

Research Article

Efficiency Evaluation of Strategies for Dynamic Management of Wireless Sensor Networks

Andrea Verónica González, Lisane Brisolara, and Paulo R. Ferreira

CDTec, Universidade Federal de Pelotas, Rua Gomes Carneiro, 1, Centro, 96010-610 Pelotas, RS, Brazil

Correspondence should be addressed to Paulo R. Ferreira; paulo@inf.ufpel.edu.br

Received 31 July 2016; Accepted 17 November 2016; Published 17 January 2017

Academic Editor: Dieter Hogrefe

Copyright © 2017 Andrea Verónica González et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents and evaluates dynamic management strategies to improve efficiency in event-triggered wireless sensor networks. We are considering mobility, where nodes move themselves to maximize the coverage, and load balancing state-of-the-art techniques, by which the number of nodes sensing the same area is reduced. To explore mobility, we present a simple method by which nodes can dynamically reorganize themselves based on the force fields approach of mobile robotics. Firstly, the strategies are evaluated separately through experiments with different network configurations and, afterwards, a joint evaluation has been conducted to observe the impact of mobility on the efficiency of load balancing techniques. We show that mobile nodes significantly contribute to keeping the coverage as nodes die in mesh and powerfully improving it in random deployments. Load balancing techniques achieve important results, increasing lifetime and the number of sensed events. However, in random deployments, these techniques lose efficiency and become unsuitable strategies. Combining these strategies with mobility, we observe that PS-based technique keeps its contribution in mesh and random deployments, as well as improving its performance for not so dense networks. Ant-based technique when combined with mobile nodes loses performance significantly in mesh and keeps its good performance in random deployed and less dense networks.

1. Introduction

Wireless sensor networks (WSN) consist of multiple nodes distributed in the area of interest in order to sense phenomena that happen inside this area. Usually, these nodes are low-cost products with limited resources and have a nonrechargeable battery. The network lifetime or its service availability depends on the discharge of nodes batteries.

The deployment of nodes determines their positions and the sensing coverage achieved by the WSN. However, in environment monitoring, it is sometimes not possible to deploy nodes in a specific location and sometimes nodes must be launched from an airplane. This nonuniform sensor nodes distribution might lead to coverage holes in the network [1]. As, in these cases, an initial random deployment is obtained, a dynamic reorganization strategy can be helpful to spread nodes into the desired area, moving them according to a given strategy. The mobility of nodes can be explored for deploying

nodes in the area of interest maximizing coverage [1–3]. Besides, the movement can be also adopted to reorganize nodes in such a way that a working node can replace a dead node in order to keep or restore coverage.

On event-triggered or reactive WSN, common network solutions include redundancy to achieve desired area coverage. As more than one node can sense the same event, load balancing techniques can improve the network lifetime and service availability avoiding that the same event can be processed by more than one node [4, 5]. Thus, load balancing is considered a relevant dynamic management strategy to achieve lifetime efficiency on event-triggered WSN.

Different strategies can be used to dynamically manage WSN's resources, aiming to optimize their usage. Researchers have proposed adaptations that can on-the-fly decide which kind of services a node will provide based on its battery level, or balance the load among nodes [4, 5], aiming to increase lifetime, or yet move nodes to improve coverage



[2]. Since several management strategies can be adopted, designers should decide which strategy applies to a given WSN configuration.

When evaluating dynamic management strategies, several aspects should be considered such as position of nodes, nodes communication, and sensing radius as well as redundancy (or density), which affect the network coverage and overall efficiency. On event-triggered WSN, beside these aspects, the workload models appropriated for the application scenarios should be adopted [6–8] in order to estimate efficiency.

Eboracum has been proposed in [6, 7] as an extensible framework for event-triggered WSN modeling and simulation at high abstraction level, which estimates the energy consumption adopting stochastic workload models. The provided primitive allows modeling and simulating WSN, providing useful metrics as lifetime and number of sensed events. The framework can be easily extended to specialize available primitives, supporting the implementation and evaluation of different WSN configurations.

In this work, we present and evaluate two different classes of dynamic management strategies for WSN, analyzing its impact on relevant efficiency metrics. The evaluated strategies were implemented using Eboracum framework. Firstly, we evaluate the adoption of sensor mobile nodes, which are dynamically reorganized in order to improve coverage. Secondly, we evaluate load balancing strategies targeted to improve lifetime. Furthermore, we combine mobile nodes with two bioinspired load balancing techniques, analyzing the impact of mobility on the performance achieved by these techniques.

The paper is organized as follows. Section 2 discusses previous work on dynamic management strategies for WSN. Section 3 introduces our mobility strategy, explaining its formal definition as well as implementation. The load balancing strategies are detailed in Section 4, describing also their implementation. Section 5 presents and discusses results achieved by the different dynamic management strategies as well as results obtained by combining load balancing with mobile nodes. Section 6 concludes and gives some remarks.

2. Related Work

2.1. Mobile Nodes. The mobility of nodes in WSN has been investigated by researchers, varying the kind of supported mobile elements from sensor nodes, to data collectors, or to intermediate nodes, or yet to sink nodes. According to [9] most of them exploit mobility to address the problem of data collection in WSN as in [10, 11].

In [10] a protocol is proposed to manage data collector nodes in WSN. To provide a longer lifetime and increase data collection rates, the research community has exploited the use of mobile sinks as in [11].

However, mobile nodes can be also employed to elevate sensing coverage or yet network connectivity. In [2, 3, 12] authors propose strategies to spread sensor nodes aiming to improve sensing coverage. As the former, this work also proposes a strategy to dynamically maximize coverage considering mobile sensor nodes. Differently from the others,

we evaluate our strategy in the context of event-triggered WSN, instead of proactive ones.

Recently, a distributed deployment algorithm for WSN has been proposed to increase coverage [1]. The proposed solution is iterative and based on geometry, using the Voronoi diagram to partition the area among sensors. This diagram represents the nearest information about a set of sensors, which is used to identify coverage holes and relocate nodes.

In [12], mobile nodes are employed in a decentralized dynamic deploying approach based on force fields and focused on achieving maximal coverage in unknown environments while keeping the communication. Their approach is focused on motion control to be applied to real mobile nodes and cannot be directly employed in WSN simulation environments. Based also on the theory of force fields, we propose here a dynamic reorganization strategy to improve coverage. Moreover, we evaluate our strategy regarding the number of sensed events and lifetimes, analyzing the discharging of nodes batteries based on an event-triggered probabilistic workload model.

Here, we evaluated the efficiency of mobile nodes, varying the WSN configurations, considering mesh and random initial deployments, and moving nodes initially to improve coverage and afterwards to keep coverage when nodes stop to work. Our approach is similar to the one proposed by [1], but our procedure to compute the new position is simpler and has lower computation cost. In [1], the proposed deployment algorithm has been evaluated using only random deployment, without comparison to mesh ones. Besides, differently from our work, authors do not consider the discharging of nodes batteries and thus do not explore the mobility to keep coverage after nodes start to die.

2.2. Load Balancing. Another relevant WSN dynamic management strategy is the load balancing. Works commonly face the load balancing in this domain as a problem of communication among nodes, presenting methods to better routing the data traffic through the network [13–16]. It happens because they consider that the network senses the environment in a proactive way, which means that all nodes have the same tasks to process. However, load balancing is mainly fruitful for event-triggered WSN where the coverage redundancy can be explored in order to save nodes battery.

When targeting event-triggered WSN applications, the load balancing should explore the coverage redundancy since several sensor nodes can be triggered by the same event but only one has to process the tasks it produces. The problem of task mapping on distributed systems has been investigated, considering different platforms as multicores, as well as sensor networks. The approaches for task mapping can be divided in static [17, 18] and dynamic approaches [4, 5, 8, 19]. Static approaches are limited since they cannot effectively adapt to network conditions.

Different approaches have been proposed for dynamic WSN load balancing, which mainly differs among each other by the adopted heuristics such as genetic algorithms [8] and bioinspired ones [19]. In these works, centralized solutions have been proposed to balance energy usage while extending the network lifetime.



Moreover, recently, works have focused on distributed solutions given the distributed nature of WSN. In [5], a bio-inspired technique based on the bee's hormonal system has been proposed named Pheromone Signaling (PS). Through this technique, nodes are periodically differentiated from other nodes to distribute the workload. Some nodes, called Queen Nodes, are allowed to process the sensed events, while remaining ones, called Worker Nodes, stay in stand-by. Nodes differentiate themselves through a periodic transmission of pheromone by Queen Nodes and its retransmission by recipients to their neighborhood.

Recently, in [4], the Ant-based load balancing technique has been proposed to allow WSN nodes to decide individually which event to process triggered by the events emergence on the environment. Following this technique, nodes decide probabilistically, on-the-fly, which events to perform applying a decision process inspired by the Ant's theoretical model of response threshold. The idea behind this technique is to divide the work among the nodes sensing the same area, considering the number of nodes who sensed an event at the same time and the number of times a given node was previously engaged in events processing. This previous work employs Eboracum and evaluates these two bioinspired load balancing techniques but considers only networks composed of static sensor nodes.

Here, we evaluate the impact of sensor nodes mobility on the sensing coverage of event-triggered WSN considering mesh and random nodes deployments using also Eboracum simulation models. Our paper also compares results achieved by the mobility strategy against the load balancing techniques ones. Furthermore, our paper combines mobile nodes to the load balancing strategies and evaluates its impact on the WSN efficiency. To the best of our knowledge, no work has been reported that combines these different dynamic management strategies in the context of event-triggered WSN.

3. Mobile Dynamic Reorganized Nodes

This section details how we explore the mobility of sensor nodes in order to improve coverage. Firstly, the proposed strategy is formally defined in Section 3.1. The implementation of the support for mobile nodes in Eboracum and the proposed dynamic reorganization are described in Section 3.2.

3.1. Proposed Reorganization Strategy. Potential fields are a well-known and applied approach to allow mobile robots navigate in unknown environments avoiding obstacles [20, 21]. Through this approach robots are seen as particles subject to virtual forces. These forces repel the robot from other robots and from the obstacles driving it through a clear path. In [12] this general idea has been applied to spread out mobile nodes through the area of interest. Reference [12] explores how to control and move the nodes according to their approach, allowing their applicability to real mobile nodes. Next we will present a simple method based on force fields to achieve maximal area coverage, which has been developed to be easily applied to mobile sensor networks running on a simulation environment like Eboracum. We

assume that mobile nodes running our strategy are able to measure angle and distance of near obstacles and control its motion to autonomously achieve a specific position in the area of interest.

Let us define J as the set of all network nodes and J_i as the set of nodes inside the communication radius ρ of the node i , henceforward called neighbors nodes, where $i \notin J_i$. For the sake of simplification, we consider the 4 area borders as 4 dummy neighbor nodes which may also be in J_i , provided that node i has the respective border inside its communication radius. This is done to keep nodes inside the area of interest. Each node i is subjected to one repelling two-dimension (2D) force vector F_i^j to each node $j \in J_i$. We set the center of the coordinate axes as the position of node i and compute all other nodes positions, including the dummy ones, relative to it. As forces acting at some angle from the coordinate axes can be decomposed into components, we compute F_i^j through its components Fx_i^j and Fy_i^j , as shown in (1) and (2), respectively, where β is a configuration constant, α is the angle formed between F_i^j and the x -axes, and $d(i, j)$ is the distance between i and j . The coordinates (x, y) of each node's center and Pythagoras Theorem are used to find the distance between i and j . Figure 1 illustrates the repelling force behavior according to the distance between nodes, adopting $\rho = 160$ and $\beta = 30$. The value of β has empirically been determined through previous experiments which evaluated the network coverage after equilibrium, while the communication radius value is used as ρ parameter:

$$Fx_i^j = \rho \cdot \exp\left(-\frac{d(i, j)}{\beta}\right) \cdot \cos(\alpha), \quad (1)$$

$$Fy_i^j = \rho \cdot \exp\left(-\frac{d(i, j)}{\beta}\right) \cdot \sin(\alpha). \quad (2)$$

The resultant force R_i , also defined by its components, is given by (3). To set the new position for the node i , we add the values Rx_i and Ry_i to the coordinates (x, y) of i , respectively,

$$\begin{aligned} Rx_i &= \sum_{j \in J_i} Fx_i^j, \\ Ry_i &= \sum_{j \in J_i} Fy_i^j. \end{aligned} \quad (3)$$

Figure 2 illustrates a hypothetical scenario where a node i , in the top right corner of an area of interest, is under the neighborhood of three nodes. Two of them are the dummy nodes j_2 and j_3 , representing the up and right borders, and j_1 is an ordinary node. Thus, i is subject of three repelling forces $F_i^{j_1}$, $F_i^{j_2}$, and $F_i^{j_3}$, by which the resultant force R_i is computed as explained before. Node i will be moved to its new position represented by i' .

Nodes can start at any position in the area of interest, randomly deployed or concentrated in a same place. After few interactions the entire area will be covered as much as possible. As nodes move according to the potential field, all



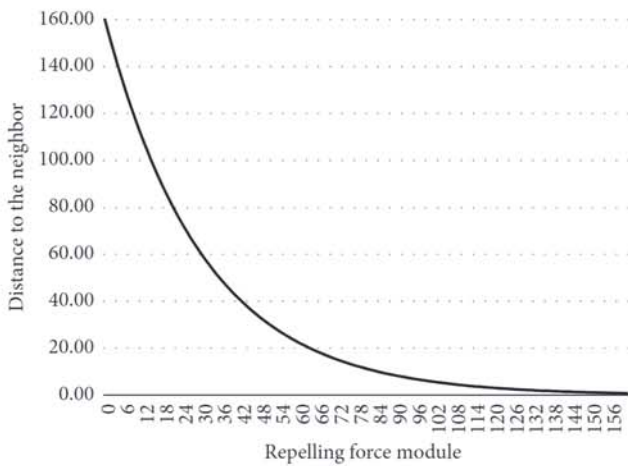


FIGURE 1: Repelling force according to the distance to the neighbor.

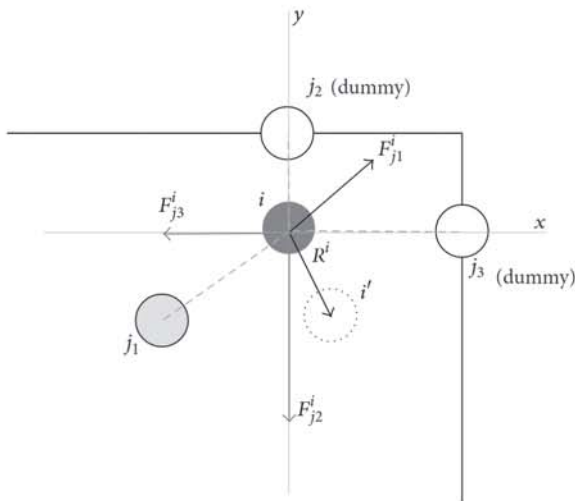


FIGURE 2: Proposed movement strategy illustrated.

repelling forces will tend to zero and the network as a whole will reach a static equilibrium.

3.2. Mobile Nodes in Eboracum. In Eboracum, the abstract class *BasicWirelessSensorNode* defines operations and attributes used for all WSN nodes. Static nodes are represented by the class *SimpleWSNNode*. To support mobile nodes, a new primitive named *BasicMobileWSNNode* has been inserted into the framework, extending *BasicWirelessSensorNode*.

The mobility strategy described here has been implemented in a specialization of the *BasicMobileWSNNode*, called *DynamicReorganizedMobileWSNNode*. It represents a node that moves according to our proposed method controlled by the *MovementRemoteController* actor. We called this actor “remote controller” because it sends movement commands to the mobile nodes, informing the next position after computing all forces. Basically, the *MovementRemoteController* works with a list of *DynamicReorganizedMobileWSNNodes* which are selected one by one to compute the repelling forces they are subjected to each time. This process

is conducted at every 10 seconds until the system achieves the equilibrium. From that moment, *DynamicReorganizedMobileWSNNode* will be reactivated again only when nodes are running out of battery and when nodes will move to replace the dead nodes covering their sensing area. Figure 3 depicts the proposed primitives hierarchy.

Figures 4(a) and 4(b) illustrate the applicability of the method proposed here, considering a WSN with 64 nodes covering a square 810 km² area and using an Eboracum-based simulation. These sensor nodes (illustrated as green circles) will be connected to compose a multihop network. Black lines show the communication paths to the sink node (black circle). Sensor nodes are randomly deployed following a uniform distribution of probability at the start of the simulation (Figure 4(a)). After that, these nodes move according to the proposed method and, in a few iterations, nodes stop moving as the system reaches the equilibrium (Figure 4(b)).

One can see that at the simulation beginning there are several uncovered locations inside the area of interest and the network is not entirely connected, once several nodes are out of the communication radius of the connected nodes. When the network reaches equilibrium, as shown in Figure 4(b), almost the entire area of interest is covered, all nodes are connected, and the network is fully functional.

4. Distributed Dynamic Load Balancing

The WSN lifetime or its service availability is dependent on the discharge of nodes batteries. To achieve desired coverage area, WSN deployments include redundancy, which motivates the adoption of dynamic strategies to balance the load, saving energy and consequently increasing the network lifetime. These strategies can be more effective on event-triggered WSN applications, avoiding that more than one node senses and processes the same event. This section revises two bioinspired techniques recently proposed for load balancing in event-triggered WSN. Furthermore, this section also discusses how these techniques were implemented at Eboracum.

4.1. State-of-the-Art Techniques. Triggered by the events emergence on the environment, the Ant-based load balancing technique [4] enables WSN nodes to autonomously decide, on-the-fly, which event to perform. The adopted decision process is inspired by the Ants theoretical model of response threshold. This technique balances the work among the nodes sensing the same area, considering the number of nodes which sensed a given event at the same time and the number of times a given node was previously engaged in events processing.

An event occurrence produces stimulus for the nodes sensing the event's location, which is computed based on the number of neighbors that sensed the event obtained by communication and the maximum number of neighbors that could have sensed the same event. Besides, each node has an internal response threshold that changes over time, capturing the node's working history. This threshold manages the node sensitiveness in such way that node becomes more sensitive



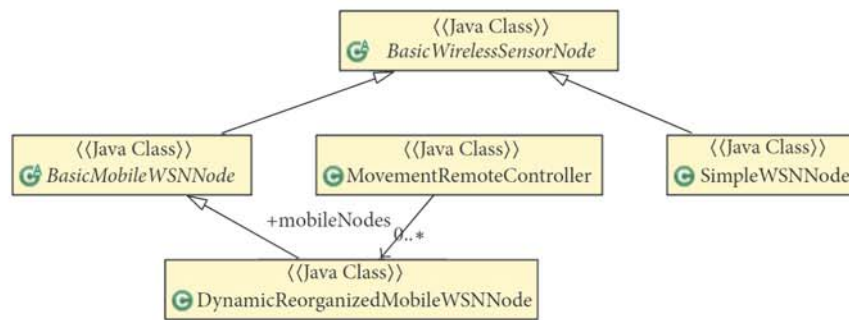


FIGURE 3: Proposed primitives to support the evaluated mobile strategy in Eboracum.

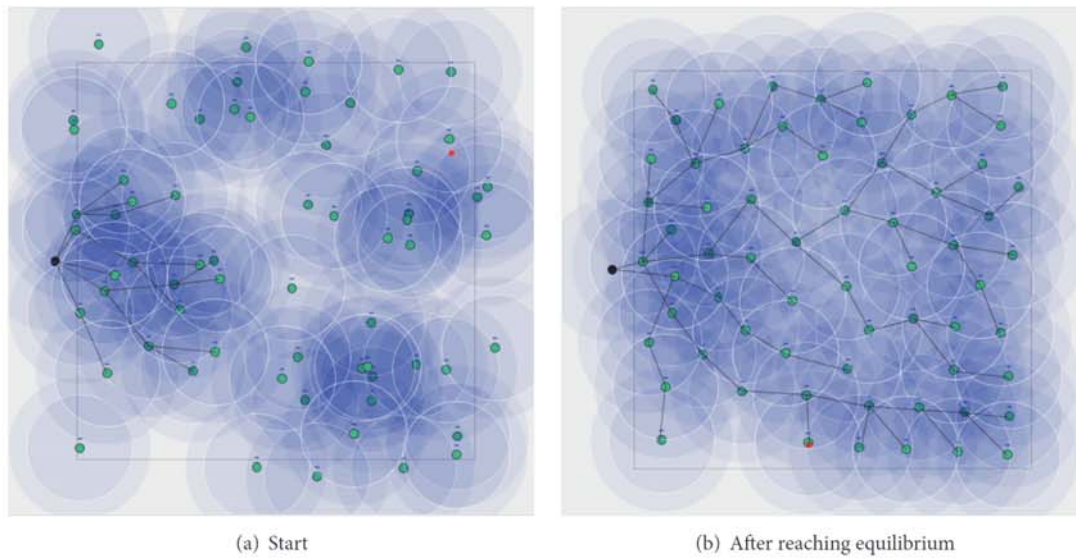


FIGURE 4: Proposed movement strategy: example with 64 nodes randomly deployed.

to the stimulus of an event it is performing and less sensitive to the other events. Thus, when a node detects an event, the node will decide to process or not this event according to its probability computed using the event's stimulus and its internal response threshold.

The Pheromone Signaling (PS) technique [5] is inspired on the bees hormonal system that ensures every beehive has only one queen. When the PS technique is used, nodes are classified as Queen Nodes and Worker Nodes according to a differentiation process. Queen Nodes are allowed to process the sensed events, while Worker Nodes stay in stand-by to save energy. The idea is to periodically change the nodes role in such a way that the workload is balanced on the network. This differentiation process is given by a periodic transmission of pheromone by Queen Nodes to inhibit neighbors from turning queens and its retransmission by recipients to their neighborhood. This retransmission is limited to a number of hops and thus PS limits the range of influence of each Queen Node, aiming to keep coverage, as much as possible. Besides, the amount of pheromone is decreased each time it is retransmitted. Following this technique, nodes became a Queen Node if they do not receive

enough pheromone from neighbors to keep their internal pheromone level above their internal threshold. When a node receives pheromones from their neighbors, its internal pheromone level increases and to compensate it, the internal pheromone level of all nodes is periodically decreased.

4.2. Load Balancing in Eboracum. To support load balancing, the Eboracum *SimpleWSNNode* primitive has specialized, creating the abstract class named *ControlledWSNNode* (Figure 5). This class represents a kind of node that is associated with an agent responsible to control it. The agent is represented in the framework by an interface named *BasicAgent* that should be implemented by classes representing a specific kind of agent. When a node senses an event, it delegates the decision to process the event to its agent. The agent should apply some load balancing technique to make this decision.

Besides the controller agent, sophisticated approaches, like PS and Ant-based, demand significant changes in the nodes behavior. Thus, to implement and evaluate the different load balancing techniques, two subclasses of controlled nodes have been defined, which are *PSControlledWSNNode* and *AntControlledWSNNode*, and their respective agents named



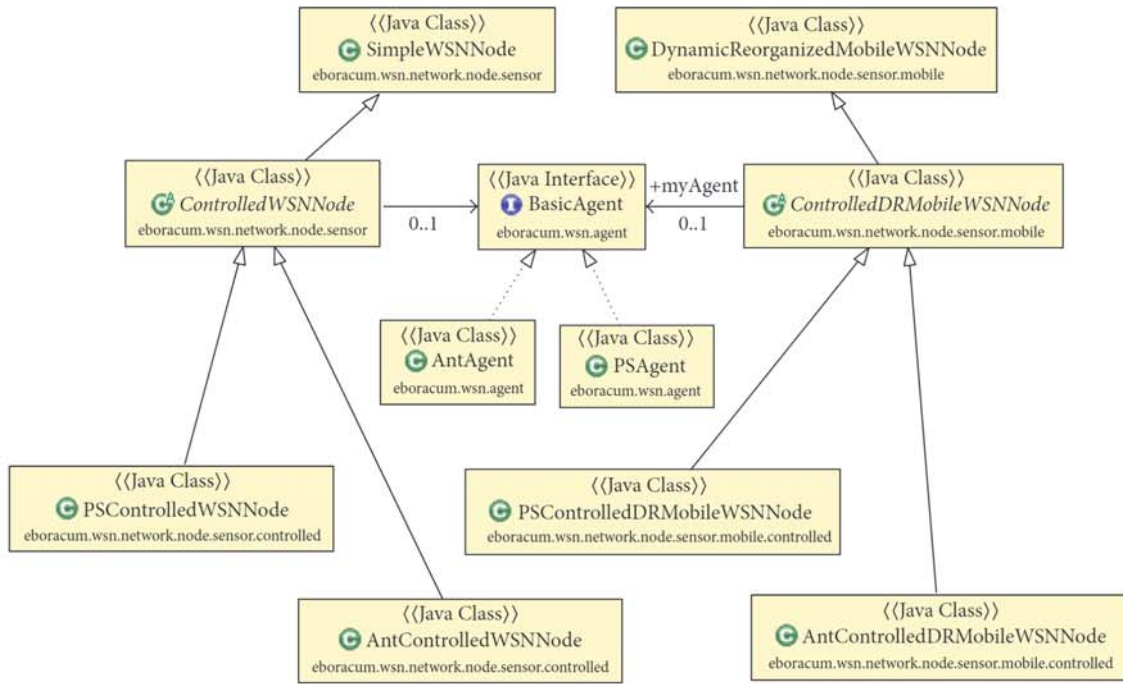


FIGURE 5: Proposed primitives for modeling dynamically controlled nodes.

as *PSAgent* and *AntAgent*. Figure 5 illustrates the new classes and the interface inserted into Eboracum. In previous work [4], experimental results obtained using this implementation are discussed, comparing these techniques regarding network lifetime.

In order to combine mobile nodes to the controlled nodes concept, the *DynamicReorganizedMobileWSNNode* primitive has specialized creating the *ControlledDRMobileWSNNode*. This new primitive represents mobile nodes dynamically reorganized according to the implemented mobility strategy and with behavior controlled by agents. Here, we extend this primitive to combine the proposed mobility to the two bio-inspired load balancing techniques, creating the subclasses named *PSControlledDRMobileWSNNode* and *AntControlledDRMobileWSNNode*. Due to limitations of Java to support inheritance, we are not able to reduce this class hierarchy.

5. Experiments and Results

In our experiments, we consider a simulation scenario composed of a square 810 km² area (900 m × 900 m) and one sink node located on the side. The nodes are connected to one of its neighbors (the closest to the sink) or directly to the sink. Nodes sensing and communication radius are 120 m and 160 m, respectively. We adopt networks with 49, 64, 81, and 100 sensor nodes covering the same area in order to compare different network densities. The adopted energy costs and also the battery capacity are defined in Table 1, according to IRIS Motes data-sheet [22]. We assume that the battery discharge is linear and when the battery is in the half of its charge, the node stops to work. The energy cost associated

TABLE 1: Energy costs.

Energy-related parameter	Value
Battery capacity	5400000 mAs
Idle discharge rate	0.3 mAs
Task computation discharge rate	3.57 mAs
Discharge rate per message (3 bytes) at 30 kbps	0.0018 mAs

with the movements of mobile nodes is out of the scope of this paper and is ignored in our experiments.

StochasticPeriodicJumperEvent class [6, 7] from Eboracum is used to represent abstract events that trigger the network. In our simulation scenarios, events are randomly distributed through the area of interest following a uniform distribution of probability. There are not simultaneous events happening and time between successive events is defined by a Poisson distribution of probability with interval of 120 seconds. Each sensed event generates a processing workload equivalent to 14 tasks and the sent messages have 3 bytes.

In our experiments, the dynamic management strategies are evaluated using different WSN configurations. Besides varying the density (from 49 to 100 nodes), we also vary the deployment strategy from mesh to random. When a mesh is adopted, nodes are distributed in the area of interest according to a grid, staying equidistant from each other. In the random strategy, we randomly deploy sensor nodes using a uniform distribution of probability. For all experiments, the same configuration was executed 30 times and mean results

by day are used in the comparisons (t -tests were run with 0.95 of confidence).

In the first set of experiments, we explore mobile nodes, as a dynamic reorganization strategy, without any other dynamic management strategy. Results achieved by the reorganization are presented in Section 5.1 and are obtained using the primitive *DynamicReorganizedMobileWSNNode*. These results are compared to results achieved by static nodes modeled as *SimpleWSNNodes*.

The load balancing techniques are evaluated in the second set of experiments employing static nodes. To build these simulation models, the primitives named *AntControlledWSNNode* and *PSControlledWSNNode* were adopted. These results are discussed in Section 5.2 and compared to static nodes modeled as *SimpleWSNNodes*, which do not move nor run any load balancing techniques.

Finally, in the third set of experiments, we evaluate the combination of mobile nodes with the two state-of-the-art load balancing techniques, aiming to observe the impact of jointly employing both kind of dynamic management strategies on event-triggered WSN. *PSControlledDRMobileWSNNode* and *AntControlledDRMobileWSNNode* were used to represent the combined version of mobile nodes with each load balancing technique. Results achieved by these versions are compared to ones obtained with mobile nodes without load balancing modeled with *DynamicReorganized-MobileWSNNode* primitive. This discussion is presented in Section 5.3.

For the simplicity, when comparing the usage of simple nodes with mobile nodes we call it “static” and when comparing to the load balancing techniques, we call it “simple.” In both cases we are making reference to the same kind of sensor node represented by the *SimpleWSNNode* primitive.

5.1. Dynamic Reorganization through Mobile Nodes. Firstly, here we evaluate the performance of mobile nodes when they are deployed as a mesh. Our aim is to show that mobiles nodes can reorganize themselves to sense areas that were left uncovered as nodes die. Figure 6 depicts the mean number of events sensed by day considering the different density setups, with static and mobile nodes. Until the day 76, the area of interest is totally covered and the number of sensed events is the same for all configurations. After this moment, nodes die and the number of sensed events starts to decrease and the efficiency of each configuration differentiates from each other. The network using 100 mobile nodes sensed around 45% more events by day, during 7 of its 84 total days, when compared to the network using 100 static nodes.

Figure 7 illustrates the sum of the mean events sensed by day, representing the overall performance of mobile nodes versus static nodes in mesh networks. One can see that, as the networks density increases, the total number of sensed events by mobile nodes also raises compared to the static ones. In the highest density networks with nodes deployed in mesh, around 2% more events are sensed by the mobile sensor nodes against static ones.

Deploying sensor nodes randomly brings no cover or communication guarantees in the network, even following a uniform distribution of probability. As exemplified in

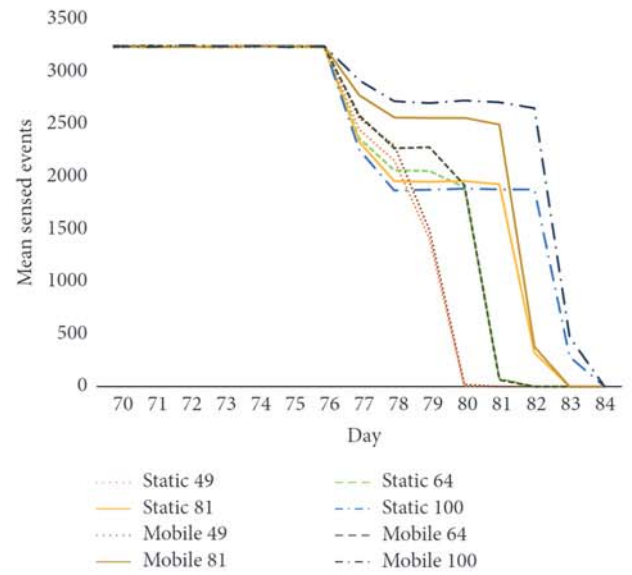


FIGURE 6: Comparing the sensed events by day of mobile versus static sensor nodes in a mesh.

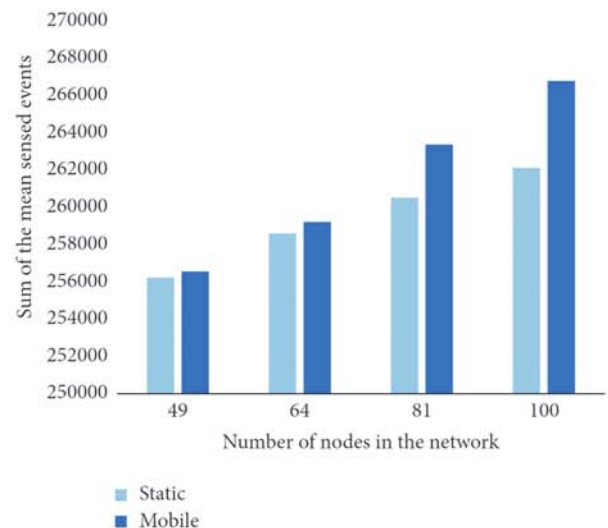


FIGURE 7: Total events sensed by static versus mobile sensor nodes in a mesh.

Figure 4(a), when this deployment strategy is adopted, several nodes have no path to the sink and there are lacks of sensing coverage in the area of interest. Following our reorganization approach, as soon as nodes are deployed, they start to move themselves in order to deal with these issues.

Figures 8 and 9 illustrate the overall performance results achieved by mobile and static nodes in random deployments in terms of mean sensed events by day and the sum of the mean events sensed by day, respectively. Comparing these results, an increasing around 220% in the mean of sensed events is obtained by mobile nodes in the lowest density networks. However, the average quality of the random deployment of static nodes improves as the density increases, as shown in Figures 8 and 9. Thus, the difference between



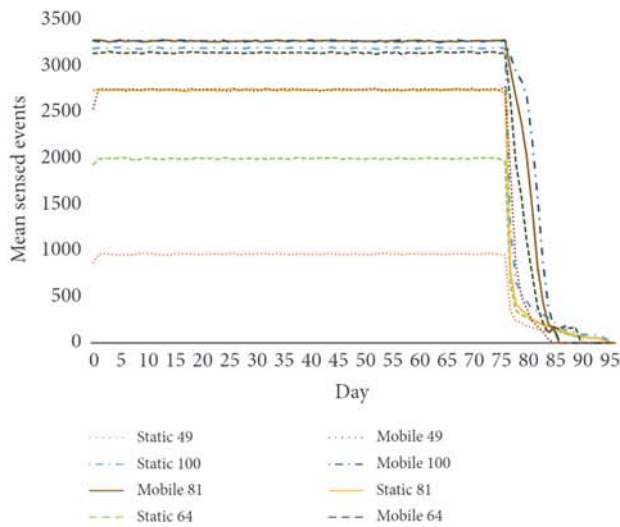


FIGURE 8: Comparing the sensed events by day of mobile versus static sensor nodes in a random deployment.

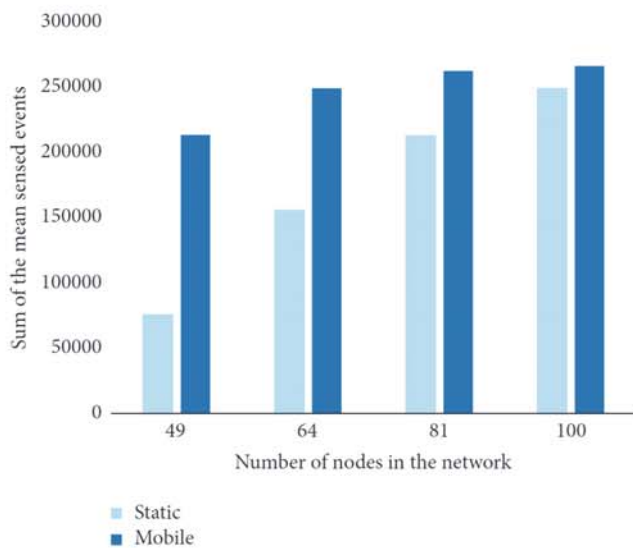


FIGURE 9: Total events sensed by static versus mobile sensor nodes in a random deployment.

static and mobile nodes results decreases to 66% when density is 64 and is almost zero in higher densities.

5.2. Load Balancing. The PS-based and Ant-based load balancing techniques are evaluated here considering mesh and random deployments in WSN composed of static nodes.

Results achieved by the load balancing techniques in mesh deployment are compared to ones achieved by using only simple nodes, which means nodes without any dynamic management strategy, in Figure 10. As previously discussed in [4], the Ant-based technique presents efficiency improvements for all densities compared to simple nodes and only achieves worst results than PS in the highest density configuration (100 nodes). On the other side, PS achieves

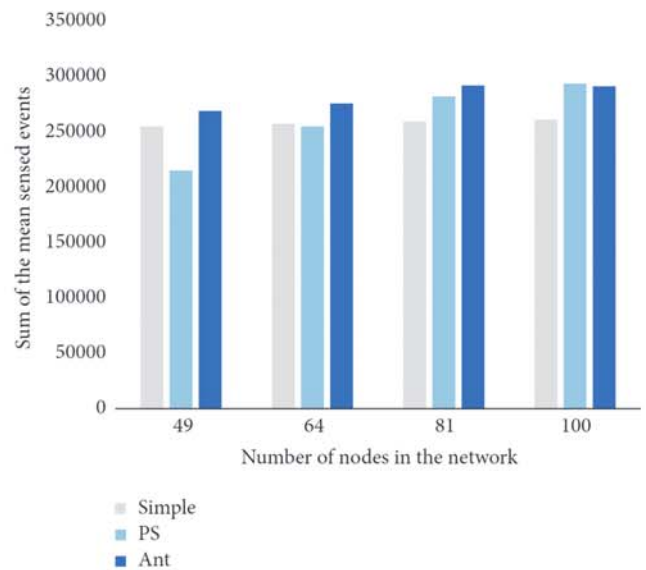


FIGURE 10: Total events sensed by load balancing techniques versus simple sensor nodes in a mesh.

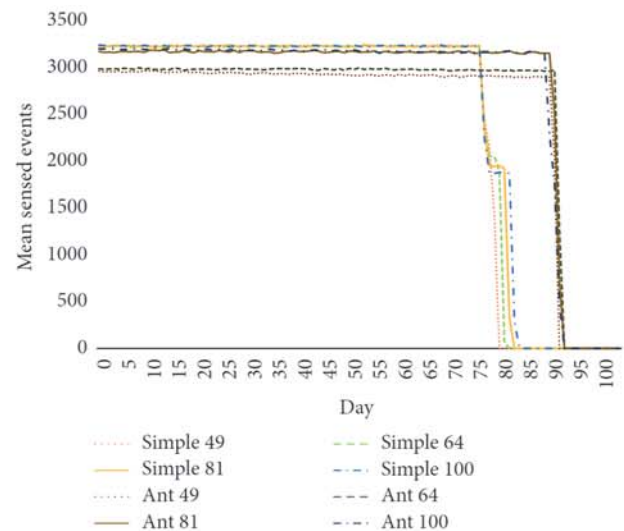


FIGURE 11: Comparing the sensed events by day of load balancing with Ant versus simple sensor nodes in a mesh.

improvements compared to simple nodes for all densities, except in the lowest density, where PS is not efficient.

Figure 11 illustrates the sensed events by day achieved by the Ant-based load balancing technique and by using simple nodes (without load balancing) for the four different densities, considering nodes deployed in mesh. The Ant-based load balancing, as shown in previous work, improves the network lifetime for all densities in around 15 days. In the lowest densities, the load balancing technique reduces the number of sensed events by day. This behavior cannot be seen in the highest densities (81 and 100 nodes), where Ant-based technique achieves the same number of sensed events by day than simple nodes and yet increases the lifetime.



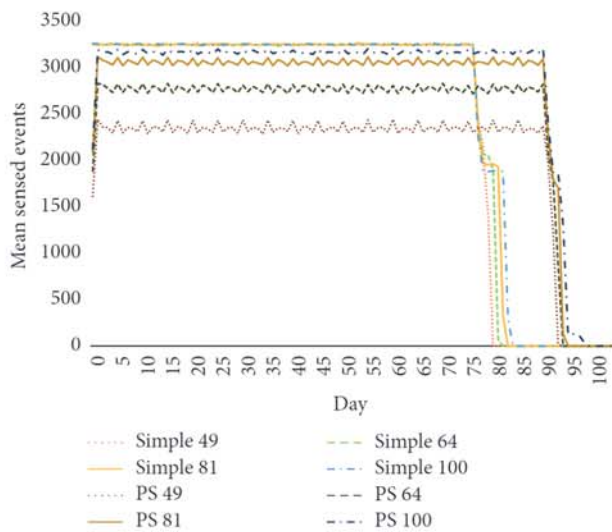


FIGURE 12: Comparing the sensed events by day of load balancing with PS versus simple sensor nodes in a mesh.

Figure 12 illustrates the results achieved by the PS-based load balancing technique and by using simple nodes for the four densities with nodes in mesh. In this set of experiments, PS also increases network lifetime for the four densities in comparison to the networks with simple nodes (without load balancing) but lost a significant amount of events. We conclude that its performance regarding service availability is even worse than a configuration with simple nodes for a network with 49 nodes. This behavior is due to the lack of covering in the area of interest, since only Queen Nodes can sense. As the density increases (81 and 100 nodes), PS achieves better results compared to simple nodes.

When the random deployment is adopted, due to the reduced coverage, the network tends to lose a huge number of events compared to mesh networks. It impacts also the performance of the load balancing techniques, as shown in Figure 13, where the total number of sensed events achieved by both techniques is compared to one achieved by simple nodes, considering randomly deployed networks. One can observe also that the load balancing techniques do not achieve significant improvements on the number of sensed events compared to simple nodes, even in networks with high density.

Results of number of sensed events by day achieved by Ant-based technique are compared to the ones achieved by simple nodes in Figure 14, considering random deployments and varying the network density. Since in random deployment high density networks present good results even with simple nodes, the Ant-based technique does not achieve a significant improvement, as can be observed in Figure 14.

Figure 15 compares results achieved by PS-based technique to ones achieved by simple nodes, considering random deployments and the four network densities. PS presents also some improvements in lifetime, but in contrast it loses a significant amount of events. Thus, as pointed out in Figure 13, PS does not achieve improvements in the sum of mean sensed events in the four densities.

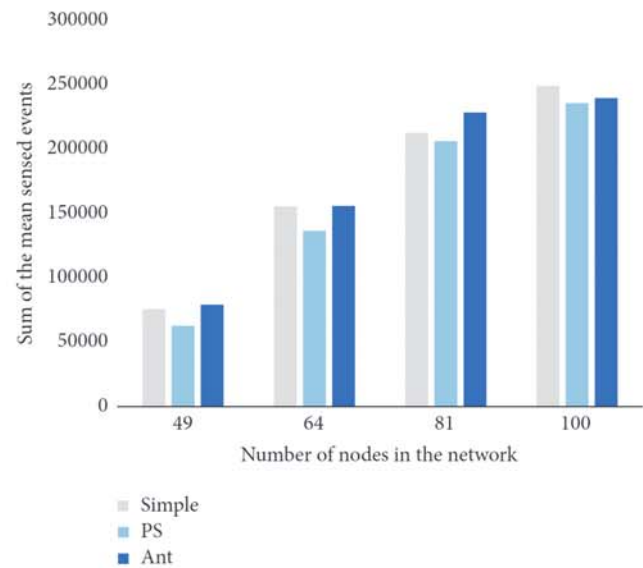


FIGURE 13: Total events sensed by load balancing techniques versus simple sensor nodes in a random deployment.

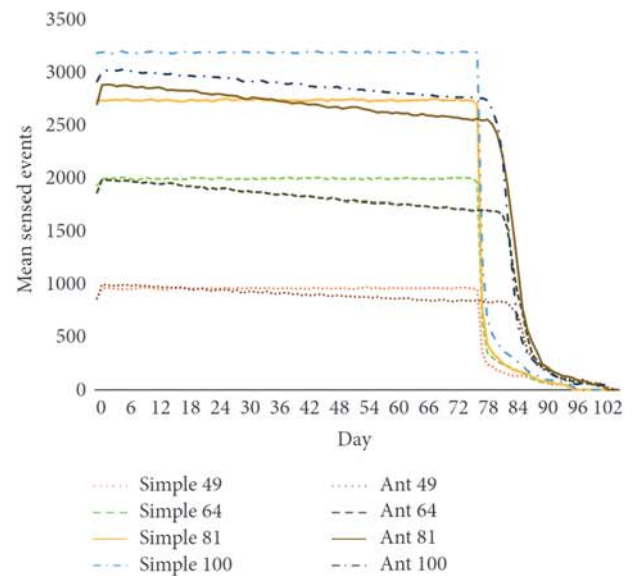


FIGURE 14: Comparing the sensed events by day of load balancing with Ant versus simple sensor nodes in a random deployment.

5.3. Combining Mobile Nodes with Load Balancing. In this section, we evaluate the combining of mobile nodes with the two load balancing techniques and compare their results against the use of simple mobile sensor nodes, varying density and deployment.

In a mesh deployment, Ant-based load balancing technique, as shown in Figure 16, loses performance when combined with mobile nodes in high density networks. This is due the fact that this technique implementation has one important parameter that depends on the number of neighbors and it is affected by the nodes mobility.

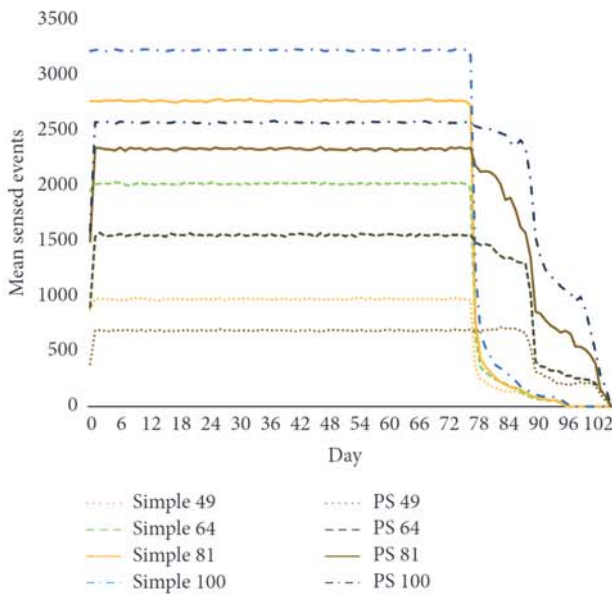


FIGURE 15: Comparing the sensed events by day of load balancing with PS versus simple sensor nodes in a random deployment.

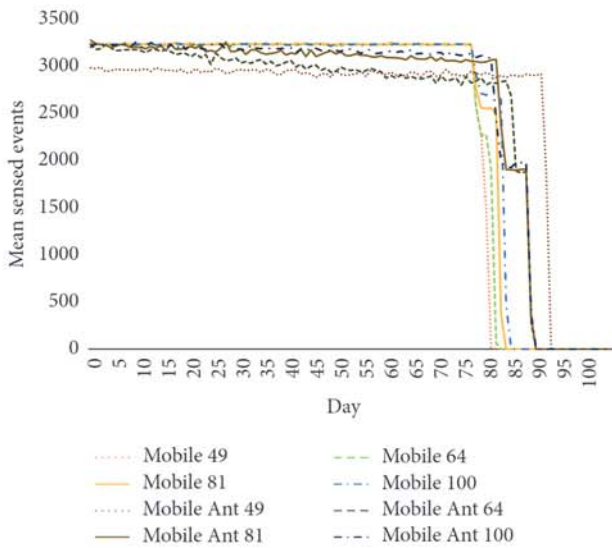


FIGURE 16: Comparing the sensed events by day of Ant-based load balancing in mobile sensor nodes versus simple mobile sensor nodes in a mesh deployment.

As explained in [4], sensor nodes running Ant-based technique use a communication process by which each node can estimate the total number of neighbors that could have sensed the same event at the same time, named $N_j(t)$. This parameter is computed according to the maximum number of messages they simultaneously received until current time t . In other words, in the way this technique was implemented; $N_j(t)$ only increases during the simulation. Thus, when the number of neighbors decays due to the movement of the nodes to reorganize the network coverage, $N_j(t)$ does not follow this decay. After a reorganization where the real number of neighbors decreases, the value of $N_j(t)$ for the

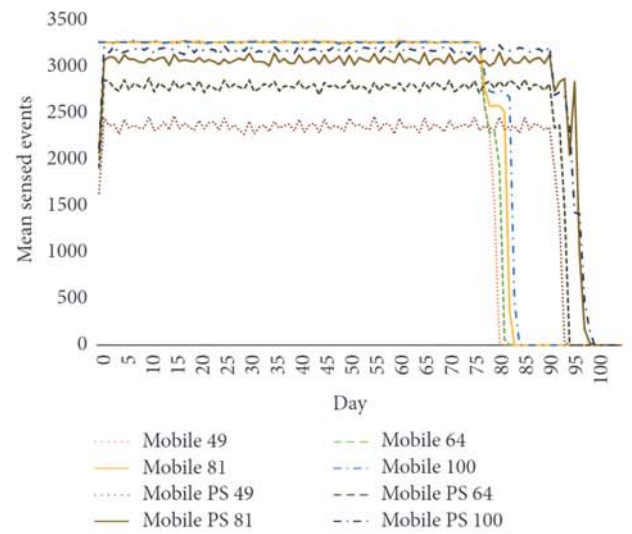


FIGURE 17: Comparing the sensed events by day of load balancing in mobile sensor nodes with PS versus simple mobile sensor nodes in a mesh deployment.

nodes is kept as the maximum total number of neighbors they have ever had. As nodes compute the stimulus to process each task through the equation $s_j(t) = s' + \delta - \alpha N_j^{\text{act}}(t)/N_j(t)$ [4], it will be artificially higher since the initial stimulus s' will be decremented by the number of neighbors that sense the same event divided by an actually wrong total number of neighbors. The loss of performance is given by the artificial higher stimulus that drives more nodes to process the same sensed event.

Figure 17 illustrates the results achieved by PS combined with mobile nodes and the ones achieved by simple mobile nodes in mesh networks. One can observe that PS combined to mobile nodes presents a performance similar to a small improvement to that obtained by PS in mesh networks composed of static nodes. This points out that the small improvement found is due to the mobility.

The overall performance of the load balancing techniques combined with mobile nodes in mesh networks can be seen in Figure 18. These are compared against the results achieved by mobility without load balancing. Figure 19 presents both results of the combination of load balancing with mobile nodes and results achieved by mobile nodes without any load balancing technique in randomly deployed networks.

For both deployment configurations, one can observe that Ant-based technique achieves better performance only in low density networks (49 and 64 nodes), while PS keeps its performance presenting improvements in high density networks (81 and 100 nodes). However, PS improvements in random deployments are smaller than that achieved in mesh for high density networks. PS increases the network lifetime but loses several events daily compared to the simple mobile sensor nodes, as we will discuss next.

Results achieved by Ant-based technique combined with mobile nodes and by the mobile nodes without load balancing, in random networks, are depicted in Figure 20.



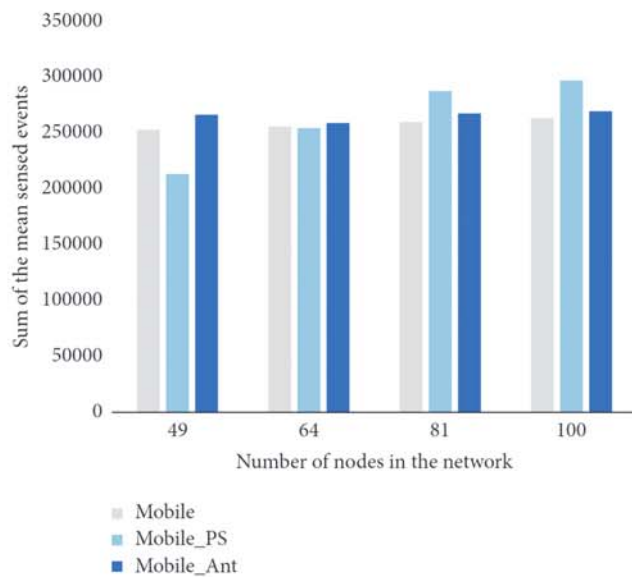


FIGURE 18: Total sensed events by day of load balancing in mobile sensor nodes versus simple mobile sensor nodes in a mesh deployment.

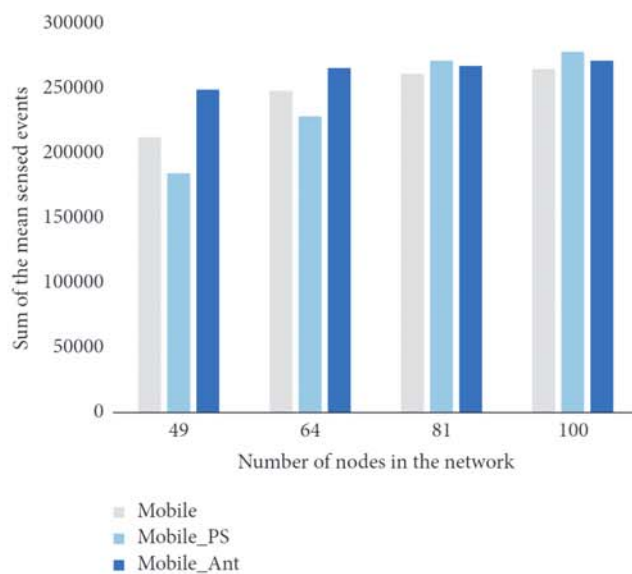


FIGURE 19: Total sensed events by day of load balancing in mobile sensor nodes versus simple mobile sensor nodes in random deployment.

As expected, Ant-based technique combined with mobile nodes achieves better results compared to the usage of Ant-based technique with static nodes in random deployments. Furthermore, because of the random deployment the drawback we mentioned before about the number of neighbors is minimized. Ant-based technique still is the best for low densities but achieves almost the same results compared to the simple mobile nodes for the higher dense ones.

Figure 21 depicts the results achieved by PS technique combined with mobile nodes and by the mobile nodes without load balancing, in random networks. PS technique

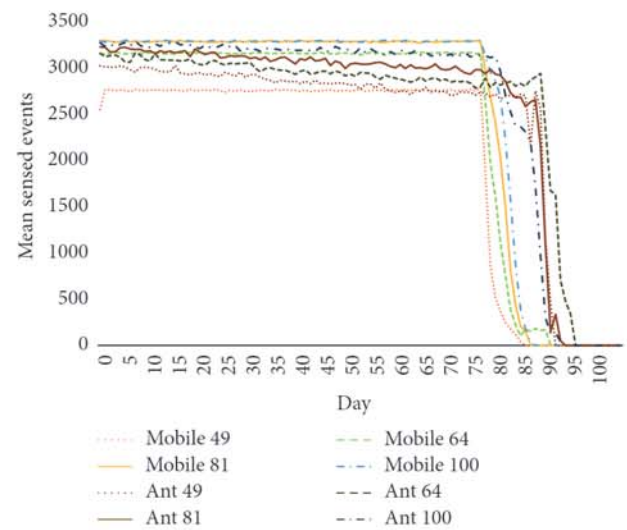


FIGURE 20: Comparing the sensed events by day of load balancing in mobile sensor nodes with Ant versus simple mobile sensor nodes in a random deployment.

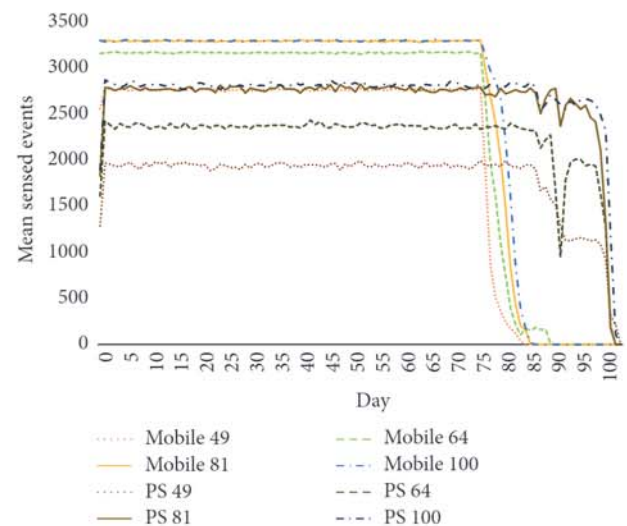


FIGURE 21: Comparing the sensed events by day of load balancing in mobile sensor nodes with PS versus simple mobile sensor nodes in a random deployment.

presents a consistent behavior, since it keeps its bad performance in low densities and better performance in high density, what the Ant-based technique cannot keep. It is important to highlight that PS combined with mobile nodes starts to be a good strategy from networks with 81 nodes, as shown in Figures 19 and 18, while PS with static nodes is the best solution only in networks with 100 nodes (see Figures 13 and 10).

6. Conclusion

In this work we have presented and discussed different dynamic management strategies applied to improve



efficiency in event-triggered WSN. The strategies have been implemented using Eboracum, a high-level modeling and simulation framework. The first strategy to be evaluated is related to the nodes mobility aiming to increase sensing coverage. A simple method has been proposed for nodes reorganization based on the force fields approach of mobile robotics, which can be easily applied to mobile sensor networks running on a high-level simulation. Moreover, different dynamic load balancing strategies have been discussed and evaluated.

Extensive experimental work has been conducted to discuss the impact of these strategies on WSN efficiency, considering different WSN setups. Firstly, experiments have evaluated each strategy separately and afterwards the combined strategies have also been evaluated. The combination allows observing the impact of mobility on the efficiency of load balancing techniques on event-triggered WSN.

These experiments show that, in random networks configurations, mobile nodes impact, as expected, increases the coverage in low density networks in around twice more sensed events by day. This advantage starts to decrease as the density becomes higher. Furthermore, randomly deployed mobile nodes always achieve results as good as the static ones deployed in a mesh. It is important to highlight also that mobile nodes present a significant improvement compared to static nodes in mesh configuration.

Our experiments have also evaluated the state-of-the-art load balancing techniques. When nodes are deployed in mesh, load balancing techniques achieve important results, increasing lifetime and the number of sensed events. However, in random deployments, these algorithms lose efficiency and become irrelevant strategies.

Moreover, this paper has evaluated the combination of mobile nodes with state-of-the-art load balancing techniques. Results point out that only PS keeps its performance in comparison to the other set of experiments, once the Ant-based technique is more dependent on number of neighbors impacted by the movement. PS combined with mobile nodes presents a meaningful performance in mesh and random deployed networks with 81 nodes, which has not been seen in previous experiments. Ant-based technique combined with mobile nodes achieves good results only for low density networks, named 49 and 64 sensor nodes.

The experiments have shown a drawback in Ant-based load balancing implementation which cannot be observed without considering the mobility. As future work, we plan to improve Ant-based algorithm to adapt itself to deal with the reorganization of nodes. Furthermore, it will be interesting to work on an energy consumption model for mobile nodes in order to allow considering the battery discharging when moving nodes.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] D. Saha and N. Das, "Self-organized area coverage in wireless sensor networks by limited node mobility," *Innovations in Systems and Software Engineering*, vol. 12, no. 3, pp. 227–238, 2016.
- [2] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, "Coverage maximization in mobile wireless sensor networks utilizing immune node deployment algorithm," in *Proceedings of the IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE '14)*, pp. 1–6, IEEE, Toronto, Canada, May 2014.
- [3] Y. Qu and S. V. Georgakopoulos, "A distributed self-relocating algorithm for randomly deployed mobile wireless sensors," in *Proceedings of the IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet '13)—2013 7th IEEE Radio and Wireless Week (RWW '13)*, pp. 148–150, Austin, Tex, USA, January 2013.
- [4] P. R. Ferreira, L. Brisolara, and L. S. Indrusiak, "Decentralised load balancing in event-triggered wsns based on ant colony work division," in *Proceedings of the 41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA '15)*, pp. 69–75, Madeira, Portugal, August 2015.
- [5] I. Caliskanelli, J. Harbin, L. S. Indrusiak, P. Mitchell, F. Polack, and D. Chesmore, "Bioinspired load balancing in large-scale WSNs using pheromone signalling," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 172012, 14 pages, 2013.
- [6] L. Brisolara, P. R. Ferreira, and L. S. Indrusiak, "Impact of temporal and spatial application modeling on event-triggered wireless sensor network evaluation," in *Proceedings of the Brazilian Symposium on Computing Systems Engineering (SBESC '15)*, pp. 30–35, Foz do Iguaçu, Brazil, November 2015.
- [7] L. Brisolara, P. R. Ferreira, and L. S. Indrusiak, "Application modeling for performance evaluation on event-triggered wireless sensor networks," *Design Automation for Embedded Systems*, vol. 20, no. 4, pp. 269–287, 2016.
- [8] Y. Jin, D. Wei, A. Gluhak, and K. Moessner, "Latency and energy-consumption optimized task allocation in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, pp. 1–6, IEEE, Sydney, Australia, April 2010.
- [9] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: a survey," *ACM Transactions on Sensor Networks*, vol. 8, no. 1, article 7, 2011.
- [10] P. B. Val, M. G. Valls, and M. B. Cunado, "A simple data-muling protocol," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 895–902, 2014.
- [11] A. Wichmann and T. Korkmaz, "Smooth path construction and adjustment for multiple mobile sinks in wireless sensor networks," *Computer Communications*, vol. 72, pp. 93–106, 2015.
- [12] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems 5*, chapter 8, pp. 299–308, Springer, Berlin, Germany, 2002.
- [13] R. Kacimi, R. Dhaou, and A.-L. Beylot, "Load balancing techniques for lifetime maximizing in wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2172–2186, 2013.



- [14] U. Dohare, D. K. Lobiyal, and S. Kumar, "Energy balanced model for lifetime maximization in randomly distributed wireless sensor networks," *Wireless Personal Communications*, vol. 78, no. 1, pp. 407–428, 2014.
- [15] O. Iova, F. Theoleyre, and T. Noel, "Using multiparent routing in RPL to increase the stability and the lifetime of the network," *Ad Hoc Networks*, vol. 29, article 1196, pp. 45–62, 2015.
- [16] M. Navarro and Y. Liang, "Efficient and balanced routing in energyconstrained wireless sensor networks for data collection," in *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks (EWSN '16)*, pp. 101–113, Junction Publishing, Graz, Austria, 2016.
- [17] A. Pathak and V. K. Prasanna, "Energy-efficient task mapping for data-driven sensor network macroprogramming," *IEEE Transactions on Computers*, vol. 59, no. 7, pp. 955–968, 2010.
- [18] Z. Zeng, A. Liu, D. Li, and J. Long, "A highly efficient DAG task scheduling algorithm for wireless sensor networks," in *Proceedings of the 9th International Conference for Young Computer Scientists (ICYCS '08)*, pp. 570–575, Hunan, China, November 2008.
- [19] J. Senthilkumar, M. Chandrasekaran, Y. Suresh, S. Arumugam, and V. Mohanraj, "Advertisement timeout driven bee's mating approach to maintain fair energy level in sensor networks," *Applied Soft Computing Journal*, vol. 12, no. 7, pp. 1884–1890, 2012.
- [20] J. R. Andrews and N. Hogan, "Impedance control as a framework for implementing obstacle avoidance in a manipulator," in *Control of Manufacturing Processes and Robotic Systems*, pp. 243–251, 1983.
- [21] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 500–505, St Louis, Miss, USA, March 1985.
- [22] "M. Inc. Iris: Wireless measurement system," <http://www.mem-sic.com/userfiles/files/Datasheets/WSN/IRIS.Datasheet.pdf>.

