

## **GDVFA: A distributed algorithm based on grid and virtual forces for the redeployment of WSNs**

Saoucene Mahfoudh, Ines Khoufi, Pascale Minet, Anis Laouiti

► **To cite this version:**

Saoucene Mahfoudh, Ines Khoufi, Pascale Minet, Anis Laouiti. GDVFA: A distributed algorithm based on grid and virtual forces for the redeployment of WSNs. WCNC: IEEE Wireless Communication and Networking Conference, Apr 2014, Istanbul, Turkey. pp.3040 - 3045, 10.1109/WCNC.2014.6952969 . hal-01088696

**HAL Id: hal-01088696**

**<https://hal.inria.fr/hal-01088696>**

Submitted on 28 Nov 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# GDVFA: A Distributed Algorithm Based on Grid and Virtual Forces for the Redeployment of WSNs

Saucene Mahfoudh\*, Ines Khoufi\*, Pascale Minet\* and Anis Laouiti†

\*INRIA, Rocquencourt, 78153 Le Chesnay Cedex, France,

Email: saucene.mahfoudh@inria.fr, ines.khoufi@inria.fr, pascale.minet@inria.fr

†TELECOM SudParis, CNRS Samovar, UMR 5157, France,

Email: anis.laouiti@it-sudparis.eu

**Abstract**—The distributed virtual forces deployment algorithm provides a very good area coverage and guarantees network connectivity for a sufficient number of sensor nodes. It relies on local information between neighboring sensor nodes. However, its main drawback is node oscillations and hence a high amount of sensor node energy wasted. The grid based strategy divides the monitored area into virtual cells. Each cell center determines the position of a sensor node. In this paper we propose GDVFA that combines the advantages of both strategies: on the one hand coverage and connectivity for virtual forces strategy and on the other hand avoidance of node oscillations for grid strategy. Simulation results reported in this paper show that GDVFA considerably reduces the energy consumed by sensor nodes. This comes from: 1) the detection of redundant nodes that are put in sleep mode, and 2) the avoidance of node oscillations by stopping nodes.

## I. INTRODUCTION

The large majority of applications in Wireless Sensor Networks (WSNs) aim at monitoring a given area in order to collect information generated by sensor nodes. To achieve this goal, two major issues should be met. The first is area coverage: any event occurring in the monitored area should be detected. The second is network connectivity: each detected event should be reported to a central entity, the sink. Then if the deployment algorithm does not ensure the required area coverage and network connectivity, the monitoring task will fail.

Because of the importance of the coverage and connectivity in WSN, we focus on redeployment algorithms that cope with coverage and connectivity problems. Many deployment algorithms are proposed in the literature. These algorithms are based on different strategies [1]: Forces based strategy that uses virtual forces to move sensor nodes [2], [3], [4], Grid based strategy that defines a grid pattern to determine sensor node positions [5], [6], [7] and Computational geometry based strategy that relies on Voronoi polygon and Delaunay triangulation to find the appropriate sensor node positions [8], [9], [10]. Among these strategies, we are interested in the forces based strategy and in the grid based strategy.

In the forces based deployment algorithms, sensor nodes move according to virtual forces that can be attractive, repulsive or null. An attractive force is exerted, if the distance separating two sensor nodes is higher than a certain threshold, denoted  $D_{th}$ . If the distance is lower than  $D_{th}$ , a repulsive force is exerted. Otherwise, the force is null. The resultant of the virtual forces exerted by all neighboring nodes determines the new

position of this node. In the new position, the resultant of the virtual forces is computed again and the next position of nodes is determined accordingly. This mechanism is very helpful to spread sensor nodes in the whole area while avoiding coverage holes and keeping the network connectivity.

Deployment algorithms based on grid pattern also provide a good result in terms of coverage, connectivity and energy consumption. In grid based strategy, the considered area is divided into similar virtual cells according to a predefined pattern that can be triangular lattice, square or hexagon. Cell size is computed with regard to the value of both sensing range,  $r$ , and radio range,  $R$ , in order to ensure the area coverage and guarantee the network connectivity. It has been proved that the triangular lattice provides the optimal number of sensor nodes to achieve coverage when  $R \geq \sqrt{3}r$  [11]. This strategy can be used to reduce the energy consumption as the node positions are predetermined.

To take advantage of these two strategies, we propose a redeployment algorithm called GDVFA, Grid Distributed Virtual Forces Algorithm, combining the virtual forces strategy to move sensor nodes and the grid based strategy to save energy. This paper is organized as follows: Firstly, in Section II, we cite some studies related to virtual forces and grid strategies and present our motivation. Secondly, we propose a new algorithm called GDVFA which is an hybridization of virtual forces and grid based strategies in Section III. Simulation results of GDVFA are given in Section IV, evaluating the coverage rate and the total distance traveled by nodes and comparing GDVFA to DVFA. In Section V, we show how to compute the value of the spreading factor that depends on the initial deployment. Finally, we conclude in Section VI.

## II. MOTIVATIONS AND RELATED WORK

The force based strategy is well studied in WSNs. The basic Virtual Forces Algorithm, VFA, is proposed in its centralized version in [2]. VFA moves sensor nodes from dense to poorly covered area while maintaining connectivity due to the attractive and repulsive forces exerted between sensor nodes. This algorithm provides a uniform deployment since the target distance,  $D_{th}$ , between sensor nodes is the same for all neighbor nodes. However, this algorithm suffers from some drawbacks, such as the existence of a bottleneck and a single point of failure, node stacking, non full coverage and instability. That is why, many enhancements of the basic VFA, namely distributed algorithms, are proposed to cope with these problems. To avoid the node stacking problem, the coefficient

of the attractive force is reduced in [3] and [12]. Hence, a uniform deployment is obtained. Furthermore, the authors of [3] and [12] propose to reduce the amplitude of oscillations by limiting the maximum distance traveled in each iteration. With these enhancements, a high coverage rate is quickly achieved. Despite these improvements, node oscillations problem remains unsolved in the distributed algorithms. These oscillations are the source of the instability of the deployment. Hence, this instability may cause a high energy consumption. Moreover, the end of the virtual forces algorithm cannot be detected even when the maximum coverage rate is reached: a sensor node is unable to distinguish between a local maximum and a global maximum of the coverage rate. Furthermore, the principle of virtual forces does not include a mechanism to detect redundant nodes and make them sleep to save energy. In this paper, we are interested in the distributed version which is based only on local information and avoids the single point of failure.

On the other hand, grid based strategy provides many advantages, such as a uniform deployment according to a predefined pattern, a full coverage and an easy identification of redundant nodes. To cope with the problems related to virtual forces, we propose to combine virtual forces with grid. The idea is to keep the simplicity of virtual forces and to take the advantage of predefined positions assigned to sensor nodes in grid. Indeed, node oscillations that are small moves caused by repulsive or attractive forces exerted between neighbors when the distance between them is not equal to the target distance  $D_{th}$ , can be avoided using predefined node positions in grid. A first approach would be to design a grid based VFA algorithm. With this algorithm, any node computes the resultant of the virtual forces and moves to the cell center the closest to its new position, if this cell center is empty. Otherwise, it moves according to the virtual forces. To show the different behaviors of DVFA, a distributed virtual forces based algorithm and the grid based VFA algorithm, we evaluate the coverage and the total distance travelled by nodes as a function of time for a random topology of 200 nodes for a network area of  $500 * 500$ . Figure 1 shows that the grid based algorithm achieves a coverage rate significantly smaller than DVFA. However, Figure 2 depicts a total distance traveled that always increases with DVFA due to node oscillations, whereas it remains constant with the grid based algorithm.

We want to combine the very good coverage of DVFA with the small distance traveled obtained with the grid. The new algorithm we propose is called GDVFA: Grid Distributed Virtual Forces Algorithm that uses the grid to cope with node oscillations caused by virtual forces.

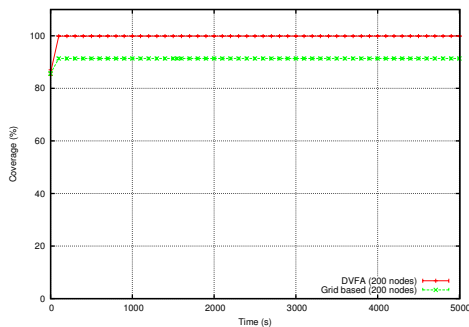


Fig. 1: Coverage rate with DVFA and a grid based algorithm.

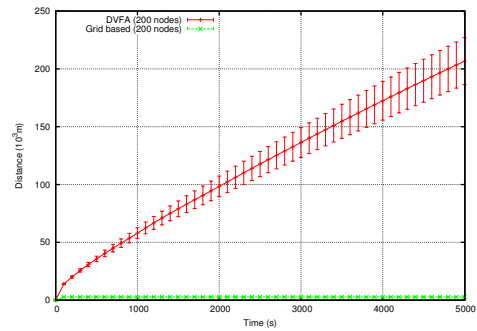


Fig. 2: Total distance traveled by nodes with DVFA and a grid based algorithm

Unlike the algorithm previously proposed, grid based VFA, GDVFA behaves like DVFA during a certain amount of time to allow the spreading of nodes in the whole area. Then, it behaves like the grid based VFA algorithm.

The main contribution of this paper is the elimination of node oscillations by stopping them once they satisfy certain conditions fixed by GDVFA. Hence, we are able to detect redundant nodes and make them sleep. Consequently, energy consumption is considerably reduced.

### III. PRESENTATION OF GDVFA

#### A. Assumptions

In this paper we adopt the following assumptions in order to ensure full area coverage and network connectivity while coping with virtual forces oscillations problem:

- Each sensor has a sensing range  $r$  and a radio range  $R$ . Furthermore, we assume that  $R \geq \sqrt{3}r$ . In such a case, it is sufficient to ensure full area coverage to get connectivity.
- Each sensor knows its own position (via GPS or other localization technology).
- The considered area is assumed to be a 2-dimension area and is divided into cells.
- Each sensor is able to determine the cell center in its surrounding.
- The network field is a rectangle of length  $L$  and width  $W$ . Each sensor knows the values of  $L$  and  $W$ .

#### B. Description of GDVFA

The GDVFA algorithm is an hybridization of the virtual forces and the grid. GDVFA, like DVFA [3], is based on virtual forces to move sensor nodes and maintain the target distance  $D_{th}$  between neighboring nodes. The new position of a sensor node is computed according to the sum of the forces exerted on it by its 1-hop and 2-hop neighbors. As we shown in a previous work [3], the node moving distance is limited to a fixed threshold called  $L_{max}$  in order to reduce the distance traveled by nodes in each iteration. The originality of GDVFA lies in the use of grid based strategy: we propose to divide the area into similar cells. Our target is to incite nodes to occupy the centers of cells. Hence, redundant nodes are nodes that do not occupy the center of a cell. They can easily be detected and switched to sleep state to save energy. Furthermore, any node whose neighboring cell centers do not change can stop moving.

So the energy consumed is reduced. These two enhancements are detailed in Subsection III-D.

GDVFA proceeds in two phases. Each phase consists of iterations. In each iteration, each node proceeds as follows:

- Step 1: each node broadcasts a *Hello*-message each *Hello*-period to discover its 1-hop and 2-hop neighbors.
- Step 2: each node computes the virtual forces exerted on it by its 1-hop and 2-hop neighbors.
- Step 3: each node determines its new position by computing the resultant force.
- Step 4: The node moves. (We will see later how.)

Each iteration has a duration of a *Hello*-period.

*Phase 1* executes the simple DVFA [3] to spread sensor nodes in the whole area while ensuring a uniform density. During this phase, each node moves in step 4 to the new position computed by the virtual forces. At the end of phase 1, the nodes are deployed in the whole area while offering a good coverage and uniform density. Nevertheless, DVFA suffers from high energy consumption due to node oscillations that may never stop.

The aim of *Phase 2* is essentially to cope with this drawback by adopting the grid strategy. In this phase, step 4 is replaced by the following step:

- Step 4': each node determines the cell containing its new position. If the center of this cell is empty and this node has the smallest identifier among all nodes in this same cell, then it moves to the cell center. Otherwise, this sensor node moves to the new position determined by the resultant force. Notice that a node that occupies the center of a cell can leave it only if its neighbors exert on it an attractive or repulsive force.

The benefit of the first phase is to spread rapidly the nodes in the whole area which lasts a predefined amount of time (the spreading factor described in V), before switching to the second phase offering stability and convergence. For example, in the performance evaluation in section IV, this spreading factor is equal to 100s for topology where nodes are randomly scattered all over the network area. Any sensor node is assumed to know the value of the spreading factor, a parameter of GDVFA.

### C. Cell definition

In this section, we explain how to define the virtual grid in GDVFA by giving the equations used to determine the cell center relative to a sensor node of coordinates  $(x, y)$  in the grid. The size of cells is computed with regard to the sensing range in order to ensure the full area coverage. The authors in [13] proved that when  $R \geq \sqrt{3}r$  and the full area coverage is ensured, network connectivity is consequently ensured. Then, in our work, network connectivity is guaranteed since we assume  $R \geq \sqrt{3}r$ .

In GDVFA, we choose to use rectangular cells to simplify the computation of the cell to which any node belongs to, knowing the coordinates  $(x, y)$  of the sensor node, the values of  $L$  and  $W$  of the network area considered and the value

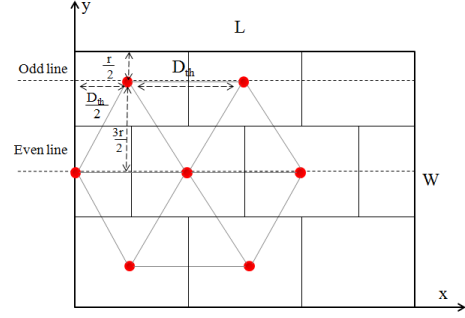


Fig. 3: Grid in GDVFA.

of the sensing range,  $r$ . Sensor nodes should occupy the center of these cells. Figure 3 depicts the grid cells with a rectangular shape. Each non-border cell has a length equal to  $D_{th}$  and a width of  $\frac{3r}{2}$ . Furthermore, each non-border cell has 6 neighboring cells. However, sensor nodes are deployed in triangular lattice. These triangles of edge  $D_{th}$  are computed as follows. As the GDVFA algorithm deploys sensor nodes in a triangular lattice, then each three neighboring nodes, vertices of the same triangle, should fully cover this equilateral triangle while minimizing their overlapping. This overlapping can be a single point which corresponds to the barycenter of the triangle. Since the distance between the barycenter of this triangle and each vertex of this triangle is  $r$ , the distance between two neighboring nodes, an edge of the triangle, is  $D_{th} = \sqrt{3}r$ . This also corresponds to the length of the cell. The width of the cell is equal to the height of the equilateral triangle  $\frac{3r}{2}$ .

The cell centers of the same floor constitute a single line. We distinguish between odd and even lines. In any odd line, the first sensor node should be located at a distance  $\frac{D_{th}}{2}$  from the left border line. Whereas, in any even line the first sensor node is located on the left border line. Border cells are a particular case. They are built to ensure full area coverage while minimizing the number of sensor nodes. That is why the first line is at a distance  $\frac{r}{2}$  from the top border line.

Let  $(x, y)$  be the coordinates of any sensor node  $s_i$  in the grid. We want to determine to which cell  $s_i$  belongs to. This cell is defined by a line number, denoted  $line(x, y)$ , and a column number, denoted  $col(x, y)$ .

$$line(x, y) = \lfloor \frac{y - \frac{r}{2}}{\frac{3r}{2}} \rfloor + 1 + \delta_l \quad (1)$$

$$with \delta_l = \begin{cases} 1 & \text{if } y - \frac{5r}{4} - \lfloor \frac{y - \frac{r}{2}}{\frac{3r}{2}} \rfloor \frac{3r}{2} > 0 \\ 0 & \text{otherwise} \end{cases}$$

For the computation of the column, we distinguish two cases:

- Case  $(line(x, y) \bmod 2) = 0$ :

Then, the node of coordinates  $(x, y)$  occupies the  $col_e(x, y)^{th}$  cell in an even line computed as follows:

$$col_e(x, y) = \lfloor \frac{x}{\sqrt{3}r} \rfloor + 1 + \delta_e \quad (2)$$

$$with \delta_e = \begin{cases} 1 & \text{if } x - \frac{D_{th}}{2} - \lfloor \frac{x}{\sqrt{3}r} \rfloor D_{th} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Hence, the coordinates of the cell center are  $(x_c, y_c)$

$$\text{with } \begin{cases} x_c = (\text{col}_e(x, y) - 1)D_{th} \\ y_c = (\text{line}(x, y) - 1)\frac{3r}{2} + \frac{r}{2} \end{cases}$$

- Case  $(\text{line}(x, y) \bmod 2) \neq 0$ :

Then the node of coordinates  $(x, y)$  occupies the  $\text{col}_o(x, y)^{th}$  cell in an odd line computed as follows:

$$\text{col}_o(x, y) = \lfloor \frac{x - \frac{D_{th}}{2}}{D_{th}} \rfloor + 1 + \delta_o \quad (3)$$

$$\text{with } \delta_o = \begin{cases} 1 & \text{if } x - D_{th} - \lfloor \frac{x - \frac{D_{th}}{2}}{D_{th}} \rfloor D_{th} > 0 \\ 0 & \text{otherwise} \end{cases}$$

and  $D_{th} = \sqrt{3}r$ .

Hence, the coordinates of the cell center are  $(x_c, y_c)$

$$\text{with } \begin{cases} x_c = (\text{col}_o(x, y) - 1)D_{th} + \frac{D_{th}}{2} \\ y_c = (\text{line}(x, y) - 1)\frac{3r}{2} + \frac{r}{2} \end{cases}$$

#### D. Stopping condition and detection of redundant nodes

By definition a node is said to be in stop state in an iteration if and only if it does not move in this iteration due to the stopping condition. However to keep the required property of reactivity to topology changes (e.g. node departure, empty cells detected), the stopping condition is always verified at each iteration.

*Stopping condition: the nodes occupying the center of its 6 neighboring cells and its cell have not changed during three consecutive iterations.*

At each iteration, any node computes the resultant of the virtual forces exerted on the node, and checks the stopping condition. If the node is not stopped and the stopping condition is true, the node stops. Furthermore, a previously stopped node moves in an iteration if and only if: either the resultant of the virtual forces differs from the previous one (e.g. arrival of a new neighbor), or the stopping condition is no more true.

A node that has stopped without occupying a cell center is said to be *redundant*. Redundant nodes are used to replace failed or depleted nodes. For the initial deployment depicted in Figure 4a, GDVFA, provides the final deployment shown in Figure 4b for 250 nodes. Nodes in red are active nodes, whereas blue nodes with small points are redundant nodes which sleep to save energy and prolong network lifetime.

## IV. PERFORMANCE EVALUATION

In this section, we present the parameters we adopted to evaluate the performances of GDVFA in various random topologies.

### A. Simulation parameters

The parameters used in the simulations are defined in Table I. Notice that the optimal number of sensor needed to provide full coverage of the network area is 178.

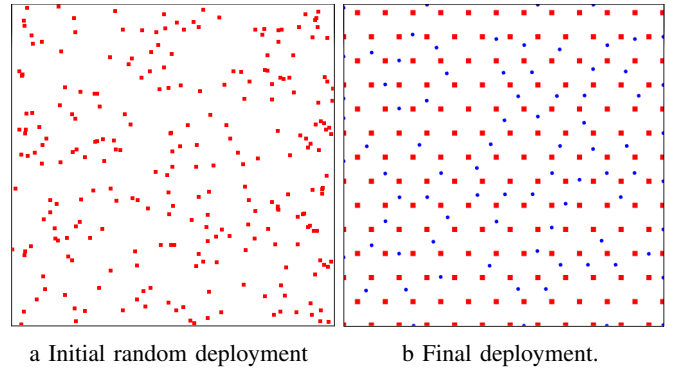


Fig. 4: GDVFA with 250 nodes.

TABLE I: Simulation parameters

<i>Topology</i>	
Sensor nodes	200, 250, 300 randomly spread in the whole area
Area size	500m x 500m
Speed	5m/s
<i>Simulation</i>	
Result	average of 30 simulation runs
Simulation time	5000s
<i>MAC</i>	
Protocol	IEEE 802.11b
Throughput	2 Mb/s
Radio range $R$	50 m
Sensing range $r$	25 m
<i>DVFA &amp; GDVFA</i>	
$K_a$	0.001
$K_r$	0.56
Hello period	2s
$L_{max}$	$D_{th}/6$
Spreading factor	100s for nodes

### B. Comparative evaluation between DVFA and GDVFA

From Figure 5, we observe that GDVFA reaches a 100% coverage rate, while considerably reducing the total distance traveled by nodes as depicted in Figures 6, 7 and 8, for different numbers of nodes 200, 250 and 300. The total distance traveled by nodes increases with time and with the number of nodes in DVFA (see Figures 6, 7 and 8).

Notice that the slope of the total distance curve is much steeper when the number of nodes exceeds the optimal number needed to fully cover the network area. This can be explained by the fact that DVFA tries to enforce equilateral pattern deployment, where the edge of the triangular is  $D_{th}$ , computed for the optimal number of sensors. Hence, the node oscillations increase, when the number of sensors increase beyond this optimal number. Notice that nodes continue to move even when the 100% coverage rate is reached. Unlike DVFA, GDVFA provides very good performances with regard to the distance traveled: this distance remains constant a short time after the end of the Phase 1, because GDVFA eliminates node oscillations by stopping nodes. Hence, with GDVFA energy consumption is reduced.

Simulation results show that the stopping condition is very efficient. In fact, with GDVFA all nodes stop whatever the number of nodes (see Figure 9).

We can distinguish two types of stopped nodes:

- the ones occupying the center of cells. These nodes are needed to ensure full coverage;
- the other ones are redundant nodes (see Figure 10).

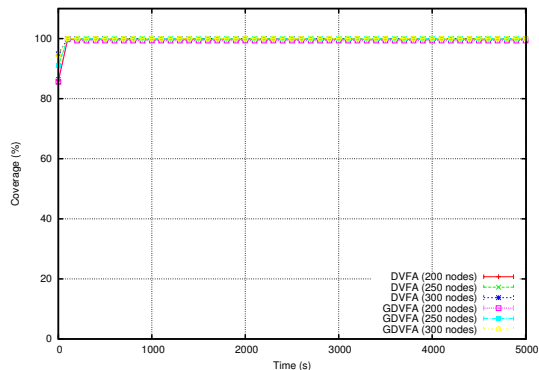


Fig. 5: Coverage with GDVFA and DVFA.

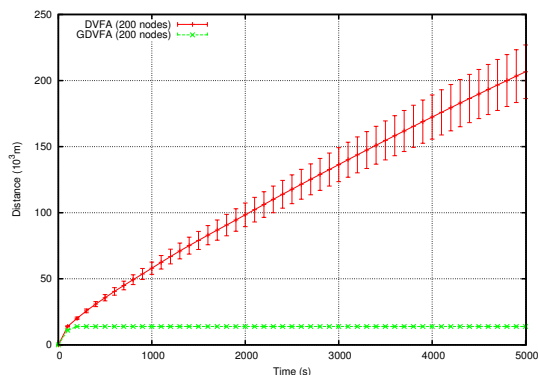


Fig. 6: Total distance traveled by nodes with GDVFA and DVFA - 200 nodes.

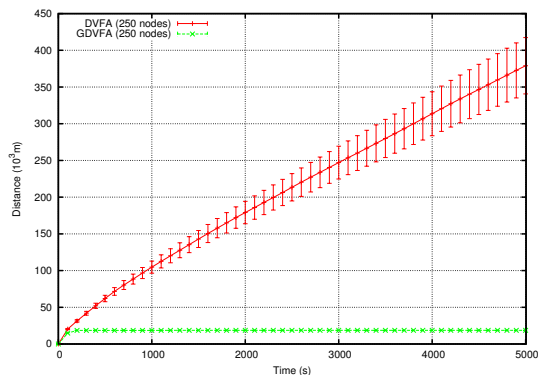


Fig. 7: Total distance traveled by nodes with GDVFA and DVFA - 250 nodes.

GDVFA enables the distributed detection of redundant nodes (see Figure 10). Such nodes can be turned off to save energy and increase network lifetime. The detection of redundant nodes can also be used to repair coverage holes or replace energy depleted nodes. Figure 11 depicts the cumulative stopping time that allows nodes to save energy by reducing useless moves and hence prolong network lifetime.

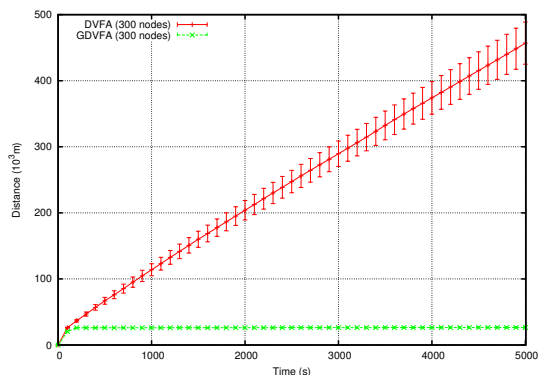


Fig. 8: Total distance traveled by nodes with GDVFA and DVFA - 300 nodes.

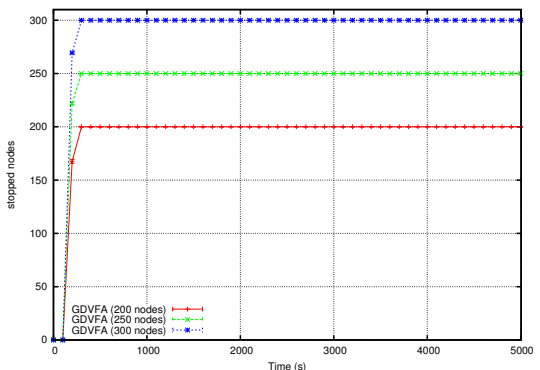


Fig. 9: Number of stopped nodes.

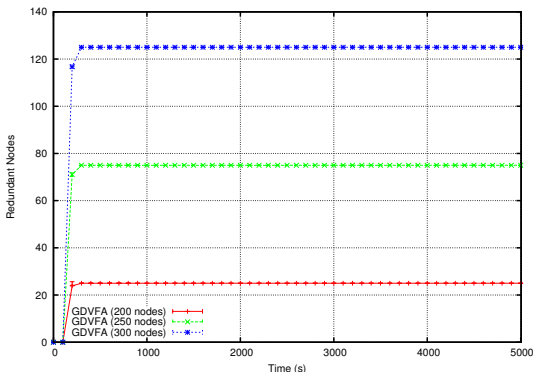


Fig. 10: Number of redundant nodes.

## V. IMPACT OF THE SPREADING FACTOR

Up to now, we have assumed that the initial deployment was random and sensor nodes were scattered in the whole network area. Simulation results reported in Section IV, show that  $f = 100s$  is sufficient to obtain a final coverage rate of nearly 100% at the end of the simulation. For many applications, this initial deployment is not representative; initially all sensor nodes are grouped together in a single entry point of the area. In this case,  $f = 100s$  may be insufficient to obtain the full coverage, even if the number of sensors is sufficient to obtain this full coverage. We can establish a lower bound for this factor in such initial topology as follows. Let  $W * L$  be the length and the width of the rectangular area. In DVFA and

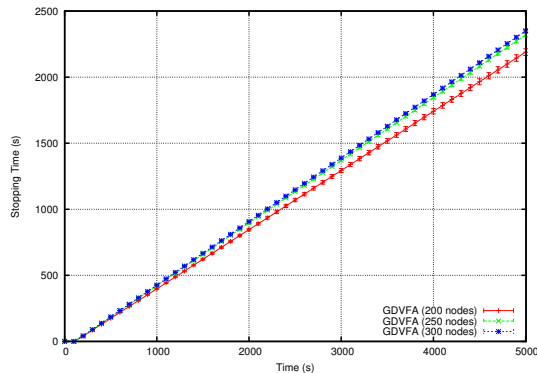


Fig. 11: Cumulative stopping time.

GDVFA, any sensor node cannot move more than  $L_{max}$  at each *Hello*-period. The lower bound on the spreading factor is computed, taking into account the time needed by a sensor to reach the farthest position from the initial one in the topology considered. For instance, in a topology with a single entry point at the corner of the rectangular area, which can be considered as one of the worst cases, we get:

$$\frac{\sqrt{L^2+W^2}}{L_{max}} * Hello\ period$$

With the simulation parameters given in Table I, we get  $f \geq 194s$ . We now study the impact of the spread factor on the performances achieved by GDVFA. Figure 12 shows that a spreading factor of 100s is not sufficient to ensure the spreading of nodes in the whole area. Consequently, the area coverage remains limited to 63%. A spreading factor larger than 200s allows to reach a 100% coverage rate. As illustrated in Figure 13, the total distance traveled by nodes increases with the spreading factor. This is due to the oscillations caused by DVFA, that occur even when the full area coverage is obtained. As a conclusion, the choice of the spreading factor is very important for the performances of GDVFA, expressed in terms of coverage rate and distance traveled. To save energy, we recommend to choose the smallest value ensuring the full coverage, (e.g. 250s in our example).

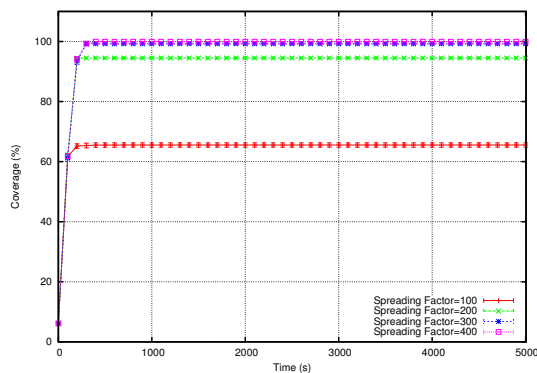


Fig. 12: Impact of the spreading factor on the coverage rate with GDVFA.

## VI. CONCLUSION

In this paper, we propose a new distributed redeployment algorithm that is an hybridization of the virtual forces and the grid based strategies. GDVFA achieves a very good coverage rate of DVFA that favors the uniform spreading of nodes in the whole network area. GDVFA eliminates the node oscillations problem by stopping nodes thanks to its grid based component. Moreover, GDVFA can detect redundant nodes in a distributed

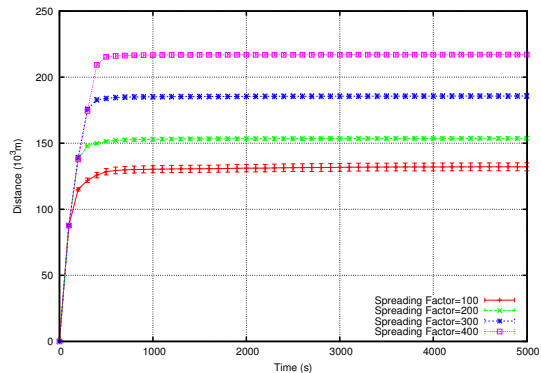


Fig. 13: Impact of the spreading factor on the total distance traveled by nodes with GDVFA.

way. These redundant nodes can be used to replace battery depleted nodes, otherwise they can switch to sleep state to save energy. We have conducted an extensive performance evaluation with two different initial topologies: a random one where nodes are scattered all over the network area and a topology where all nodes are grouped at a single entry point. These simulations show the merit of the two phases of GDVFA. We have also determined how to choose the spreading factor depending on the initial topology considered, the dimensions of the area to cover and the speed of nodes.

## ACKNOWLEDGMENT

This work has been partially funded by the Cluster Connection project.

## REFERENCES

- [1] N. A. Ab. Aziz, K. Ab. Aziz, and W. Z. W. Ismail, "Coverage Strategies for Wireless Sensor Networks", World Academy of Science, Engineering and Technology, 2009.
- [2] Y. Zou and K. Chakrabarty "Sensor Deployment and Target Localization Based on Virtual Forces" IEEE INFOCOM 2003.
- [3] K. Mougou, S. Mahfoudh, P. Minet and A. Laouiti "Redeployment of Randomly Deployed Wireless Mobile Sensor Nodes" IEEE VTC, 2012.
- [4] S. Mahfoudh, I. Khoufi P. Minet and A. Laouiti "Relocation of Mobile Wireless Sensors in the Presence of Obstacles" ICT, Marroco, 2013.
- [5] S. Han, Y. Zhang and G. Xu, "Hexagonal grid-based sensor deployment algorithm ", Control and Decision Conference, Chinese, 2010.
- [6] P. Park, S. G. Min and Y. H. Han "A Grid-based Self-deployment Schemes in Mobile Sensor Networks" 2010.
- [7] L. Xu, A. Nayak and I. Stojmenovic, "Back Tracking based Sensor Deployment by a Robot Team" USA, 2010.
- [8] G. Wang, G. Cao and T. F. La Porta "Movement Assisted Sensor Deployment" IEEE Transactions on Mobile Computing, 2006.
- [9] S. Babaie and S. S. Pirahesh "Hole Detection Fore Increasing Coverage in Wireless Sensor Network Using Triangular Structure" IJCSI, 2012.
- [10] H. Tan, Y. Wang, X. Hao, Q. S. Hua and F. C.M. Lau "Arbitrary obstacles constrained full coverage in wireless sensor networks" In Wireless Algorithms Systems and Applications, Springer Berlin Heidelberg, 2010.
- [11] X. Bai, S. Kumar, D. Xuan, Z. Yun and T.H. Lai "Deploying Wireless Sensors to Achieve Both Coverage and Connectivity" MobiHoc, 2006.
- [12] J. Li, B. Zhang, L. Cui and S. Chai, "An Extended Virtual Force-Based Approach to Distributed Self-Deployment in Mobile Sensor Networks", International Journal of Distributed Sensor Networks, 2012.
- [13] X. Wang, Y. Zhang, C. Lu, R. Pless and C. Gill "Integred Coverage and Connectivity Configuration in Wireless Sensor Networks" SenSys, Embedded networked sensor systems, p. 28-39, 2003.