in the network. Coverage holes can be detected using the Voronoi polygons. Authors in (9) propose three algorithms, the VECtor based algorithm (VEC), the VORonoi based algorithm (VOR) or Minimax algorithm, in order to reduce or eliminate coverage holes. VEC algorithm pushes sensor nodes away from higher coverage area while VOR algorithm pulls sensor nodes to the poor coverage area. Minimax algorithm is similar to VOR algorithm since it reduces coverage holes by moving sensor nodes toward the furthest Voronoi vertex. In centralized mode, geometry based redeployment algorithms require to know the position of all sensor nodes. This global knowledge is hard to obtain in large wireless sensor networks and in disconnected islands of communication. In distributed mode, the existence of disconnected islands is still a problem. In both modes, geometry based redeployment algorithms require a high computation complexity to provide a uniform deployment with similar Voronoi polygon.

- Grid based approach

is used to determine sensor positions. Each sensor node is placed exactly at the appropriate grid point. Grid point positions are calculated according to the communication range R and the sensing range r of sensor nodes. As a result, this strategy provides a high coverage rate and guarantees the network connectivity. We distinguish three types of grid: triangular lattice, square grid and hexagonal grid (see Figure 4). Notice that the triangular lattice also provides hexagonal grids (see Figure 4c), but there is a node at the center of each hexagonal cell, unlike the hexagonal grid (see Figure 4b). The triangular lattice offers the smallest overlapping area and requires the least number of sensor nodes if $R \geq \sqrt{3}r$ (10). For instance, the deployment algorithm HGSDA (11) deploys sensor nodes in an triangular lattice. It identifies redundant sensor nodes in order to place them in empty hexagonal cells. A distributed version, using the same pattern is proposed in (12). A node selects six of its neighbors with which it forms an hexagonal cell centered at itself. Grid based approach is also used for sensors deployment assisted by a mobile robot. As an example in (13), a robot places sensor nodes at the vertices of a square.

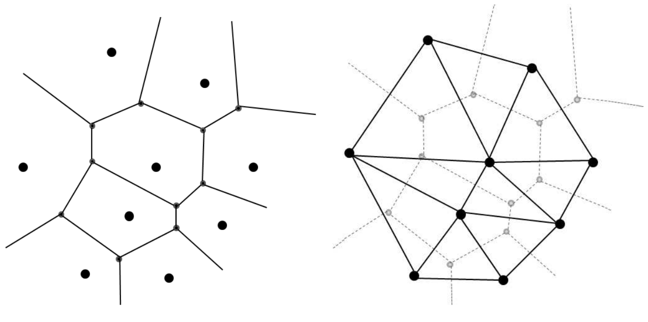

a Voronoi diagram b Delaunay triangulation

Figure 3: Computational geometry approach.

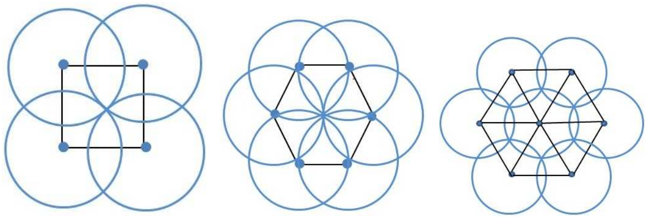

a Square b Hexagonal c Triangular

Figure 4: Grid based approach.

In this paper, we take advantage of the optimal deployment given by the triangular lattice to optimize our redeployment algorithm.

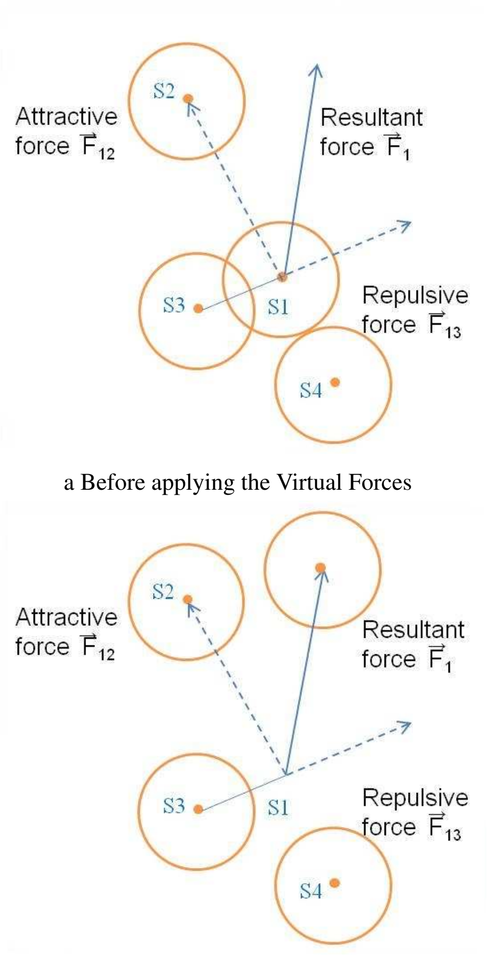
- Forces based approach

relies on the mobility of wireless sensor nodes and the existence of attractive or repulsive virtual forces between nodes. An attractive force is exerted if the distance between two neighboring sensor nodes is higher than the target distance. But, if the distance between two neighboring sensor nodes is lower than the target distance, a repulsive force is exerted to enhance the coverage in the surrounding. Figure 5 shows the position of one node before and after applying the virtual forces approach. This strategy tends to obtain a deployment where nodes are uniformly distributed in the whole area and all neighboring nodes are equidistant. In the literature, the Virtual Forces Algorithm (VFA) is proposed in a centralized (14), (15), (16) and a distributed (15), (17), (18) versions. It aims at ensuring a high coverage rate and maintaining the network connectivity. Several variants exist.

For instance in this, IVFA (19), Improved Virtual Force Algorithm, limits the scope of the virtual forces to twice the sensing range and defines a maximum movement in each iteration to reduce useless move and energy consumption. Another example of VFA enhancement is given by CPVF, Connectivity-Preserved Virtual Force (20). CPVF aims at maximizing the sensing coverage and guaranteeing the network connectivity. The sensor node that does not receive the message flooded by the sink is considered disconnected and should move towards the sink to reconnect. To improve the global coverage, the area is divided into virtual equidistant floors and sensors are encouraged to stay on the floors. An extended virtual force-based approach (21) is proposed to cope with different ratio of communication range to sensing range, R/r. This paper focuses on ensuring the ideal deployment defined by an equilateral triangular grid of edge value equal to $\sqrt{3}r$ like (14). The two distributed algorithms proposed (one for low R/r and one for high R/r) use damping coefficients to reduce node oscillations. The difficulty lies in the selection of the damping coefficient values. Another force based approach called FGA, Force based Genetic Algorithm is designed for unmanned vehicles (22). It consists in generating r chromosomes which correspond to r possible positions and speeds of sensor nodes. A node runs FGA for g generations and chooses the chromosome that indicates the most suitable direction to take. This solution needs a high computational power.

- Other approaches

PSO (23), Particle Swarm Optimization, is a computational method that tries to move particles in order to reach their best position in the considered area to achieve the global swarm best position. Another example is an hybrid algorithm (24) that combines the virtual forces and the particle swarm optimization where the Virtual forces is used to improve the convergence of PSO.


S2
Attractive force $\vec{F}_{12}$
Resultant force $\vec{F}_1$
S3
S1
Repulsive force $\vec{F}_{13}$
S4
S2
Attractive force $\vec{F}_{12}$
Resultant force $\vec{F}_1$
S3
S1
Repulsive force $\vec{F}_{13}$
S4

a Before applying the Virtual Forces

b After applying the Virtual Forces

Figure 5: Virtual forces approach.

2.4 Our contribution

In this paper, we propose an architecture used for data gathering in temporary worksites, based on a uniform redeployment of mobile sensor nodes. Such a redeployment should ensure full area coverage and maintain network connectivity to have an accurate data gathering.

As explained in the state of the art, a redeployment algorithm can be either centralized or distributed. Centralized redeployment algorithms assume that any node is able to communicate with the central entity. This assumption is violated in case of an initial deployment with several connected components. Moreover, the central entity may not be able to communicate the final positions to all nodes due to network partitioning. In addition, the centralized algorithm is not scalable due to the overhead needed to gather information required by the computation and dissemination of the final deployment. That is why, we focus on distributed algorithms. We choose the virtual force based approach because it relies on a simple principle and requires only localized information in its distributed version.

In this paper, we show how to compute the optimal number of sensor nodes needed to cover a given rectangular area. We propose also two redeployment algorithms, DVFA and its enhancement ADVFA that provide a uniform redeployment of mobile sensor nodes. These two distributed deployment algorithms require only a minimum a priori knowledge. The minimum knowledge needed is the dimensions of the area to cover. Notice that this assumption is usually made in all redeployment algorithms. Furthermore, DVFA and ADVFA do not make unrealistic assumptions such as the knowledge of the number of operational sensor nodes that can differ from the number of sensors deployed due to possible failures.

A disconnected sensor network, caused by the failure of some nodes, a random deployment with coverage holes or an initial deployment with several connected components, are examples of problems that must be solved to ensure an accurate data gathering. The algorithms we propose dynamically discover their environment and are self-adaptive. Basically, at each algorithm iteration, each sensor node discovers its neighbors, computes the virtual forces exerted on itself and moves according to the resulting force to a new location where it starts a new iteration of the algorithm. Then, a uniform deployment that ensures the full area coverage and network connectivity is provided. Furthermore, this uniform deployment is the cornerstone of our architecture for an efficient data measurement and gathering in temporary worksites.

2.5 Statement of the redeployment problem

To ensure full area coverage and network connectivity in this work, we adopt the following assumptions:

A1 Each sensor node can be seen as an autonomous, mobile robot. Hence, WSN can be seen as a group of mobile robots collaborating to fulfill the mission given by the application: i.e. to monitor the temporary worksite and deliver the data collected to the controller robot.

A2 Each sensor node/robot has:

- a sensing capacity characterized by a sensing range r,
- a communication capacity characterized by a radio range R. We assume that $R \geq \sqrt{3}r$.
- a processing capacity,
- an energy capacity,
- a memory capacity.

A3 Each sensor node/robot knows its own position (via GPS or other positioning technology).

A4 The temporary worksite, also called monitored area, is assumed to be a 2-dimension area.

A5 Each sensor knows the characteristics of the monitored area: for instance, the length L and the width W for a rectangular area.

3 Theoretical computations

In this section, we show how to compute the optimal number (i.e. the minimum number) of sensors needed to obtain the full coverage of a given area. As proved in (10), an optimal placement of sensors in a 2-dimension area offering full coverage can be obtained by a triangular lattice as illustrated in Figure 6. Let D_{th} be the target distance between two neighboring sensors. This distance is the target one to cover the considered area with the minimum number of sensors. If the targeted deployment is the optimal one, each sensor node has six neighbors at a distance D_{th}. The optimal deployment is obtained with an equilateral triangular lattice, (see Figure 6) where each sensor node has 6 neighbors at the same fixed distance D_{th}. Each sensor node occupies a vertex of an equilateral triangle. In Figure 6, a circle of radius r around a sensor node denotes its sensing area.

Figure 7 represents three sensors A, B and C in the optimal deployment. The coverage area of each sensor is presented by a disk of radius r. The centers of these three disks form an equilateral triangle ABC since these sensors are neighbors and are separated by the same distance D_{th}.

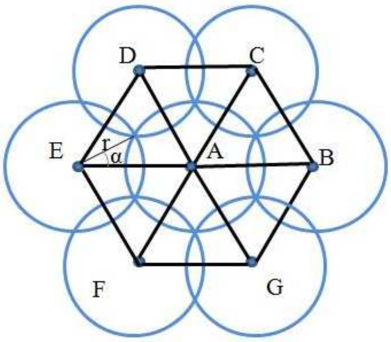


Figure 6: Triangular lattice deployment.

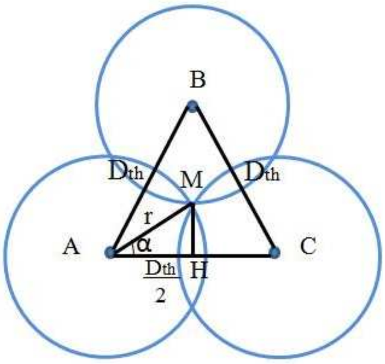


Figure 7: Basic pattern in an optimal deployment.

3.1 Computation of the target distance in the optimal deployment

Let M be the point of intersection of these three disks. AM is the radius r of the circle whose center is A. Since H is situated in the medium of AC then MH is the mediator of AC.

To compute the value of D_{th}, we consider the angle $H\widehat{A}M$, denoted by α (see Figure 7). As $\cos\alpha = \frac{AH}{AM} = \frac{\frac{D_{th}}{2}}{r}$, we can deduce:

$$D_{th} = 2rcos\alpha \tag{1}$$

In the optimal deployment, the angle $C\widehat{A}B$ is equal to $\frac{\pi}{3}$, because of the equilateral triangle. Since α is the half of the angle $C\widehat{A}B = \frac{\pi}{3}$, we have $\alpha = \frac{\pi}{6}$.

Consequently,

$$\boxed{D_{th} = \sqrt{3}r \text{ in the optimal deployment}} \tag{2}$$

To ensure network connectivity, the communication range R must be higher than the distance separating two neighbor sensors (i.e $R \geq D_{th}$) . Therefore, when the optimal deployment is reached, we have:

$$R \geq \sqrt{3}r$$

Coverage and connectivity are closely related. In fact, if the sensing range r and the transmission range R meet $R \geq \sqrt{3}r$, then it is sufficient to ensure coverage, connectivity is a consequence. In that case, we can relax the connectivity constraint, the only goal considered is to achieve coverage.

In the following, we assume $R \geq \sqrt{3}r$.

3.2 Optimal number of sensors to cover a given area

ADVFA adapts the distance separating neighboring sensor nodes to the total number of already discovered nodes. So before computing D_{th}, the appropriate target distance for any given number of deployed nodes, we need to compute the target distance for an optimal number of sensor nodes.

To determine the optimal number of sensors required to achieve the full coverage, we consider the optimal deployment illustrated in Figure 8 in an area of length L and width W. It is based on an equilateral triangular lattice of edge D_{th} (see triangle ABC in Figure 8). Since in the optimal deployment of sensors, the pattern of the first line is reproduced identically at each odd line and similarly the pattern of the second line is reproduced identically at each even line, we compute the number of sensors in odd lines and even lines (see Figure 8). We then compute the total number of lines and finally deduce the total number of deployed sensors.

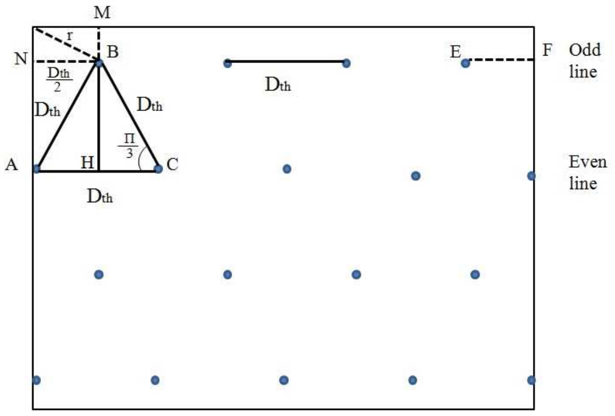


Figure 8: Optimal deployment of sensors.

- **Number of sensors in odd lines**

In the first line and in any odd line, the first sensor is located at a distance $\frac{D_{th}}{2}$ (represented by NB in Figure 8) from the left boundary of the considered area. On a line all sensors are uniformly distributed at a distance of D_{th}. Let $N_{s,o}$ be the number of sensors in odd lines. Let $\delta_{s,o}$ be an integer equal to 0 or 1 computed as follows:

$$N_{s,o} = \lfloor \frac{L - \frac{D_{th}}{2}}{D_{th}} \rfloor + 1 + \delta_{s,o} \tag{3}$$

$$with\ \delta_{s,o} = \begin{cases} 1\, if\ L - D_{th} - \lfloor \frac{L - \frac{D_{th}}{2}}{D_{th}} \rfloor D_{th} > 0 \\ 0\, otherwise \end{cases}$$

$\delta_{s,o}$ is equal to 1 when the distance between the last sensor in the line and the right boundary (represented by EF in Figure 8) is higher than $\frac{D_{th}}{2}$.

- **Number of sensors in even lines**

In even lines, the first sensor is located at the left boundary of the given area.
Let $N_{s,e}$ be the number of sensors in even lines. We have

$$N_{s,e} = \lfloor \frac{L}{D_{th}} \rfloor + 1 + \delta_{s,e} \tag{4}$$

$$with\ \delta_{s,e} = \begin{cases} 1\, if\ L - \frac{D_{th}}{2} - \lfloor \frac{L}{D_{th}} \rfloor D_{th} > 0 \\ 0\, otherwise \end{cases}$$

$\delta_{s,e}$ is equal to 1, if the distance between the last sensor and the right boundary is higher than $\frac{D_{th}}{2}$.

- **Number of sensors lines**

The first line starts at a distance BM from the top of the considered area (see Figure 8). The computation of BM is done in the triangle NBM of Figure 8.
$BM^2 + (\frac{D_{th}}{2})^2 = r^2$
As $D_{th} = 2rcos\alpha$, then
$BM^2 = r^2(1 - cos^2\alpha)$
And then, $BM = rsin\alpha$.
The distance between lines is represented by BH in Figure 8.
$BH = D_{th}sin\frac{\pi}{3} = \frac{\sqrt{3}}{2}D_{th}$. Finally, we get:
$BH = \sqrt{3}rcos\alpha$
Consequently the number of lines denoted by N_l is given by:

$$N_l = \lfloor \frac{W - rsin\alpha}{\sqrt{3}rcos\alpha} \rfloor + 1 + \delta_l \tag{5}$$

$$with\ \delta_l = \begin{cases} 1\, if\ W - 2rsin\alpha - \lfloor \frac{W - rsin\alpha}{\sqrt{3}rcos\alpha} \rfloor \sqrt{3}rcos\alpha > 0 \\ 0\, otherwise \end{cases}$$

- **Number of sensors**

The total number of sensors in a given area, is the sum of the total number of sensors in odd lines and the total number of sensors in even lines denoted by N_{opt} is:

$$N_{opt} = \lfloor \frac{N_l}{2} \rfloor N_{s,e} + \lceil \frac{N_l}{2} \rceil N_{s,o}$$

$$N_{opt} = \lfloor \frac{\lfloor \frac{W - rsin\alpha}{\sqrt{3}rcos\alpha} \rfloor + 1 + \delta_l}{2} \rfloor (\lfloor \frac{L}{D_{th}} \rfloor + 1 + \delta_{s,e})$$

$$+ \lceil \frac{\lfloor \frac{W - rsin\alpha}{\sqrt{3}rcos\alpha} \rfloor + 1 + \delta_l}{2} \rceil (\lfloor \frac{L - \frac{D_{th}}{2}}{D_{th}} \rfloor + 1 + \delta_{s,o}) \tag{6}$$

3.3 Computation of the effective distance

We now assume that N, the number of operational sensor nodes, is given with $N \geq N_{opt}$. Our goal is now to obtain a uniform redeployment in a given area $L * W$, using all the N sensors. This uniform redeployment is also based on a triangular lattice, where any node is at a distance D_{eff} from its adjacent neighbors.
According to Equation 6, we get:

$$N = \lfloor \frac{\lfloor \frac{W - rsin\alpha}{\sqrt{3}rcos\alpha} \rfloor + 1 + \delta_l}{2} \rfloor (\lfloor \frac{L}{D_{eff}} \rfloor + 1 + \delta_{s,e})$$

$$+ \lceil \frac{\lfloor \frac{W - rsin\alpha}{\sqrt{3}rcos\alpha} \rfloor + 1 + \delta_l}{2} \rceil (\lfloor \frac{L - \frac{D_{eff}}{2}}{D_{eff}} \rfloor + 1 + \delta_{s,o}) \qquad (7)$$

In this work, we use the mathematical software Maple, to solve Equation 7. Knowing the size of the considered area, we deduce the value of D_{eff} while varying N, the number of operational nodes.

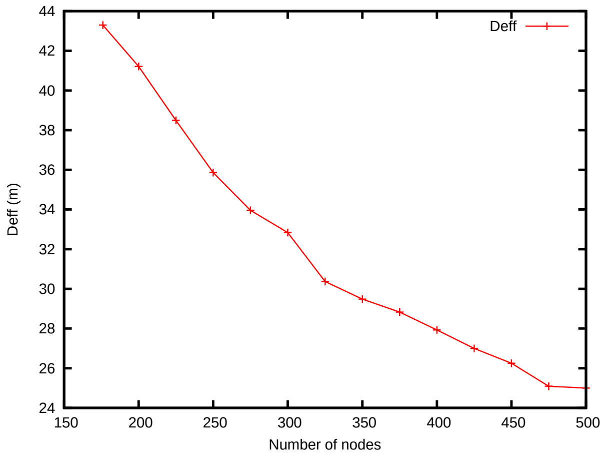


Figure 9: The effective optimal distance.

Figure 9 depicts the value of D_{eff} for a $500m$x$500m$ area and a sensing range $r = 25m$. The optimal value D_{th} is equal to $43.3m$ and it is obtained for 178 nodes.
As expected the distance D_{eff} decreases when the number of nodes increases. This corresponds to a higher density of nodes in the considered area. In the following, we consider only numbers of nodes higher than N_{opt}.

4 Parameters for the performance evaluation

4.1 Simulation parameters

We have implemented the DVFA algorithm as an agent in the NS2 simulator and have performed simulations for different wireless sensor networks. Simulation parameters are given in Table 1.
The values of Ka and Kr have been experimentally determined to increase the area coverage and the convergence of the centralized virtual forces algorithm (14). Notice that in each iteration the move of any sensor node is limited by $Lmax$ to reduce oscillations. We use the $Hello$ period value of $2s$, value recommended by the IETF for neighboring discovery (29) in wireless networks with nodes moving at moderate speeds. Notice that this value meets rule **R1** given in Section 5.1. The IEEE 802.11b MAC protocol has been used, because many mobile robots are equipped with such an interface. Furthermore, this assumption makes sense, knowing that the evolution of the IEEE 802.15.4 MAC protocol and its performances are getting closer to the 802.11b protocols. Using these parameters, we compute the value of D_{th} in the optimal triangular lattice using Equation 2. The obtained value is $D_{th} = 43.3m$. From Equation 6, we compute $N_{opt} = 178$. For these simulations, we use a number of sensor nodes equal to $250 > N_{opt}$.

Table 1 Simulation parameters.

Topology	
Sensor nodes	250 or 220 (failed nodes) for different initial topologies
Area size	500m x 500m
Speed	5m/s
Simulation	
Result	average of 30 simulation runs
Simulation time	5000s
MAC	
Protocol	IEEE 802.11b
Throughput	2 Mb/s
Radio range R	50 m
Sensing range r	25 m
DVFA	
Ka	0.001
Kr	0.56
Hello period	2s
Bitmap period	5s
$Lmax$	$D_{th}/6$

4.2 Simulated topologies

The data gathering process can partially fail if the network is disconnected, specially when sensor nodes cooperate to report the detected information to the controller robot. That is why we study the topologies depicted in Figures 10b and 10d. Each of them corresponds to a temporary worksite.

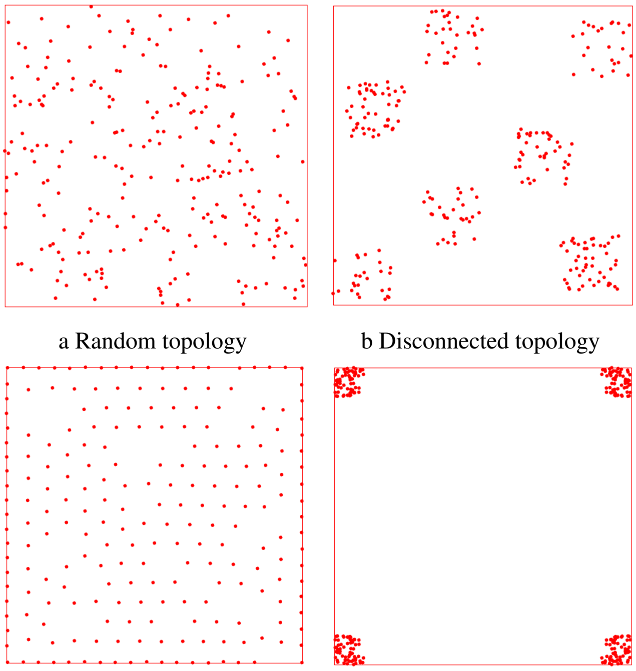
- Disconnected topology: In the disconnected topology, several disconnected islands of connected nodes exist in the temporary worksite (see Figure 10b). This topology corresponds to several groups operating in the same worksite, but in non contiguous zones.

- Four entry points topology: The initial topology depicted in Figure 10d, corresponds to a scenario where different teams organize themselves to monitor the worksite starting from different entry points (four entry points in our case).

Moreover, the presence of coverage holes in deployment causes a problem for data gathering process since data corresponding to coverage holes are missing. Hence, we will study the two configurations depicted in Figures 10a and 10c.

- Random topology: In the random topology, sensor nodes are randomly scattered in the worksite (see Figure 10a).
- Failed topology: The topology depicted in Figure 10c presents a uniform deployment where some sensor nodes have failed. These failures are due to, for instance, a battery depletion.

For each initial topology, 30 random configurations are simulated. The figures given in Section 6 depict the average value with the standard deviation.


a Random topology

b Disconnected topology

c Topology with failed nodes

d Topology with 4 entry points

Figure 10: Initial topologies.

4.3 Computation of the coverage

To compute the coverage rate, we virtually divide the network area into LxW grid units. A grid unit is considered covered if and only if its centered point is covered by at least one sensor node. The coverage rate is computed as the percentage of grid units covered. Notice that in the distributed redeployment algorithm, the computation of the coverage rate cannot be done locally by nodes. In the performance evaluation, we evaluate the coverage rate dynamically as a function of time. This evaluation is done using the node positions at the current time.

4.4 Computation of the traveled distance

During the deployment, the main source of energy consumption in sensor nodes is due to the sensor moves. In our simulations, we have not directly measured the energy consumed during the deployment. However, we evaluate the total distance traveled by sensor nodes. The energy is proportional to this distance. Hence, the values given by Figures 14, 17, 19, 21 and 23 reflect the energy consumed during the deployment.

5 DVFA: Distributed Virtual Forces Algorithm

5.1 DVFA principles

DVFA, Distributed Virtual Forces Algorithm, is a distributed sensors redeployment algorithm applying the virtual forces approach. The goal of DVFA is to ensure the full coverage of the considered area while maintaining the network connectivity. Autonomous sensor nodes move according to the virtual forces exerted on them by their surrounding nodes. The idea is to maintain a target distance D_{th} between two neighbors. Knowing the dimensions of the area to cover, the algorithm computes D_{th} as the result of Equation 2, assuming that the number of nodes is higher than or equal to N_{opt} given in Equation 6. Notice that, if the number of operational sensors is smaller than the minimal number required to achieve full coverage, DVFA maximizes the coverage that can be obtained with this number.

In DVFA, each node repeats the following steps: neighborhood discovery, forces computation and move to its new position, as shown in Figure 11. More precisely, it proceeds as follows:

- Each node s_i sends periodically a $Hello$ message that contains its position obtained from a GPS and its 1-hop neighbors with their positions. This message allows the node to discover its 1-hop and 2-hop neighbors.
- Each node s_i computes the forces exerted on it by its 1-hop and 2-hop neighbors. The force exerted by s_j on s_i where s_j is any 1-hop or 2-hop neighbor of s_i is:
  - Attractive if $d_{ij} > D_{th}$, where d_{ij} is the euclidean distance between s_i and s_j. We have $\overrightarrow{F_{ij}} = K_a(d_{ij} - D_{th})\frac{(x_j - x_i, y_j - y_i)}{d_{ij}}$, where K_a is a coefficient in $[0, 1)$, (x_i, y_i) and (x_j, y_j) are the coordinates of s_i and s_j respectively;
  - Repulsive if $d_{ij} < D_{th}$. We have $\overrightarrow{F_{ij}} = K_r(D_{th} - d_{ij})\frac{(x_j - x_i, y_j - y_i)}{d_{ij}}$, where K_r is a coefficient in $[0, 1)$;
  - Null if $d_{ij} = D_{th}$.

  The resulting force exerted on s_i is equal to $\overrightarrow{F_i} = \sum_j \overrightarrow{F_{ij}}$.
- Each node s_i moves to its new position (x'_i, y'_i) with $x'_i = (x_i +$ x-coordinate of $\overrightarrow{F_i})$ and $y'_i = (y_i +$y-coordinate of $\overrightarrow{F_i})$.

- Before moving, each node s_i sends a Bye message containing its new position. This message allows neighbors to update their 1-hop and 2-hop neighbor table. The Bye message decreases the convergence time of DVFA.

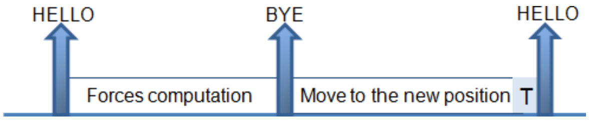


Figure 11: The three steps in an iteration of DVFA.

To maintain network connectivity and limit the total distance traveled by each node at each iteration, the distance to the new position can never exceed a fixed threshold $Lmax$. $Lmax$ reduces oscillations in sensor moves and then enables nodes to save energy.
Rule R1**:** The $Hello$ period must be larger than the time needed to compute DVFA and to travel the distance $Lmax$, as shown in Figure 11.
We notice that DVFA does not need the knowledge of the exact number of operational nodes. For this reason, DVFA uses the value of D_{th} computed for the minimum number of nodes needed to fully cover the given area.

5.2 Performance results for DVFA

Figure 12, illustrates the final deployment obtained with DVFA for initial topology 4, providing a quasi-uniform deployment with a 99.9% coverage rate. Figure 13 depicts the coverage rate as a function of the time for these four initial topologies. The first 500s are crucial to improve the coverage rate. After this time, the additional gain is small and almost null. For all topologies, DVFA achieves a very good coverage, it reaches 99.9% for the four toplogies described in Section 4.2.

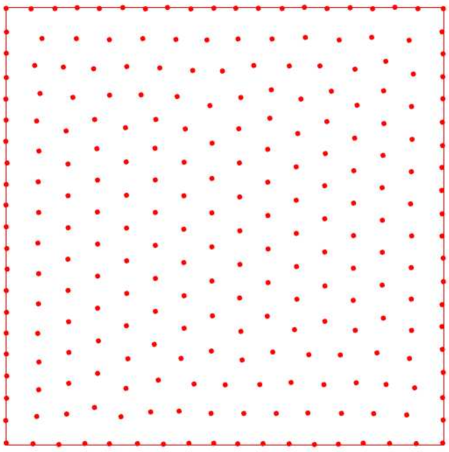


Figure 12: Final deployment of topology 4 with DVFA.

We now evaluate the total distance traveled by nodes in DVFA as shown in Figure 14. We observe a very big gap between the total distance traveled by nodes during the simulation and the distance traveled by nodes when the maximum coverage is reached for the first time. This gap can be explained by the node oscillations. In fact, even if the maximum coverage rate is reached, the nodes continue to run the DVFA algorithm and move accordingly. These oscillations lead to energy waste.

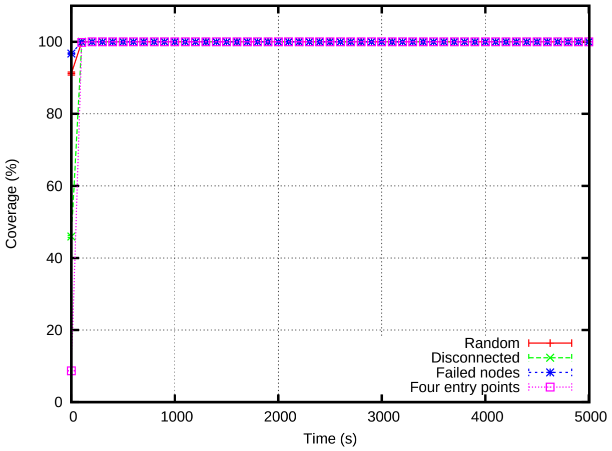


Figure 13: Coverage rate as the function of time.

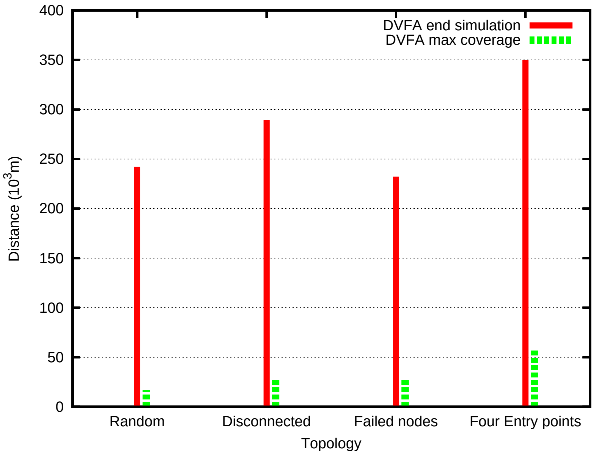


Figure 14: Distance traveled by nodes.

5.3 How to reduce node oscillations

DVFA is easy to implement. Since we have reduced the scope of virtual forces to 1-hop and 2-hop, the information needed by DVFA is limited to 1-hop and 2-hop neighborhood. Hence, DVFA can be easily coupled with any neighborhood discovery protocol. The main advantages of DVFA is to be fully distributed. Indeed, it is based on only local information to ensure full area coverage. Simulation results show that a coverage rate of 99.99% can be reached in many configurations. However, DVFA suffers from a major problem. Nodes move continuously, oscillating between different nearby positions, even when the maximum coverage is reached. It comes from the fact that a node does not know when the maximum coverage of the area is reached. Indeed, it is difficult to distinguish between a local optimum and a global one. This problem is still an open issue. Simulation results show that the use of an inappropriate D_{th} independent of the number of nodes amplifies the oscillation phenomenon. With this D_{th}, it becomes impossible to obtain an equilibrium where virtual forces are null. This behavior leads to high energy waste. We try to avoid this problem by proposing ADVFA: an Adaptive Distributed Virtual Forces Algorithm, which adapts the distance between neighbors to the total number of nodes.

6 ADVFA: Adaptive Distributed Virtual Forces Algorithm

6.1 ADVFA principles

ADVFA is also a fully distributed redeployment algorithm ensuring the full coverage of the considered area. Unlike DVFA, the target distance between two neighbors is not fixed but varies as a function of the number of nodes discovered. ADVFA is highly adaptive to any environment. Indeed, it adjusts its target distance according to the new discovered connected components. The goal is to obtain an homogeneous deployment to avoid oscillations using more appropriate distance between two neighbors depending on the number of nodes.

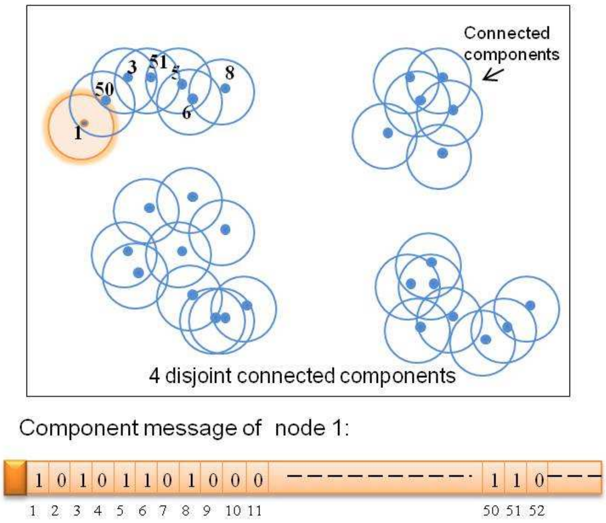


Figure 15: Bitmap of node 1 in its $Component$ message.

Like DVFA, ADVFA uses $Hello$ messages to build 1-hop and 2-hop neighborhood. Additional messages, called $Component$, are exchanged periodically between 1-hop neighbors to compute the number of connected operational nodes discovered in the area. The $Component$ message sent by a node s_i determines the operational nodes already discovered in its connected component. These operational nodes are represented in the $Component$ message by a bitmap: the j^{th} bit represents the node s_j. If it is equal to 1, node s_j is present in the connected component of s_i. See Figure 15 for an example of bitmap. Initially, each node s_i marks the i^{th} bit to one in its $Component$ message and sends it. Upon reception of the $Component$ messages, node s_i makes an OR operation between its own message and all $Component$ messages received and sends it in the next period. Consequently, node s_i is able to determine N, the number of operational nodes in its connected component by counting the number of marked bits:

- If $N \leq N_{opt}$ then $D_{eff} = D_{th}$ where N_{opt} is the optimal number of nodes needed to fully cover the given area and computed according to equations 3 to 6, and D_{eff} is the expected distance between two neighbors.
- If $N > N_{opt}$ then D_{eff} is solution of Equation 7.

ADVFA allows the discovery of connected components and naturally handles the merge of them. In fact, the first contact between two disjoint components will allow the exchange of $Component$ messages with their different bitmaps included. Thus, the corresponding D_{eff} is immediately deduced and broadcast in the new connected component resulting from the merge. Some nodes may fail due for instance to energy depletion. To take into account node failures occurring during the deployment algorithm, the bitmap is periodically recomputed from scratch to remove failed nodes. A re-computation of the bitmap of a connected component is triggered by an elected node (e.g. the node with the smallest address in this component).

6.2 Comparative evaluation of ADVFA and DVFA

In this series of simulations, the period of $Component$ messages is fixed to 5s. A short period of $Component$ messages is needed to track the number of the connected nodes already discovered. ADVFA adapts its parameters to this number in order to maintain the appropriate distance between neighboring nodes. Hence, it avoids useless moves. As long as new operational nodes are discovered, the target distance is updated. Depending on the adaptivity requirement, we may reduce the frequency of $Component$ messages, to save bandwidth.

6.2.1 Random topology

Figure 16 shows that ADVFA and DVFA provide an excellent coverage rate of 99.9%. This is due to the principle of virtual forces that contribute to maintain the target distance between neighbor nodes, as computed in Section 3. As a consequence, sensor nodes occupy the whole area leading to this result. This result is achieved at the cost of a total distance traveled depicted in Figure 17. We observe that ADVFA considerably reduces this distance by 64%.

6.2.2 Disconnected topology

Figure 18 shows that after a short time, the full coverage is achieved by both DVFA and ADVFA. However ADVFA has the merit of reaching this coverage with a smaller total distance traveled. As depicted in Figure 19, this distance is reduced by 61% compared to DVFA. As a conclusion ADVFA keeps the full coverage provided by DVFA and maximizes the network lifetime by reducing the energy consumption.

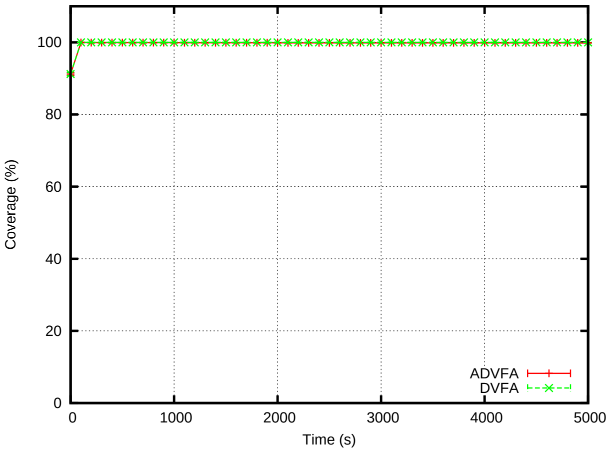


Figure 16: Coverage rate as the function of time (Random topology).

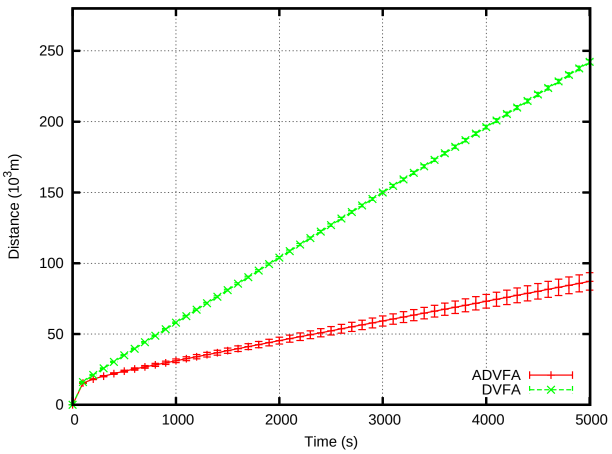


Figure 17: Distance traveled by nodes (Random topology).

6.2.3 Failed nodes

In the monitoring area, sensor nodes may fail due to their battery depletion. These failures are detected by both algorithms that use *Hello* messages to discover node neighborhood. However only ADVFA adapts the target distance to the new number of operational nodes. This is made possible by the exchange of the *Component* message that is periodically updated. We observe that ADVFA and DVFA achieve the full coverage rate as depicted in Figure 20. However, the distance traveled is considerably smaller with ADVFA (see figure 21). This is due to a target distance computed with the effective number of operational nodes, leading to a more stable redeployment.

ADVFA is robust with regard to node failures: it is able to adapt to the number of operational nodes that it progressively discovers. This quality of ADVFA can be very important for applications where sensors can be damaged during their initial drop or can fail because of energy depletion.

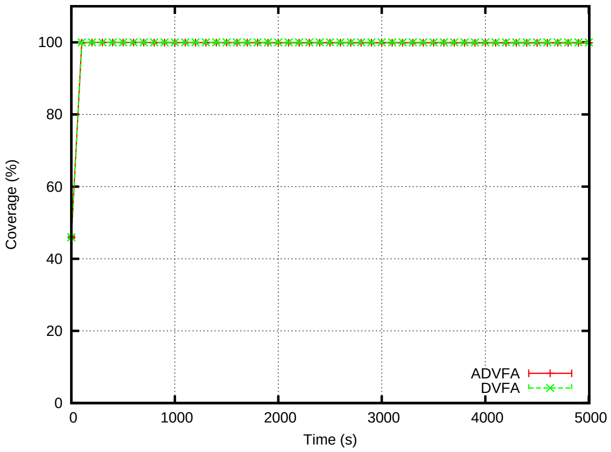


Figure 18: Coverage rate as the function of time (Disconnected topology).

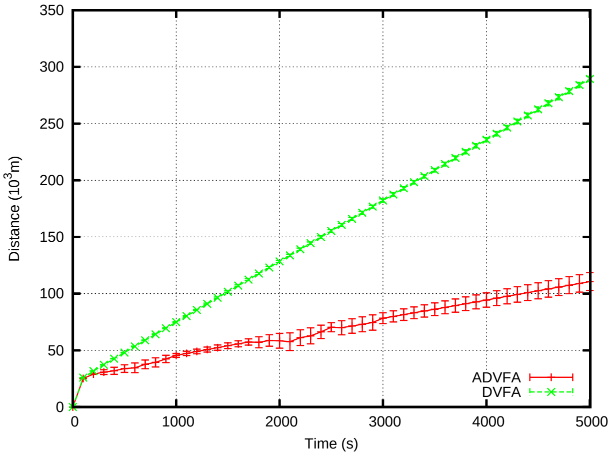


Figure 19: Distance traveled by nodes (Disconnected topology).

6.2.4 Topology with four entry points

Figures 22 depicts a very good coverage rates for both ADVFA and DVFA. However ADVFA considerably reduces the total distance traveled by nodes. We can observe in Figure 23, that the distance traveled by DVFA increases rapidly to reach 300Km at the end of the deployment whereas the distance traveled by ADVFA does not exceed 140Km. Hence, ADVFA is more energy efficient than DVFA.

6.3 Sensitivity to positioning accuracy

Positioning system like GPS may fail to provide the accurate position. Then, an error in positioning will be introduced and can cause some coverage holes in the network. In such a case, event occurring in this coverage holes can neither be detected nor collected and reported to the sink. In this section we show that ADVFA still provides good performances, even after relaxing some assumptions, like the accuracy of node positioning.

Until now we have assumed that each node has a perfect positioning system (Assumption A3). We now assume that the

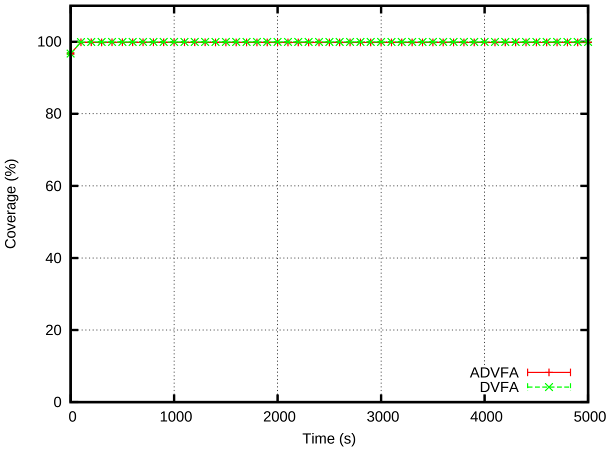


Figure 20: Coverage rate as the function of time (Failed nodes topology).

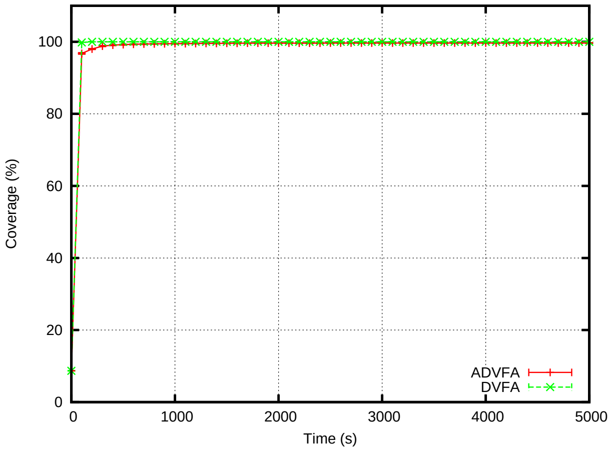


Figure 22: Coverage rate as the function of time (Four entry points topology).

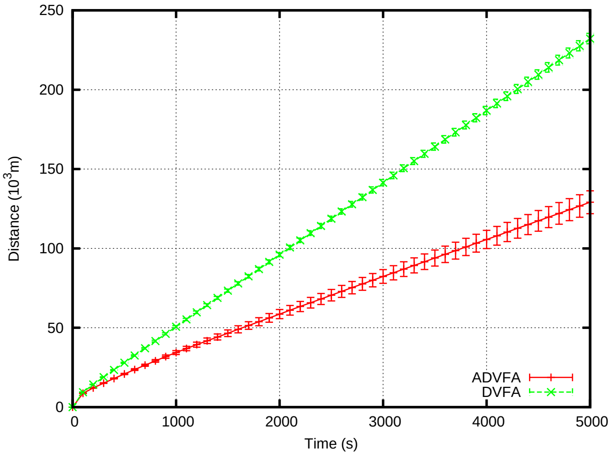


Figure 21: Distance traveled by nodes (Failed nodes topology).

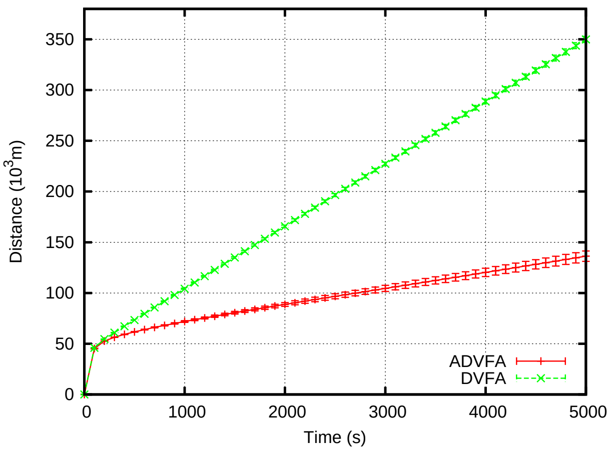


Figure 23: Distance traveled by nodes (Four entry points topology).

positioning system introduces a random error which is very common in positioning systems like GPS. The exact position of a node may differ from this computed by ADVFA by an angle $w \in [0, 2\pi]$, and a distance $d \in [-0m, +4m]$.

We now evaluate the sensitivity of ADVFA to the positioning accuracy. Even in the presence of positioning errors, ADVFA provides a very good coverage rate as shown in Figure 24 for the initial topologies. Figure 25 shows the final deployment provided by ADVFA when errors in positioning occur. We can observe that sensor nodes cover the whole area. So, ADVFA performs well even when there are some errors in positioning. This can be explained by the fact that ADVFA gets the node position from GPS before computing the resultant force in the current iteration. This principle prevents an accumulation of positioning errors.

7 Comparison with other deployment strategies

In this paper, we adopt the virtual forces based strategy to deploy sensor nodes in the area considered. Since this strategy uses repulsive and attractive forces, the values of repulsive and attractive coefficients should be determined. We favor the repulsive force to encourage the spreading of all the nodes in the area considered and we fix a small value of the attractive coefficient. However, a great value of the repulsive coefficient may cause network dis-connectivity since the attractive coefficient is small. To cope with the network connectivity problem and reduce node oscillations, we limit the distance traveled by a node in each iteration. In addition, since the virtual forces strategy aims to maintain the same target distance between neighboring sensor nodes, we compute this target distance based on the optimal deployment to minimize the overlapping zone between neighbors and then minimize the number of nodes needed to achieve full coverage of the area considered. In (36), the authors attribute a high coefficient to the attractive forces. For this reason, their deployment algorithm suffers from the stacking problem.

In the literature, many studies propose the centralized version of the virtual forces algorithm. However, this centralized version needs the global knowledge of the area to monitor. It requires also the connectivity between all the sensor nodes and the sink to send their initial positions and to receive their final positions. When the number of the sensor nodes is great,

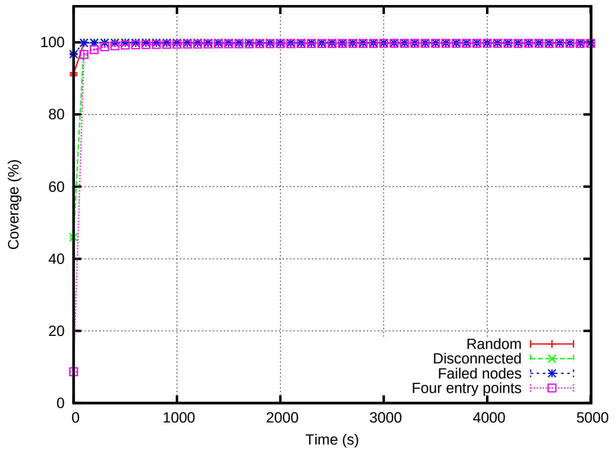


Figure 24: Coverage rate as a function of time with error in positioning with ADVFA.

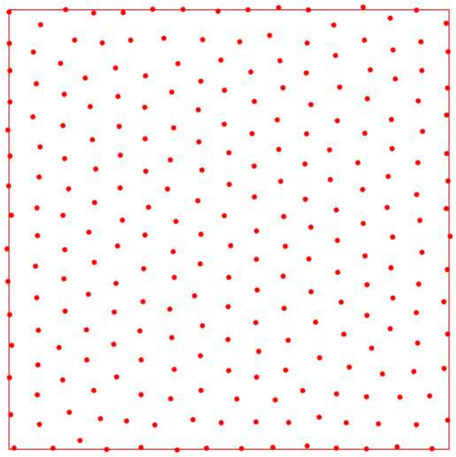


Figure 25: Final deployment with error in positioning with ADVFA.

this exchange of messages is expensive in bandwidth and energy. Then, the centralized virtual forces algorithm cannot be scalable.

In the distributed version of the virtual forces, sensor nodes are autonomous and able to spread in the whole area. For this reason, the distributed version is considered as the appropriate solution to monitor an area partially or totally unknown. ADVFA is an enhanced solution of the distributed virtual forces algorithm. In a previous work (37), we proved that the presence of obstacles in the area monitored did not inhibit sensor nodes to turn around the obstacle and cover the whole area. When the environment is partially or totally unknown, the number of deployed nodes may be higher than the optimal number needed to cover the area considered. Then, the deployment cannot be uniform since the sensor nodes will try to maintain a target distance that has been computed using the optimal number of nodes. To cope with this problem, ADVFA adapts the target distance between neighboring nodes to the number of connected nodes when this number is greater than the optimal one. Then, the uniformity of the deployment is ensured using ADVFA. Furthermore, since ADVFA detects the changes of the number of connected nodes and adapts the deployment to this number, it can easily cope with sensor failures.

Concerning the grid based strategy, it provides full area coverage and network connectivity. However, it requires a perfect knowledge of the environment which is not always possible. In addition the computation of the grid is centralized that makes this strategy non scalable. When, the environment has an irregular shape and contains obstacles, the computation of the grid may be not obvious (e.g. when the center of the cell is within an obstacle or outside the area to monitor).

The grid based strategy can be the most efficient solution in terms of the number of sensor nodes deployed and energy consumed, when the area considered is known, does not have an irregular shape and does not contain obstacles. These conditions are not usually met in the real life. In a previous work (38), we proposed an hybrid solution based on the grid based strategy to predetermine the node positions on the one hand and the virtual forces strategy to spread the sensor nodes in the whole area on the other hand. After a certain time (called spreading factor), each sensor node moves to the nearest cell center and stops moving. Then, redundant nodes are easily detected as they do not occupy cell centers. This algorithm provides a uniform and regular deployment that ensures full area coverage, maintains network connectivity and saves energy by switching off redundant nodes or using them to replace depleted nodes. However, a rectangular area without obstacles may not be a realistic environment.

The computational geometry based strategy is adopted in many deployment algorithms. It is based on adjusting the distance between neighboring sensor nodes like the virtual forces strategy. However, its principle is different and more complex than the virtual forces. The computational geometry based strategy aims to maximize the area coverage by finding the coverage holes and trying to relocate the nodes in order to cover these holes. This strategy is greedy in computation unlike the virtual forces strategy. It runs in centralized mode and has been extended in distributed mode. The existence of disconnected islands of communication is a problem in both modes. In addition, this strategy requires a high computational capacity to get the uniform deployment where all nodes occupy the center of similar Voronoi polygons for instance.

8 Global system functioning

8.1 Overview

The uniform deployment of sensor nodes provided by ADVFA is the cornerstone of our architecture for data management. This three-tier architecture comprises:

- *mobile sensor nodes* that should be uniformly deployed to sense the temporary worksite and produce data.

- *a controller robot and a local sink* in charge of local data management.
- *a global database* where data coming from different worksites are processed. Different applications can exploit these data using mining tools (e.g. a principle component analysis) to draw trends.

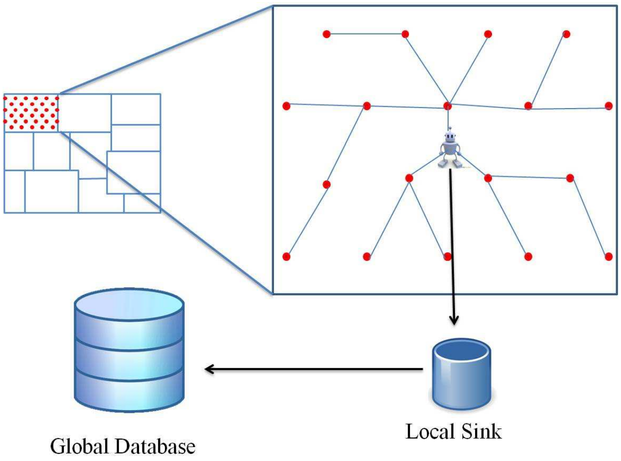


Figure 26: Data management steps.

Data management is achieved in 3 steps:

Step 1: uniform deployment of mobile sensor nodes using ADVFA. As described in Section 6, nodes will collaborate together and deploy uniformly in the considered area.

Step 2: after the redeployment provided by ADVFA, nodes stop moving. The phase of data measuring and gathering will start. This phase may last a long time. As a consequence, the energy spent in communication matters. The use of techniques to save energy is highly required. The data produced by the sensor nodes are delivered to the controller robot. The controller robot should be located at the barycenter of the worksite and initiates the building of the routing tree rooted at itself, as depicted in Figure 26. To build this routing tree, EOLSR, an energy efficient routing protocol (34) can be used. The uniform deployment of sensor nodes ensures a better balancing of the routing tree.

Such a communication paradigm is called *convergecast*. Depending on the capability of sensor nodes, we distinguish two types of convergecast:

- *convergecast with aggregation*, where any node aggregates the data received from its children with its own before transmitting them to its parent.
- *raw data convergecast*, where any node forwards the data received from its children without any additional processing.

In both cases, the nodes close to the controller robot will have a traffic load higher than nodes far away. This medium access contention can lead to collisions followed by retransmissions that increase data gathering delays and waste energy of sensor nodes. For a convergecast with aggregation, we have proposed SERENA (33), a node coloring algorithm such that each node transmits in the time slot associated with its color. For a raw data convergecast, we have proposed joint time slot and channel assignment algorithms, centralized or distributed. Such algorithms (32) and (35) take advantage of multichannel communications to reduce interferences and to increase the parallelism of transmissions. Their goal is to minimize the total number of slots required by the convergecast. As a consequence, the data gathering delays are reduced, leading to a better freshness and time consistency of the data measured. Furthermore, a node that is neither sender nor receiver in a slot turns off its radio to save energy. In any slot, only the senders and receivers are awake. As a consequence, there is no energy wasted in overhearing, interference and idle state. Energy efficiency is then maximized. Notice that we have proved in (32) that the number of slots required by the convergecast is minimized with a balanced routing tree. Hence, the importance of a uniform deployment of sensor nodes.

Any node with low residual energy notifies the controller robot. The controller robot will fire a new redeployment to prevent coverage holes or network partitioning.
The controller robot and the local sink collaborate to organize the data gathered according to predefined rules before sending them to the global database.

Step 3: global data management will centralize all the data coming from different worksites and different gathering times to draw evolution and trends.

8.2 Discussion about design efficiency

The strengths of our solution are:

- Energy and time efficiency

– During sensor deployment, ADVFA reduces nodes oscillations by adapting the target distance to the number of connected nodes. Then, it decreases node oscillations and save the energy consumption. The number of nodes deployed may change during the monitoring task due to the energy depletion of some sensor nodes. ADVFA copes with this problem since it is able to detect the node failures and relocate the sensor nodes to ensure full coverage, network connectivity while keeping the uniformity of the deployment.

– During data gathering, ADVFA provides a uniform deployment that makes the construction of a balanced tree of data collection simple and easy. Due to this balanced tree, data can be collected in a small delay. As explained in Section 7 step 2, our solution saves energy during data gathering, avoiding data retransmissions due to collisions, idle listening and overhearing.

- Dynamic discovery of the area to monitor

ADVFA is designed such that sensor nodes are autonomous and able to dynamically discover the area monitored that may contain obstacles. Due to the principle of the distributed version of the virtual forces, sensor nodes exchange Hello messages to discover their neighborhood and also to maintain network connectivity. Network connectivity is required to send data to the sink according to the architecture we have selected (i.e.: Architecture 1 in Section 2.2.1). Messages broadcast by the sink are used to build and maintain the routing tree during data gathering. Hence, our solution is helpful to ensure the monitoring task in an unknown or hostile environment.

- Robustness against sensor nodes failures

Sensor nodes may fail due to energy depletion or environmental conditions. Consequently, not only coverage holes will occur but also the network may become disconnected. ADVFA is able to cope with this problem due to the Component messages exchanged between sensor nodes to check the number of connected nodes. The sensor nodes can detect the node failures and notify the controller robot. The controller robot takes the decision to fire a new deployment based on a target distance adapted to the new number of nodes to avoid coverage holes and network partitioning.

9 Conclusion and future work

Generally, temporary industrial worksites are monitored by mobile wireless sensors. These sensor nodes transmit data measured on the worksite to a controller robot that processes them and sends then to a local sink. To ensure an accurate data gathering, a uniform deployment is needed. It produces a balanced routing tree that minimizes the data gathering delays. That is why we focus on uniform redeployment ensuring full coverage while maintaining network connectivity. In this paper, we proposed a three-tier architecture for data gathering and two redeployment algorithms ADVFA and DVFA, based on virtual forces, providing a uniform deployment. We also computed the optimal number of sensor nodes required to fully cover a given area. This computation is used by both ADVFA and DVFA. The main drawback of a distributed redeployment algorithm based on virtual forces is node oscillations: nodes are still moving even when the maximum coverage is reached. This is due to an inadequate value of the target distance D_{th} between neighboring nodes. This inaccuracy is corrected by ADVFA that adapts this target distance to the total number of operational nodes in the considered connected component. We have applied ADVFA in different initial topologies corresponding to various use cases: random topology (e.g.: dropping of sensor nodes in the whole area), disconnected topology (e.g.: several disconnected subsets of sensor nodes corresponding for instance to several teams operating in different zones) and a topology with coverage holes due to battery depletion of some nodes. Simulation results show that both DVFA and ADVFA achieve a very good coverage rate that is 99.9%, whereas the total distance traveled by sensor nodes is drastically reduced (at least 44%) with ADVFA. As a conclusion, the enhancement provided by ADVFA enables to save energy by reducing node oscillations and then to maximize network lifetime.

We can identify several tracks for further work. In a real environment, the two-dimension considered area is never flat, it has many obstacles. In (28), we show how to extend DVFA to cope with obstacles. This approach could also be applied to ADVFA.

The area considered in the performance evaluation is a rectangle. Notice that this is not required by the DVFA and ADVFA algorithms that work, even in case of other regular shapes. In case of non regular shapes like those considered in (31) and (30), the challenge is to approximate the optimal number of sensor nodes and that depends on the border shape. Temporary industrial worksites can be indoor worksites (e.g. renovation of an industrial plant). As said in the problem statement, both DVFA and ADVFA rely on the knowledge of nodes positions. Since GPS does not work in an indoor environment, another mechanism should be used. Notice, that it is sufficient to have a relative positioning. Each sensor node should express its position relatively to a common reference point, whose absolute coordinates can be unknown. For that purpose, a laser coupled with a camera could be used. Other solutions exist, based on ultrasounds for instance.

Acknowledgment

This work has been partially funded by the Cluster Connexion project. For more details see http://www.cluster-connexion.fr/.

References

[1] Y. Miao (2005) 'Applications of sensor networks', Seminar Wireless Self-Organization Networks.

[2] M. Di Francesco, S. K. Das and G. Anastasi (2011) 'Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey', ACM Transactions on Sensor Networks (TOSN).

[3] M. Cardei, Y. Yang, and J. Wu (2008) 'Non-Uniform Sensor Deployment in Mobile Wireless Sensor Networks', World of Wireless, Mobile and Multimedia Networks (WoWMoM).

[4] V. Kavitha and E. Altman (2010) 'Analysis and Design of Message Ferry Routes in Sensor Networks using Polling Models' Modeling and Optimization in Mobile, *Ad Hoc and Wireless Networks (WiOpt)*, Avignon.

[5] E. Sabir, A. Kobbane, M. A. Koulali and M. Erradi (2013) 'Design of an Annular Ring Ferry-Assisted Topology for Wireless Sensor Networks', WMNC.

[6] E. M. Saad, M. H. Awadalla and R. R. Darwish (2008) 'A Data Gathering Algorithm for a Mobile Sink in Large-Scale Sensor Networks', *Wireless and Mobile Communications, ICWMC*, Athens.

[7] H. Zhang and J. C. Hou (2005) 'Maintaining Sensing Coverage and Connectivity in Large Sensor Networks', *Ad Hoc and Sensor Wireless Networks*, vol. 1, no. 1-2.

[8] P. Sahu and S. R. Gupta (2012) 'Deployment Techniques in Wireless Sensor Networks', *International Journal of Soft Computing and Engineering*, vol. 2(3), pp. 525-526.

[9] G. Wang, G. Cao, T. F. La Portab (2006) 'Movement Assisted Sensor Deployment', *IEEE Transactions on Mobile Computing*, Vol 5, no 6.

[10] N. A. Ab. Aziz, K. Ab. Aziz, and W. Z. W. Ismail (2009) 'Coverage Strategies for Wireless Sensor Networks', *World Academy of Science, Engineering and Technology*.

[11] S. Han, Y. Zhang, G. Xu (2010) 'Hexagonal grid-based sensor deployment algorithm', *Control and Decision Conference*, Chinese.

[12] N. Bartolini, T. Calamoneri, E. G. Fusco, A. Massini and S. Silvestri (2010) 'Autonomous deployment of mobile sensors for a complete coverage' *Journal Wireless Networks*, vol. 16, I. 3, p. 607-625.

[13] L. Xu, A. Nayak, I. Stojmenovic (2010) 'Back Tracking based Sensor Deployment by a Robot Tea', *Sensor Mesh and Ad Hoc Communications and Networks*, USA.

[14] F. Kribi, P. Minet and A. Laouiti (2009) 'Redeploying mobile wireless sensor networks with virtual forces', *Proceedings of the 2nd IFIP conference on Wireless days*, USA.

[15] K. Mougou, S. Mahfoudh, P. Minet and A. Laouiti (2012) 'Redeployment of Randomly Deployed Wireless Mobile Sensor Nodes', *VTC*.

[16] Y. Zou, K. Chakrabarty (2003) 'Sensor Deployment and Target Localization Based on Virtual Forces', *IEEE INFOCOM*.

[17] Y. Zou, K. Chakrabarty (2007) 'Redundancy Analysis and a Distributed Self-Organization Protocol for Fault-Tolerant Wireless Sensor Networks', *IJDSN*.

[18] Y. Zou, K. Chakrabarty (2004) 'Sensor deployment and target localization in distributed sensor networks', *ACM Trans. Embedded Comput. Syst*.

[19] J. Chen, S. Li, and Y. Sun (2007) 'Novel deployment schemes for mobile sensor networks', *Sensors*, vol. 7, no. 11, pp. 2907-2919.

[20] G. Tan, S. A. Jarvis, A. M. Kermarrec (2009) 'Connectivity-Guaranteed and Obstacle-Adaptive Deployment Schemes for Mobile Sensor Networks', *IEEE Transactions on Mobile Computing*.

[21] J. Li, B. Zhang, L. Cui and S. Chai (2012) 'An Extended Virtual Force-Based Approach to Distributed Self-Deployment in Mobile Sensor Networks', *International Journal of Distributed Sensor Networks*.

[22] C. Safak Sahin, E. Urrea, M. Umit Uyar, M. Conner, G. Bertoli, C. Pizzo (2010) 'Design of genetic algorithms for topology control of unmanned vehicles', *International Journal of Applied Decision Sciences*, Vol. 3, No.3 pp. 221-238.

[23] G. Ciuprina, D. Ioan, and I. Munteanu, (2002) 'Use of intelligent-particle swarm optimization in electromagnetics', *IEEE transaction on magnetics*, vol. 38, no. 2.

[24] X. Wang, S. Wang and J. Ma (2007) 'An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment', *Sensors*.

[25] Y. Zou and K. Chakrabarty (2003) 'Sensor Deployment and Target Localization Based on Virtual Forces', *INFOCOM*.

[26] N. Heo and P. K. Varshney (2003) 'A distributed self spreading algorithm for mobile wireless sensor networks', *Wireless Communications and Networking (WCNC)*, 1597-1602 vol.3, New Orleans, LA, USA.

[27] F. Villanueva, R. German and F. Dressler (2011) 'Distributed Mass-Spring-Relaxation for Anchor-Free Self-Localization in Sensor and Actor Networks', *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, Maui, HI, USA.

[28] S. Mahfoudh, I. Khoufi P. Minet and A. Laouiti (2013) 'Relocation of Mobile Wireless Sensors in the Presence of Obstacles' *ICT*, CasaBlanca, Marroco.

[29] T. Clausen and P. Jacquet (2003) 'Optimized Link State Routing Protocol (OLSR)', *RFC 3626, Optimized Link State Routin*.

[30] Y. Wang, S. Lederer and J. Gao (2009) 'Connectivity-based Sensor Network Localization with Incremental Delaunay Refinement Method', IEEE *INFOCOM*.

[31] H. Tan, Y. Wang, X. Hao, Q. S. Hua and F. C.M. Lau (2010) 'Arbitrary obstacles constrained full coverage in wireless sensor networks', *In Wireless Algorithms Systems and Applications*, Springer Berlin Heidelberg.

[32] R. Soua, P. Minet, E. Livolant (2012) 'MODESA: an Optimized Multichannel Slot Assignment for Raw Data Convergecast in Wireless Sensor Networks', *IPCCC 2012, the 31st IEEE International Performance Computing and Communications Conference*, Austin, Texas.

[33] I. Amdouni, P. Minet, C. Adjih (2012) 'OSERENA: a coloring algorithm optimized for dense wireless networks', *the International Journal of Networked and Distributed Computing (IJNDC)*, Volume 1, Issue 1.

[34] S. Mahfoudh, P. Minet, I. Amdouni (2010) 'Energy efficient routing and node activity scheduling in the OCARI wireless sensor network', *Future Internet*, 2(3), 308-340.

[35] R. Soua, E. Livolant, P. Minet (2013) 'Adaptive Strategy for an Optimized Collision-Free Slot Assignment in Multichannel Wireless Sensor Networks', *Journal of Sensor and Actuator Networks*, Special Issue on Advances in Sensor Network Operating Systems.

[36] J. Li, B. Zhang, L. Cui, and S. Chai (2012) 'An Extended Virtual Force-Based Approach to Distributed Self-Deployment in Mobile Sensor Networks', International Journal of Distributed Sensor Networks.

[37] S. Mahfoudh, I. Khoufi, P. Minet and A. Laouiti (2013) 'Relocation of Mobile Wireless Sensors in the Presence of Obstacles' ICT 2013 - 20th International Conference on Telecommunications, Casablanca : Morocco.

[38] S. Mahfoudh, I. Khoufi, P. Minet and A. Laouiti (2014) 'GDVFA: A Distributed Algorithm Based on Grid and Virtual Forces for the Redeployment of WSNs', WCNC.