Prolonging network lifetime under probabilistic target coverage in wireless mobile sensor networks

Dimitrios Zorbas, Tahiry Razafindralambo

Prolonging network lifetime under probabilistic target coverage in wireless mobile sensor networks

Dimitrios Zorbas, Tahiry Razafindralambo

Inria Lille - Nord Europe
France

Abstract

One of the main operations in wireless sensor networks is the surveillance of a set of events (targets) that occur in the field. In practice, a node monitors an event accurately when it is located closer to it, while the opposite happens when the node is moving away from the target. This detection accuracy can be represented by a probabilistic distribution. Since the network nodes are usually randomly deployed, some of the events are monitored by a few nodes and others by many nodes. In applications where there is a need of a full coverage and of a minimum allowed detection accuracy, a single node may not be able to sufficiently cover an event by itself. In this case, two or more nodes are needed to collaborate and to cover a single target. Moreover, all the nodes must be connected with a base station that collects the monitoring data. In this paper we describe the problem of *the minimum sampling quality*, where an event must be sufficiently detected by the maximum possible amount of time. Since the probability of detecting a single target using randomly deployed static nodes is quite low, we present a lo-

Email addresses: `Dimitrios.Zormpas@inria.fr` (Dimitrios Zorbas), `Tahiry.Razafindralambo@inria.fr` (Tahiry Razafindralambo)

calized algorithm based on mobile nodes. Our algorithm sacrifices a part of the energy of the nodes by moving them to a new location in order to satisfy the desired detection accuracy. It divides the monitoring process in rounds to extend the network lifetime, while it ensures connectivity with the base station. Furthermore, since the network lifetime is strongly related to the number of rounds, we propose two redeployment schemes that enhance the performance of our approach by balancing the number of sensors between densely covered areas and areas that are poorly covered. Finally, our evaluation results show an over 10 times improvement on the network lifetime compared to the case where the sensors are static. Our approaches, also, outperform a virtual forces algorithm when connectivity with the base station is required. The redeployment schemes present a good balance between network lifetime and convergence time.
Keywords: Wireless sensor networks, Quality of sampling, Probabilistic coverage, Node redeployment, Target coverage, Energy efficiency

1. Introduction

Wireless sensor networks have attracted a lot of attention recently due to the plenty of the applications and their connections with the physical world [1]. One of these applications is the surveillance of a set of static events that occur in the field [2]. These events are represented by a point in the network and are commonly named “targets”. In order to monitor an event each node is equipped with one or more sensing modules. Moreover, the nodes can communicate with each other and exchange information constructing an ad-hoc network. The monitoring of the targets produces data that the sensors forward to a usually wired and power unlimited machine, called the "base station" (BS). The tiny dimensions of the sensors as well as their limited battery resources lead to various algorithmic challenges in every kind of operation including coverage and communication [3].

The coverage is an important operation and it is usually assumed that it can be achieved when a target lies in the sensing range of a node (binary detection model). It practically means that the event will be covered with probability one, either the target is located very close to the sensor, or it lies on the borders of the sensing range. In this paper, we assume a more realistic coverage model where the signal propagation that derives from a source (target) and is detected by a sensor, follows a probabilistic distribution. This assumption makes our system desirable for many kind of sensor nodes like radio, acoustic and seismic, where the signal strength decays with the distance [4]. According to the probabilistic distribution and a log-distance path loss model [5] we can predict the path loss that a signal encounters over distance. Additionally, this means that the signal weakens with the distance and it can be detected with high probability closer to the source.

The nodes are randomly deployed in the field. Hence, some events are covered by few sensors while others are covered by many sensors. Moreover, some nodes are located closer to a target while others may cover the same target by a longer distance. The nodes that cannot cover a target, because they are far away and their coverage probability is too low to be taken into consideration, are used as relay sensors in order to connect the sensing nodes with the BS. We, also, assume that it can be moved inside the target coverage region in order to increase the detection probability. A node may be sensing node and relay node at the same time when it is close to events and it is used by other nodes in order to forward their monitoring data to the base station. A relay node can operate only as a router in the network and it cannot change position.

Considering a probabilistic coverage model and a randomly deployed network with mobile nodes, we can define the minimum detection quality problem. The objective of the problem is to prolong the network lifetime as much as possible and to achieve a minimum allowed detection probability for all the targets in the network. Additionally, the network should retain connectivity with the BS throughout the monitoring operation. To solve the above problem we present Localized Coverage Quality ALgorithm (*LoCQAl*), an algorithm that divides the monitoring operation in rounds. In each round only a set of sensors is active while the rest of the nodes remain in sleep mode, and conserve energy. LoCQAl moves some nodes, when it is possible, in order to achieve the desired detection accuracy. Since some sensing nodes are used for relaying, they cannot be moved to achieve the minimum allowed probability, so other nodes assist with the monitoring of a target. Hence, our algorithm aims at minimizing the number of active nodes as well as the node's traveling distance within the target's coverage region.

Since the nodes consume the same amount of energy to monitor an event, an event that is covered by few sensors can be monitored by shorter time than an event that is covered by many sensors. As the network lifetime depends on the coverage of all the events, it is upper bounded by the amount of energy of the sensors covering the most sparsely covered target [6]. For this purpose we present two schemes that redeploy the nodes by moving nodes from dense populated target areas to sparse populated target areas. These schemes differ on the timing that the redeployment takes place. The first approach redeploys the nodes at the beginning of the process, exhibiting sometimes a long delay, while the second approach moves the sensors during the monitoring process, reducing the delay in the network.

The contributions of this paper are: a) we introduce the minimum detection quality problem for a set of targets in the network, b) we present LoCQAl, a localized algorithm to solve the previous problem and to ensure connectivity, and c) we enhance the performance of LoCQAl using two approaches that redeploy the nodes in the network, such that all the targets will be covered by almost the same number of sensors.

The rest of paper is organized as follows. In Section 2, we make a discussion about the previous works related to our problem. In Section 3, we analyze the minimum detection quality problem along with the parameters and the assumptions of the model, while in Section 4 we describe the motivation of our work. In Section 5 we present our solution and the limitations of the approach. To tackle these limitations we present, in Section 6, the two sensor redeployment schemes. In Section 7, we simulate our approaches and we compare them with a common technique found in the literature that considers the nodes non mobile as well as with an approach that uses virtual forces to move the nodes. Section 8 discusses how our algorithm could be used to solve similar problems such as the coverage and redeployment in presence of mobile targets and obstacles. We, also, present a possible extension of our work in order to move relay nodes towards targets. Finally, Section 9 concludes the paper and presents ideas for future work.

2. Related work

Among the huge number of works in the research field of energy efficiency in wireless sensor networks, we only refer to works that are related to at least one of the properties of our approach, such as the probabilistic coverage, the network lifetime extension under the target coverage constraint, and the node redeployment.

Many works deal with the problem of how well an event is monitored. They consider the probabilistic detection model and use a given number of nodes. These works differ from our approach either because the objectives are slightly different (e.g. the maximization of the coverage degree) [7, 8, 9], or because they consider only static nodes [10]. In [4] a comparison between the probabilistic and the binary detection model is done. The authors conclude through simulation that the area coverage is more accurate using the probabilistic model than the idealistic binary detection model. Moreover, in [7] a coverage protocol is presented that targets to maximize the coverage degree of a given area by activating the minimum number of nodes. In [8] the objective is to find the minimum number of nodes needed in order to provide full coverage under a deterministic setup for any arbitrary shape region. The authors provide, also, formulas to describe the probability of coverage of a desired area by k sensors. The objective in [10] is to deploy the minimum number of sensors so that the distribution of the sensors meets the probability of detection requirements. The authors solve an optimization problem to determine the position of the nodes. Finally, in [9] a number of approaches for the probabilistic area coverage problem is presented. These approaches divide the available sensors in active and sleep mode nodes, while they try to minimize the number of active nodes. The simulation results show that the proposed solutions outperform the approach of [7].

The most common approach to achieve network lifetime extension is to divide the available sensors in sets, such that only one set is active at any time [11, 6]. By successively activating the sets, the network lifetime can be extended. The connectivity with the BS can be achieved by constructing a minimum shortest path tree [6] or a minimum spanning tree [11]. All the works that deal with the target coverage problem consider the binary detection model, so the produced sets may not fulfill, in practice, the desired sampling quality. To tackle this problem, each target can be covered by k sensors [12, 13], however, this approach increases the number of active nodes.

Since the nodes consume the same amount of energy to monitor an event, an event that is covered by few sensors can be monitored by shorter time than an event that is covered by many sensors. As the network lifetime depends on the coverage of all the events, it is upper bounded by the amount of energy of the sensors covering the most sparsely covered target [6]. The only way to achieve full coverage and at the same time to prolong further the network lifetime is to apply a redeployment strategy. In [14], nodes are moved from targets that are covered by many sensors towards targets that are more sparsely covered. This approach sacrifices an amount of energy for movements, but significantly increases the number of nodes covering sparsely covered targets. Since the node redeployment can be considered as a pickup and delivery problem [15], it is important to minimize the total traveling distance of the nodes, as well as the delay of the approach. The works that are related with the network redeployment are based on the use of virtual forces or voronoi diagrams [16, 17, 18], where each node pushes its neighbors towards more sparsely covered areas. However, these algorithms cannot be ideally used in the target coverage problem, since they do not guarantee connectivity and same coverage needs for all the targets.

3. Problem description

In this section we formulate the minimum detection accuracy problem (MDA problem) and we describe the assumptions and the general parameters of the model.

Given a set of targets $T_0 = \{t_1, t_2, ..., t_k\}$, k sets of nodes $N_1, N_2, ..., N_k$, where all the nodes of N_i cover t_i ($\forall\ i \in [1, k]$), and a minimum quality of detection P_{min}, the objective is to prolong network lifetime as much as possible, achieving a probability of detection more than or equal to P_{min} for all the targets in T_0 throughout the monitoring process. The nodes that are used for sensing must be BS-connected using relay and/or sensing nodes found in the field. The network prolonging can be achieved by dividing the sensors in sets, such that only one set of sensors is active at any time. Assuming that each set lasts for τ_δ period of time, the MDA problem can be described as follows:

$\max \sum_{\delta=1}^{\mu} \tau_\delta$, subject to $P_{t_i} \geq P_{min}\ \forall\ t_i \in T_0,$ and $\delta \in [1, \mu]$, where μ is the number of sets (rounds).

All the nodes have an initial amount of energy l_0 and they are assumed mobile. They can spend a part of this energy to move towards a target and achieve P_{min} when it is needed. The amount of energy that is spent for movements is based on the traveling distance and it is assumed constant per distance unit.

P_{min} represents the minimum quality of detection that must be achieved in order to consider a target sufficiently covered. This is a user-given value and depends on the nature of the problem and on the requirements of the monitoring application. In fact, P_{min} is the minimum allowed probability that one or more sensors could yield in order to cover a single target. The probability of detecting an event decreases exponentially as the sensor is moving away from the source. The path loss of the signal at distance d in dB is given by the following formula, according to the log-distance path loss model [5]:

$$PL_d = P_{tx} - P_{rx_d} = PL_0 + 10 \cdot \alpha \cdot \log \frac{d}{d_0} + X_\sigma, \tag{1}$$

where P_{tx} is the power of the source, P_{rx_d} is the received power at distance d, PL_0 is the path loss at the reference distance d_0, α is the path loss exponent, and X_σ is a Gaussian random value with zero mean. PL_0, d_0, α and X_σ can be measured experimentally [4]. For a given distance d and using (1), one could compute the received power at this distance.

The probability $P_{s_j}^d$ describes if a received signal of sensor s_j at distance d will be above a threshold value γ and it is given by the following formula [4]:

$$P_{s_j}^d = 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{\frac{-x^2}{2}} dx, \tag{2}$$

where z is given by $\dfrac{\gamma - P_{rx_d}}{\sigma}$, when $P_{rx_d} > \gamma$. The threshold γ is the minimum power of the signal that can be correctly received by the decoder electronics [4]. We define a maximum distance d_{max} that a node can be found away from a target in order to cover it. Beyond this distance the probability of detection is too low and it cannot be taken into account. We assume that d_{max} is two times less than the communication range of the sensors.

Two or more sensors can collaborate to achieve P_{min} for a single target t_i. The overall probability is given by:

$$P_{t_i} = 1 - \prod_{j=1}^{\nu}(1 - P_{s_j}), \tag{3}$$

where P_{s_j} is the probability of sensor s_j and can be computed using (2), while ν is the number of sensors covering the particular target.

4. Motivation

Considering non-mobile nodes and a uniform node distribution, the probability of detecting an event by a single node with probability P_{min}, deploying n sensors in the field is given by:

$$P_n = 1 - (1 - P_1)^n, \tag{4}$$

where n is the number of deployed nodes and P_1 is the probability of covering a target deploying one sensor. P_1 is given by:

$$P_1 = \frac{\pi d^2}{\text{terrain area}}, \tag{5}$$

where d depends on P_{min}'s value. As it is shown in Figure 1, in the case where a target has to be covered by one sensor, deploying 150 nodes, P_n is quite high even when the terrain size is large (sparse network) or P_{min} is high. However, if we want to extend the network lifetime, each particular target area should be occupied by many sensors. The probability of covering a target, for example, by 3 sensors with probability at least P_{min} is illustrated in Figure 2. As it is shown P_n remains very low even when P_{min} is small. Given a uniform distribution, the solution to this problem is either to deploy a huge number of nodes in the field (> 2000), or to use mobile nodes in order to move them towards targets that are not efficiently covered.

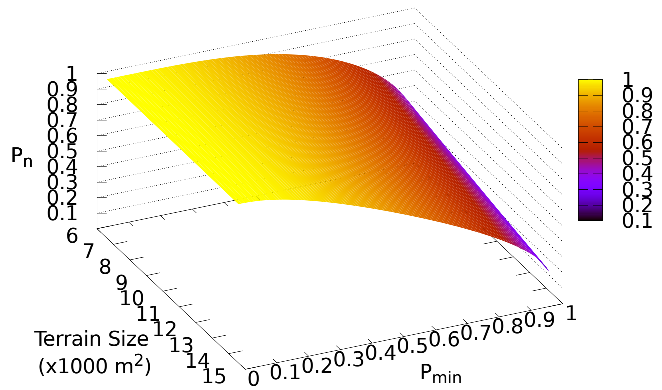

Figure 1: Probability of detecting an event with probability P_{min} by one sensor

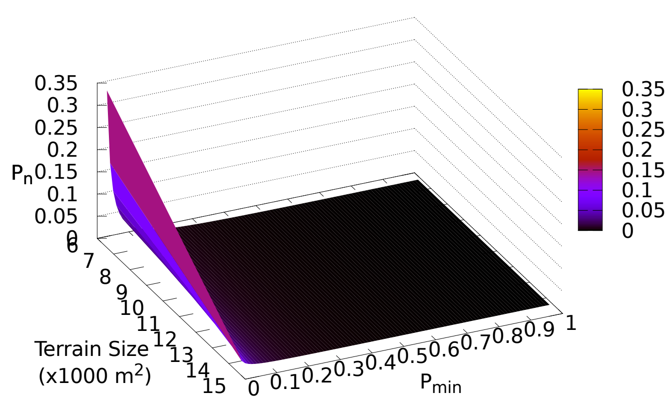

Figure 2: Probability of detecting an event by 3 sensors with probability P_{min} each

The connectivity is an important issue of the MDA problem. The sensing nodes must report their measurements to the BS using a multi-hop approach since they may not be able to directly communicate with it. This approach requires the presence of one or more relay nodes that route the monitoring data to the BS. Since the sensors are assumed mobile, they can move closer to a target and achieve P_{min}. However, the node's movement could lead to a loss of connectivity and, thus, to a loss of information. In addition, the nodes that are part of the backbone network that forwards the information to the BS cannot be moved. In this case, at least two nodes should be used to achieve P_{min}. The finding of the number of active nodes and the position of these nodes are two open issues in MDA problem.

5. The "LoCQAl" solution

In this section we present **LoCQAl** (i.e. LOcalized Coverage Quality ALgorithm), a localized algorithm that depends on the neighboring information in order to solve the MDA problem. LoCQAl uses the ability of the sensors to move in order to achieve the minimum allowed detection probability and to tackle the detection issue described in the previous section.

According to LoCQAl, each node knows its own position, the position of the targets, and the minimum allowed probability P_{min}. The position of a sensor could be also detected using the strength of the coverage signal and localization techniques [19].

LoCQAl considers the network as a connected undirected graph $G = (V, E)$, where V consists of sensing nodes, relay nodes and the BS. Every two vertices u, v that belong to V and they can directly communicate with each other, establish an edge uv in G and $uv \in E$. A node can cover a target following the probabilistic coverage model of [4] that has been described in the previous section.

LoCQAl works in rounds. Each round incorporates two main phases; the connectivity phase and the coverage phase. LoCQAl ensures the BS-connectivity by computing a connected dominated set (CDS). Both sensing and relay nodes take part in the computation of the CDS. Each node decides if it will be dominant or not using the localized algorithm of [20]. The formulated CDS operates as a backbone network which the sensing nodes can use to reach the BS. After the CDS computation all the nodes in the network are BS-connected. Depending on the number of targets and the network topology, none, one or more sensing nodes may be dominant in each round of LoCQAl.

According to the position of the sensors, the probability P_{min} and the computation of the CDS, LoCQAl distinguishes three categories of sensing nodes for the coverage phase:

1. The first category includes nodes that are parts of the CDS and cannot move. If such a node cannot achieve P_{min} by itself, a second node is needed to assist the dominant node. The latter node checks if one of the neighboring nodes that covers the particular target could assist without being moved. If the cooperative probability is still lower than P_{min}, the closer to the target node is moved towards it to increase the probability of detection and satisfy P_{min}. Note, that more than two nodes may need to achieve P_{min} in the case that none of the neighboring nodes can be moved. The traveling distance of the assistant node depends on its own

probability and on the probability of the dominant node. This distance is given by:

$$d = 10^{\frac{z \cdot \sigma - \gamma + P_{tx} - PL_0 - X_\sigma}{10 \cdot \alpha}}, z = \mathrm{indf}(1 - \frac{P - P_{min}}{P - 1}), \quad (6)$$

where `indf` is the inverse of the normal distribution function of (2) and P is the probability of the dominant node(s).

2. The second category includes nodes that cover multiple targets. This kind of nodes sometimes cannot be moved since a possible movement could lead to uncovered targets. A node of this category looks for assistance when it cannot achieve P_{min} by itself for at least one of the targets it covers. The assistant(s) selection process is the same with the previous case. In the special case where the rest of the targets can be covered by other sensors, the node that covers multiple targets can be safely moved to a new position. If a node is dominant and covers multiple targets at the same time, it cannot be moved.

3. Finally, the third category includes nodes that are not part of the CDS and they do not cover multiple targets. These nodes can be moved closer to a target when they are not able to achieve P_{min} from their initial location. The new position is computed using (6), where P, in this case, is the probability of the moving node at its initial position.

LoCQAl prioritizes the selection of nodes of the first category. When a sensor declares itself dominant after the connectivity phase, it means that it will be already active during the coverage phase. Thus, it is preferable to use its coverage contribution in order to decrease the traveling distance of a possible assistant node. In order to high prioritize nodes of the first cate-

Algorithm 1: Decision process of a sensing node s in LoCQAl

```
require: nbrs_s, trgs_s, CDS
if dominant_s = 1 then
    Category1(s);
else
    if |trgs| > 1 then
        Category2(s);
    else
        Category3(s);
```

gory and cover a particular target, each node communicates with its 1-hop neighbors. Nodes that do not cover any target or nodes that cover different targets are ignored. When a node of the other two categories receives a reply message from a dominant node, it remains inactive, leaving the dominant node to select which node will assist with the monitoring of the target. If there is no dominant neighboring node, LoCQAl prioritizes the selection of nodes of the second category. If there are no nodes of the second category either, a node from the third category is selected to cover the target.

The above operation for a single sensor is described in Algorithm 1. The decision if a sensor will be active or not during the next round, is taken by communicating with its 1-hop sensing neighbors and by building set $nbrs_s$. This set contains the nodes that cover the same targets that s covers. s, also, keeps in $trgs_s$ the targets that it covers, while it executes one of the three procedures in order to decide its status.

Theorem 1. *Each round of LoCQAl produces BS-connected sensor sets.*

Proof. If a node keeps its initial position during the coverage phase, it is sure that it will still communicate with at least one node of the CDS. Con-

Procedure Category1

```
foreach t ∈ trgs_s do
    active_s = 1;
    foreach n ∈ nbrs_s do
        if active_n = 1 then
            active_s = 0;
    if active_s = 1 then
        if P_s < P_min then
            select a n ∈ nbrs_s with the minimum probability, such that P_{s,n} ≥ P_min;
            active_n = 1;
            if none selected then
                select a n ∈ nbrs_s with the highest probability;
            if none selected then
                send a process termination message;
            else
                compute selected n's new location according to Formula (6);
                active_n = 1;
```

cerning the moving nodes and according to the node classification described before, a node can change position due to following three reasons:

1. It is assistant of a dominant node. The maximum traveling distance of a node is equal to d_{max}. Because d_{max} is at least two times lower than the communication range, and the dominant node is located inside the target coverage region, the moving node will be still connected to the dominant node at its new position.
2. It is assistant of a node covering multiple targets that cannot be moved. If the node that covers multiple targets cannot be moved, it will still retain connectivity with at least one dominant node. Since the maximum traveling distance is d_{max} and d_{max} is at least two times lower

Procedure Category2

```
active_s = 1;
i = 0;
foreach n ∈ nbrs_s do
    if active_n = 1 or dominant_n = 1 then
        i = i + 1;
if i = |trgs_s| then
    active_s = 0;
if active_s = 1 then
    need_cover = ∅;
    foreach t ∈ trgs_s do
        if P_s < P_min then
            put t in need_cover if none of the nodes covering t can be moved;
    if |need_cover| > 1 then
        send a process termination message;
    else if |need_cover| = 1 then
        compute new location according to Formula (6);
    else if |need_cover| = 0 then
        foreach t ∈ trgs_s do
            if P_s < P_min then
                select a n ∈ nbrs_s with the minimum probability, such that P_{s,n} ≥ P_min;
                active_n = 1;
                if none selected then
                    select a n ∈ nbrs_s with the highest probability;
                if none selected then
                    send a process termination message;
                else
                    compute selected n's new location according to Formula (6);
                    active_n = 1;
```

than the communication range, the moving node will still be connected to a dominant node or to the node that covers multiple targets.

Procedure Category3

```
active_s = 1;
foreach n ∈ nbrs_s do
    if active_n = 1 or dominant_n = 1 or |trgs_n| > 1 then
        active_s = 0;
if active_s = 1 then
    if P_s < P_min then
        compute new location according to Formula (6);
```

3. It is a node of the third category. It is not assistant, but it moves to achieve P_{min}. At its initial position this node can directly communicate with at least one dominant node. Moving to a new position it may lose connectivity with these nodes. In this case, the node will look for a neighbor that is connected with a dominant node. The only case that there will be no neighbor, is when the moving node is the only node that still covers the particular target. In this case, the node will not be BS-connected, but the process will stop anyway, since the minimum coverage detection is not satisfied.

Since each moving node knows its destination and the energy consumption per meter, it can be moved if it has enough remaining energy. Since the nodes are moved inside a small circular area with range d_{max}, the maximum energy consumption is $e_m d_{max}$, where e_m is the energy consumption per traveling distance unit. Using a common electrical motor [21] and a d_{max} value equal to $10m$ this energy consumption is many times less than the energy provided by two AA batteries [22].

In LoCQAl each set will last until one node depletes its battery. For simplicity reasons, we assume that all the active nodes consume the same amount of energy per time unit. However, LoCQAl could work fine, taking into account other energy consumption models; for example an energy consumption model based on the distance between the nodes and the produced date rate. Following this simple model, the node that has traveled the longest distance, will die first. If this node is part of the CDS, the CDS has to be recomputed at the beginning of the next round. If the exhausted node is not part of the CDS, the backbone network will still consist of the same nodes during the next round, reducing the overhead that comes from the CDS computation. LoCQAl terminates when at least one target cannot be sufficiently covered or when the CDS computation fails.

LoCQAl exhibits a low communication complexity, since it mainly based on the complexity of the CDS. According to the algorithm of [20] each node must have a two-hop knowledge of its neighbors or 1-hop neighbors with their geographic positions in order to decide if it will be dominant or not. Regarding the coverage phase, each node that covers a target t_i sends and receives $|N_i| - 1$ messages to decide if it will be active or not in this round, where $|N_i|$ is the number of sensors covering target t_i. When an assistant node is selected, at least $|N_i| - 1$ more messages are needed to be sent by the sensor that cannot be moved.

Despite the fact that LoCQAl minimizes the number of active nodes, and thus it conserves energy, it still depends on the number of nodes covering the most sparsely covered target in the network. For example, what could be done in the case where a target is covered by a single node which can't achieve P_{min} and is part of the CDS as well? In this case LoCQAl will fail if there is no mechanism to provide the poorly covered area with sensors from neighboring densely covered target areas.

6. Increasing the number of rounds

As explained in Section 1, since the maximum achieved network lifetime highly depends on the number of sensors covering the most sparsely covered target in the network [6], some nodes could be moved from areas that are densely covered balancing the number of sensing nodes. For this purpose we present two localized sensor redeployment approaches.

The first approach is based on *Local-TCR* redeployment algorithm [14]. According to this algorithm sensors are moved from a target area to a neighboring target area if the number of sensors covering the two areas differs by at least two nodes. This redeployment scheme takes place before the operation of LoCQAl in order to balance the number of sensing nodes among the targets and to increase the number of nodes covering the most poorly covered targets. However, Local-TCR does not guarantee that the network will be still BS-connected after the redeployment process. For this reason we have modified Local-TCR in order to fit to our problem's requirements. In more detail, we allow the algorithm to run only after the first CDS computation. The nodes that are parts of the CDS cannot be moved during the coverage phase in order to maintain connectivity. The rest of the sensing nodes can be safely moved to a neighboring target area.

The redeployment operation works in rounds and in each round a sensor that covers the most densely covered target in its 2-hop neighborhood checks if, how, where, and how many sensors will be moved. At the beginning of the operation each node covering a target discovers the neighboring nodes that cover the same targets. Using this information, each node builds a set of elements for each target, such as how many nodes cover each target (i.e. target cardinality), the position of the neighboring nodes and their available energy. Since all the nodes have built these sets of elements, they exchange these sets with their 1-hop neighbors in order to compute the sharing factor. The sharing factor is based on the average cardinalities of the neighboring targets. If this average is less than the cardinalities of its own covered targets, a node declares itself as a moving node candidate. The higher the difference between the average of the neighboring target cardinalities and the cardinalities of the targets a node covers, the higher the sharing factor of the node. Each node shares its factor with its 2-hop neighbors, and it compares it with their own. The node with the highest sharing factor in its 2-hop neighborhood will be a “head-node” for this round. A head-node decides who, where, and how many nodes will be moved from the richly covered target to the poorly covered target. More than one head-node may be selected in the same round, but only one is selected for each particular 2-hop neighborhood. In order to avoid having two head-nodes in the same neighborhood, nodes multiply their sharing factor with a random value in $(1, 1.01]$ before they send it to the other candidates.

The number of moving nodes depends on the sharing factor of the head-nodes. Since the head-nodes know the positions of the neighboring nodes and the positions of the targets, they select the nodes that are closer to their neighboring target with the lowest cardinality. The selected nodes are moved towards those targets and they stop at distance d_{max} from it. The nodes are not moved further in order to cover the target with probability P_{min}, leaving this decision to LoCQAl during the next round. Actually, this action may not be necessary, since P_{min} may be achieved by this position when other active nodes cover the same target at the same time.

The redeployment process terminates when there are no other possible movements and all the targets are covered by about the same number of sensors. It has been proven that the previously described approach always balances the number of monitoring nodes between two neighboring targets when the number of nodes covering the two targets differs by two or more sensors [14].

Figure 3 illustrates a network with four targets (squares) covered by 5, 7, 2 and 4 nodes respectively (dots). A node covering target B will be the only head-node (all the sensing nodes belong to the same 2-hop neighborhood) and it takes the decision of how many sensors will be moved towards target C (neighboring target with the lowest cardinality). Since this number of nodes has been decided, nodes covering target B are moving towards C. In the example of Figure 3 two nodes have been moved. The two nodes have stopped on the borders of the coverage area of target C (big circle) and are indicated with a small circle in Figure 4.

In some scenarios, a number of targets may be far away from the other targets. It means that some nodes covering these targets may not be able to reach a node that covers different targets and to exchange the target cardinality sets. In this case, the redeployment fails, since only the targets with neighboring nodes will be balanced. In such a scenario, the CDS can be used to forward the messages to the closest neighboring sensing nodes.

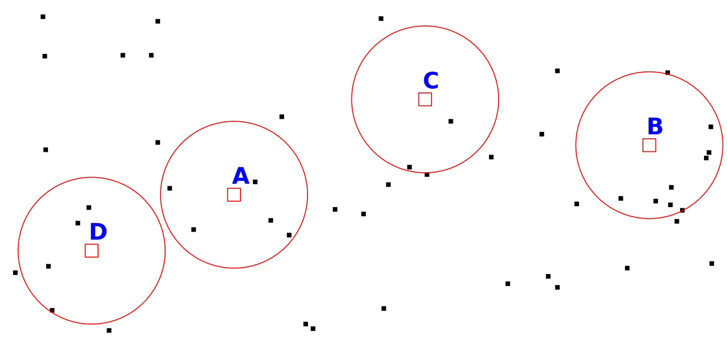

Figure 3: An unbalanced network with 4 targets

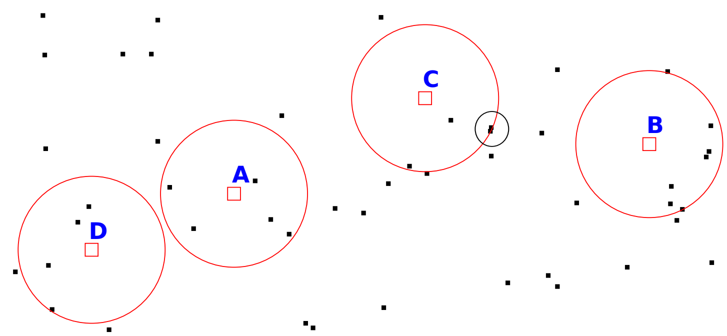

Figure 4: A balanced network with 4 targets

The nodes can use a parameter called "effective redeployment range" (R_{eff}) that describes how many hops away a message will be disseminated. If R_{eff} is one, no CDS node will be used. Since the nodes know the positions of the targets and the position of their 1-hop neighbors, they can ideally compute their R_{eff} value.

The traveling distance of a node and its energy consumption are based on the effective redeployment range. The higher the R_{eff} value, the longer distance the node may travel in order to move towards another target area. The longest distance that a node could travel is $R_{eff}R_c + 2d_{max}$, where R_c is the communication range of the node. Figure 5 depicts the energy consumption of a moving node for different R_{eff} values, $R_c = 50m$ and $e_m = 100J/m$. Depending on the battery capacity of the node, we can specify the maximum value of R_{eff} that could be used in order to be benefited from the redeployment process. For example, if l_0 is equal to 20K $Joules$ (energy capacity of two AA batteries [22]), an effective range value up to 3 can be used. This means that our redeployment approach is capable of balancing a network lying on square terrain of 25K m^2. For larger terrain sizes, we should use sensor nodes with higher energy storage.

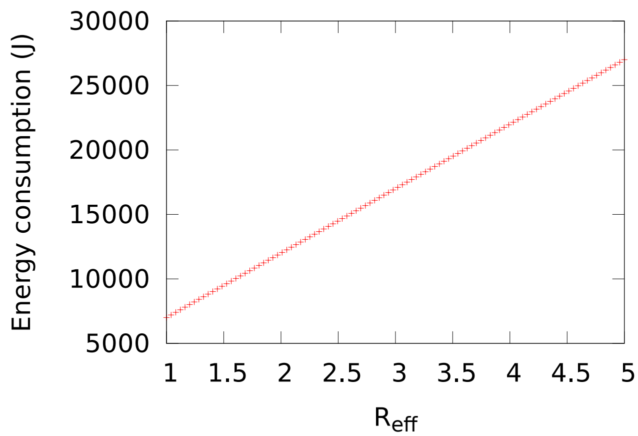

Figure 5: Energy consumption related to the effective redeployment range

LoCQAl begins after the termination of the redeployment process. At this point, all the nodes are BS-connected, however, the nodes may have different remaining energy levels, since some of them may have already spent a part of their available energy in order to move to a neighboring target.

Since a node cannot be instantly moved to a new location, the overall approach exhibits a long delay at the beginning of its operation. This delay depends on the number of moving sensors and the distance between the targets. For time constraint applications, where this long delay is not acceptable, we also present an algorithm that moves nodes only during the monitoring phase. This approach works similar to the previous one, but only one redeployment round takes place per monitoring round, allowing in fact two operations to take place at the same time; the monitoring and the redeployment operation.

The two redeployment approaches in combination with LoCQAl are illustrated in Figure 6. This figure shows the operation of two monitoring rounds. Each round consists of the negotiation phase and the monitoring phase. During the negotiation phase all the nodes are active and decide which nodes will be active during the monitoring phase. The monitoring phase may last from a few minutes to a few hours depending on the available battery lifetime. In the first approach, the redeployment takes place at the beginning of the process and before the first monitoring round. In the second approach, during the negotiation phase, apart from which sensors will be active during the monitoring phase, the nodes decide which nodes will be moved to a new location as well. The redeployment of the nodes begins when the monitoring phase has just started. However, the nodes' movements may last longer than the monitoring phase. In this case, the non-moving nodes will wait until the moving nodes reach their destination and a third monitoring round begins. This, apparently, causes a delay on the monitoring operation, although shorter than the delay of the first approach.

The second approach reduces the redeployment delay, however it is not expected to achieve always the same performance in terms of network lifetime compared to the first approach. This may happen due to the following two

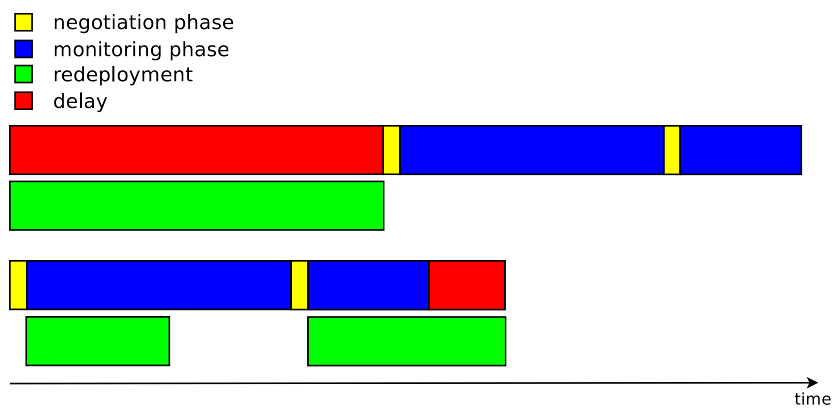

Figure 6: The two redeployment schemes combined with LoCQAl

reasons: a) some nodes are not allowed to be moved because they are marked active in the current round (they will provide coverage). This means that either the redeployment will be postponed or some other nodes will be selected, traveling although a longer distance. b) Local-TCR moves sensors only between neighboring target areas. In the case where the densely populated area is many hops away from the most sparse area, a few rounds are required to balance the number of nodes. In those few rounds, the nodes covering the most sparse target may have been already exhausted their batteries.

7. Evaluation & discussion of the results

In this section, we simulate LoCQAl and compare its performance to an algorithm that considers the nodes non-mobile. This algorithm uses multiple nodes in order to achieve P_{min}, when it is not possible to cover a target by a single node. As explained in Section 2, this approach is named κ-coverage and it is commonly used in the literature [12]. In our case, κ may differ from target to target depending on the position of the nodes and the value of P_{min}. Moreover, we evaluate the performance of the two algorithms that use LoCQAl in conjunction with the two redeployment schemes described in Section 6, against algorithm of [17]. We have slightly modified this algorithm in order to fit to our problem requirements. We make a discussion about those modifications in the next section. Our redeployment algorithms are named LoCQAl-InstR and LoCQAl-GradR respectively. LoCQAl-InstR uses the first redeployment scheme, where the redeployment takes place at the beginning of the operation, and LoCQAl-GradR uses the second scheme which reduces the delay by allowing sensors to move during the coverage phase.

7.1. Evaluation environment

We assess the algorithms in 2 different scenarios using 150 nodes with random target and sensor deployments. In the first scenario we keep a constant terrain size equal to 10K m^2 and we vary the number of targets (10 - 30), while in the second scenario we vary the terrain size (6.5K - 14K m^2), keeping a fixed number of targets (15 targets). The detection probability varies for both scenarios. For each scenario and for each probability we run 50 instances and we compute the average network lifetime of these 50 instances. The 95% confidence intervals are shown in the 2-dimensional figures. The communication range of the nodes is $50m$ and d_{max} is equal to $10m$. Concerning the battery capacity, each node initially has $20KJoules$ available and it spends $100Joules/m$ when it is moved and $0.1Joules/sec$ for sensing and communication. The speed of the sensors is $1m/sec$ [1]. The parameters we used for the shadowing model [5] are: $P_{tx} = 24.5dBm, \gamma = -27.85dBm, \alpha = 2, \sigma = 4dBm, PL_0 = -50dBm, d_0 = 1000m$, and $X_\sigma = 147.476dBm$. These values are similar to those used in [4]. We, also, assumed that all the sensing nodes produce data with the same rate.

We developed our own simulator that produces terrain scripts with uniform random node and target deployments[2]. Our software is, also, capable of producing 2-dimensional square terrain images for each one of the monitoring rounds. Figure 7 illustrates an image of a scenario with 150 nodes and 10 targets. The small squares below the letters denote the targets, the big surrounding circles denote the maximum distance that a node can be found away from a target in order to cover it, while the small surrounding circles denote the maximum distance that a node can be found away from a target in order to cover it with probability at least P_{min}. In this example, P_{min} is equal to 0.9. The active nodes are marked in bold, while the CDS nodes are drawn inside a rectangle. The rest of the nodes are assumed to be in sleep mode. The BS is positioned in the middle of the left side of the terrain (filled square).

Concerning Garetto et al. algorithm [17] we implemented it by slightly changing the definition of the virtual forces. The algorithm consists of three

---

[1] This is the maximum energy consumption recorded and has been measured experimentally in our lab using Wifibots robots using the maximum speed [21]

[2] The simulation software as well as the proposed algorithms have been implemented in Perl programming language and they can be found at http://rainbow.cs.unipi.gr/projects/sensors under the GNU General Public License

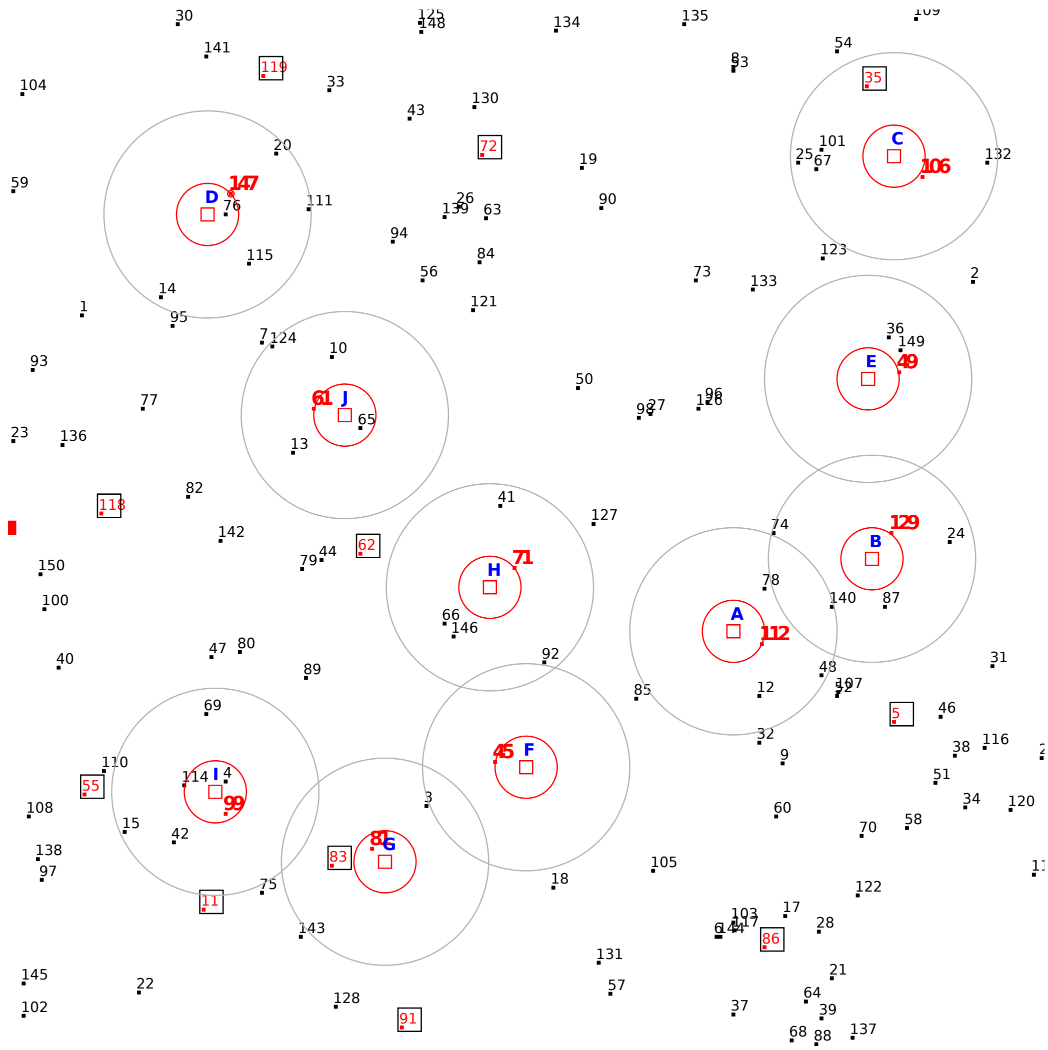

Figure 7: A WSN with 10 targets (small squares below letters), active sensing nodes (bold numbers), CDS nodes (numbers inside a rectangle) and sleep-mode nodes (rest of numbers)

phases. During the first phase (self-deployment) the nodes form a regular triangular tessellation. In the second phase (coverage), the nodes that are close to an event start to move towards it pulling the rest of the nodes as well. We skipped the third phase during which the nodes reform the regular triangular tessellation when an event has disappeared.

We also assumed the following strategy in order to fit this algorithm to our needs. Since the approach does not guarantee connectivity with the BS, which is a requirement for our problem, we allowed the nodes to spread in all over the terrain during the self-deployment phase. We computed a CDS network to ensure connectivity when redeployment has finished. We, also, assumed that a sensor can move towards a target during phase 2 if it is in distance less than or equal to d_{max} from it, while it stops moving when it has achieved the desired detection probability. However, it can still attract or push other neighboring sensors. A characteristic of the algorithm is that it does not provide equally covered targets. This means that when the nodes are moving towards the events during phase 2, they are not split equally causing some targets to be covered by many sensors and others by few sensors. This is reasonable since the density of the sensors at the end of phase 1 is more or less uniform, but the distribution of the targets is not consistent.

Concerning the simulation parameters, we used the following values: $D_m = 45$, $k_0 = 0.01$, $k_v = 0.1$, $P = 10$, $G = 30$, while G increases when a sensor is closer to a target. The maximum speed of a nodes is $1m/sec$ and the redeployment takes $450secs$. The nodes compute the virtual forces every second. The name of the algorithm on the plots is “Virtual Forces”.

7.2. Evaluation results

In the first set of measurements, we assess the impact of the target density and the minimum coverage probability on the network lifetime. The number of targets affects the performance of LoCQAl, since when many targets are deployed, more sensing nodes are becoming part of the CDS. When a few targets are used (Figure 8 left), LoCQAl presents a small decrease on the network lifetime as the detection probability increases. Looking at the scenario with many targets, we can observe that the performance of LoCQAl decreases fast. This happens because when the detection probability is low, the sensors that cannot be moved and they can monitor the targets without the assistance of a second node. As the probability increases, an increased need of assistance is required and, thus, the network lifetime is reduced. However, LoCQAl increases dramatically the network lifetime compared to the non-mobile approach, since the latter approach activates many sensors, depleting rapidly the energy of the available nodes. Figure 9 presents a depiction of the same round for the two compared approaches. On the right figure, Non-mobile activates too many sensors, while LoCQAl, on the left, keeps active only one node per target. The minimum detection probability of the example is 0.5.

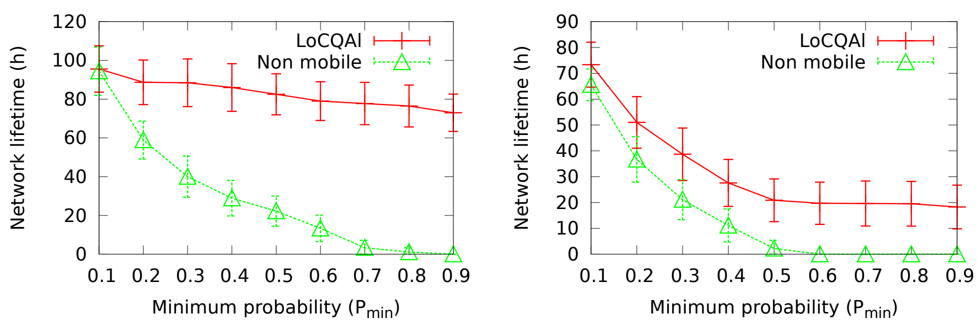

Figure 8: Network lifetime produced for a scenario with 10 targets (left) and 30 targets (right)

Figure 10 illustrates the performance of the algorithms in two different

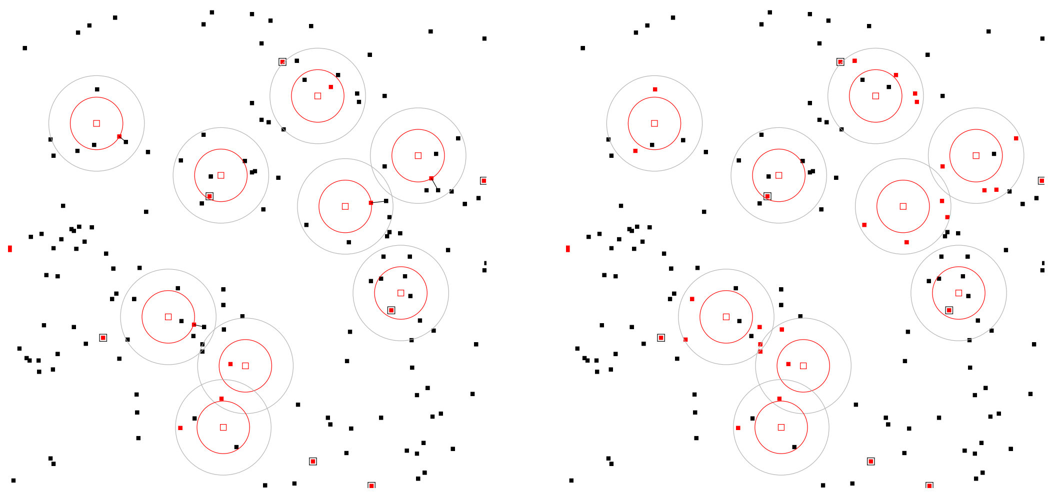

Figure 9: A depiction of the same round of LoCQAl and Non-mobile for a scenario with $P_{min} = 0.5$, 150 sensors, 10 targets and 10K m^2 terrain size

terrain sizes. Varying the terrain size, we can estimate the impact of the network density on the algorithms' performance. As it was expected, in the dense node deployment, the targets are covered by many sensors and, thus, the network remains alive for a long time, especially when the minimum allowed coverage probability is low. When this probability is low, the two algorithms are very close, but LoCQAl produces over 10 times better result when P_{min} is getting high. In the sparse deployment, the gap is larger since the targets are covered by fewer sensors. At this point, the simulation results validate the theoretical results presented in Section 4 (see Figure 2). Figure 11 depicts the percentage of the targets that have been covered by at least one or three nodes with probability P_{min}. The results show that for a scenario with 150 sensors and 15 targets and different terrain sizes, only a small percentage of the targets can be covered with high probability. This percentage dramatically decreases when the targets are required to be covered by 3 sensors (Figure 11 right) making impossible for Non-mobile to find enough sensing nodes that sufficiently cover each particular target.

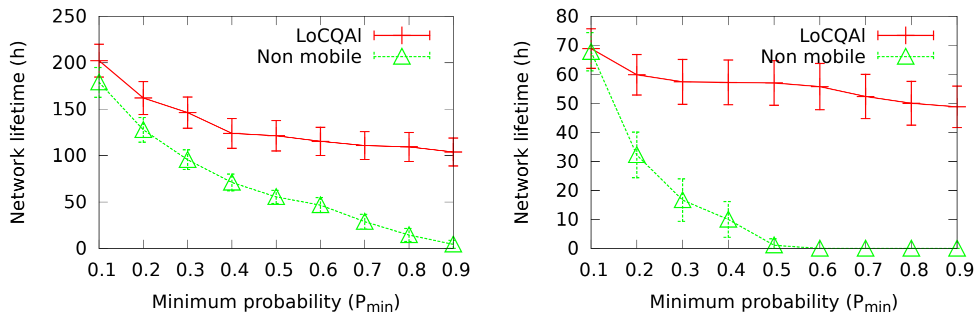

Figure 10: Network lifetime produced for a scenario with high node density (left) and low node density (right)

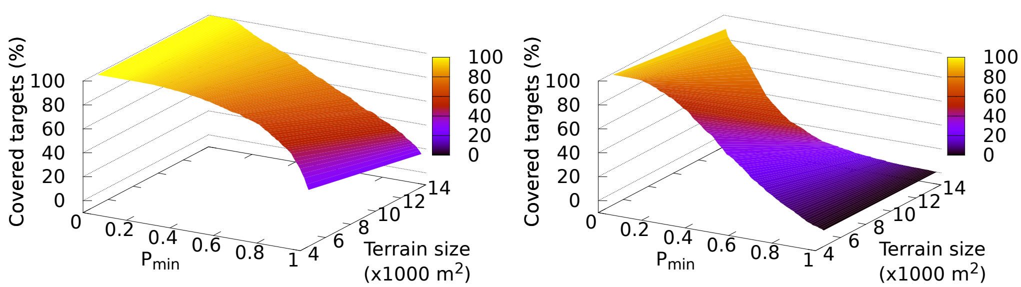

Figure 11: Targets (%) that have been covered by at least one node (left) or three nodes (right) with probability P_{min}

Figure 13 illustrates the number of CDS computations needed throughout the monitoring process for the previous scenario. Since the number of computations is related to the produced network lifetime, we present results with the relative values. LoCQAl presents a constant trend on the number of CDS computations per monitoring hour, while Non-mobile presents similar results, except of the cases where P_{min} is high. In those cases the produced network lifetime is negligible and, thus, the number of CDS computations per hour is extremely high.

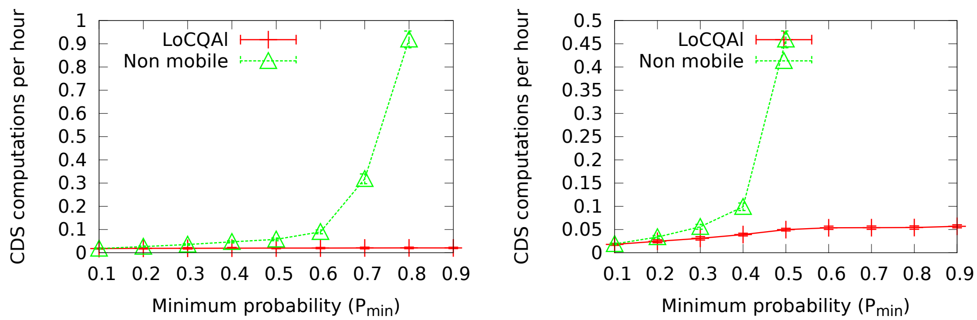

Figure 12: CDS computations per hour for a scenario with 10 targets (left) and 30 targets (right)

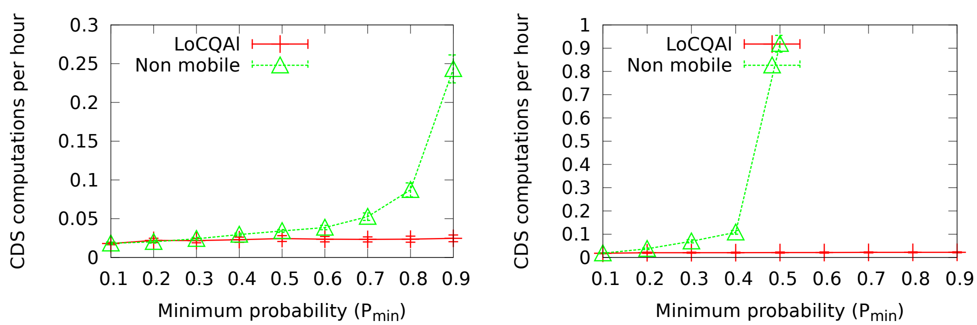

Figure 13: CDS computations per hour for a scenario with high node density (left) and low node density (right)

Taking into consideration the redeployment schemes described in Section 6 the network lifetime can be improved. This can be justified by the simulation results presented in Figures 14 and 19. LoCQAl-InstR presents the best results, while LoCQAl-GradR is close. The algorithm that uses virtual forces increases the average number of sensors covering the targets, but some targets are still covered by few sensors limiting the network lifetime. When the number of targets increases, many sensing nodes are used as CDS nodes and cannot be moved, increasing the number of active nodes in the network.

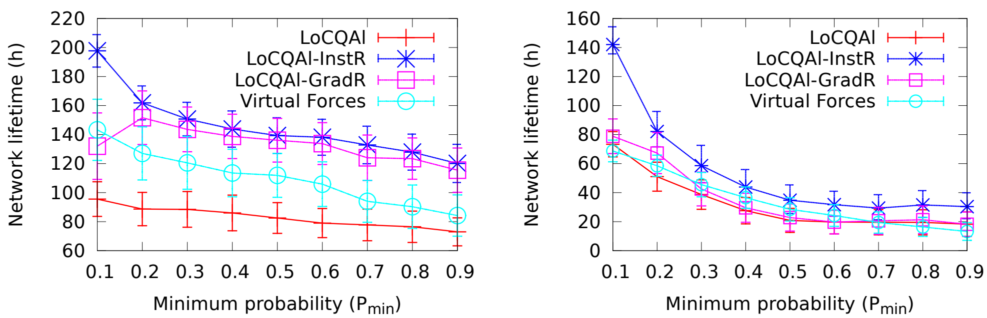

Figure 14: Network lifetime produced for a scenario with 10 targets (left) and 30 targets (right)

For the same scenario, we measured the number of CDS computations and the number of exchange messages per node. The normalized values instead of the absolute values are presented in Figures 15 and 16. The results show that Virtual Forces computes the CDS more often than the other algorithms. It happens due to the fact that the relay nodes have spent part of their energy in order to move to another location during the self-deployment phase. That means that these nodes will die faster causing consecutive executions of the CDS process. Concerning the number of exchange messages of the approaches, it is clear that the algorithm with the Virtual Forces produces more overhead in the network, since each node communicates with its neighbors in order to advertise its position. The plotted values include the overhead caused by redeployment operation as well as by the CDS computations. Figures 17 and 18 illustrate the absolute number of messages in detail. LoCQAl-InstR produces low number of messages when a few targets are covered, while almost the half of the messages are produced during the CDS process. When many targets are covered, the number of messages increases up to 300%, and it looks to be proportional to the number of targets. On the other hand, Virtual Forces requires a high number of messages in both cases. Even if we assume that the nodes exchange their position every 5 secs instead of 1 sec, the overhead will be still more than that of LoCQAl-InstR.

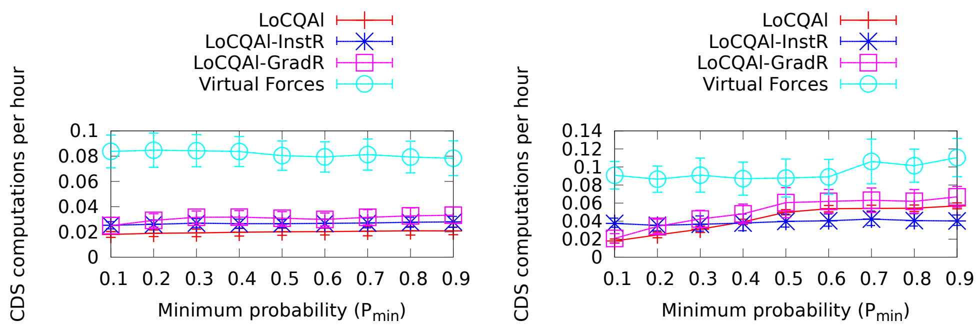

Figure 15: Number of CDS computations per hour for a scenario with 10 targets (left) and 30 targets (right)

In the case where the network density changes, LoCQAl-GradR presents comparable or sometimes better performance than LoCQAl-InstR at least when the density is high (Figure 19 left). This happens because the redeployment is achieved in a few only rounds, since all the targets are not far away from each other and nodes can be moved directly to the most sparsely covered area. The opposite holds true, when the node density is low. In this case, the redeployment takes more rounds and the possibility of depleting the energy of the nodes that already cover the most sparsely covered target

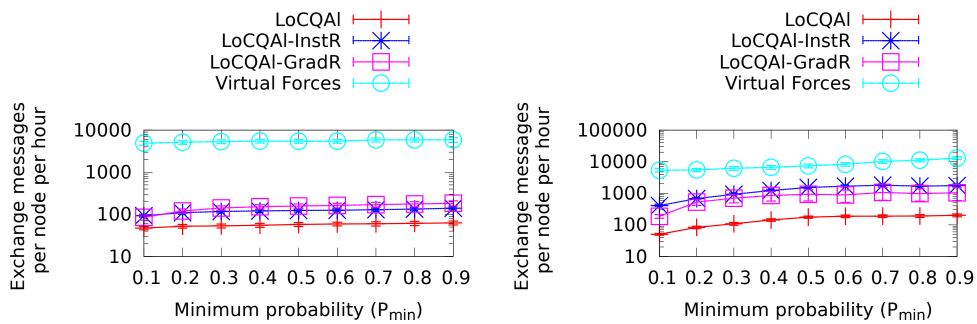

Figure 16: Number of exchange messages per node and per hour for a scenario with 10 targets (left) and 30 targets (right)

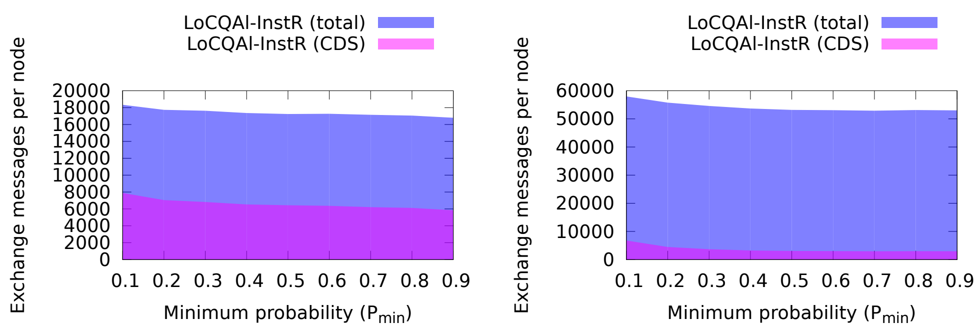

Figure 17: Absolute number of exchange messages per node of LoCQAl-InstR for a scenario with 10 targets (left) and 30 targets (right)

before more sensors occupy this target area is high (Figure 19 right). Virtual Forces performs better in dense scenarios than in sparse scenarios. When the density is low despite the fact that many nodes move towards the targets, the probability of a network disconnection is higher since the nodes are moving away from each other.

The results in Figure 20 show that for the same scenario Virtual Forces exhibits higher number of CDS computations. The reason is the same as in

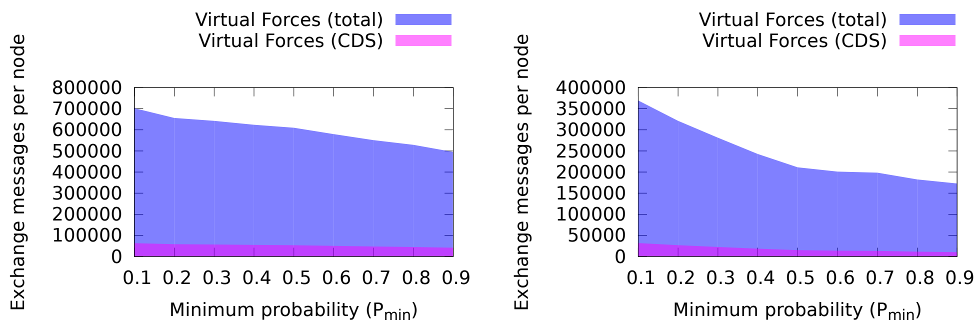

Figure 18: Absolute number of exchange messages per node of Virtual Forces for a scenario with 10 targets (left) and 30 targets (right)

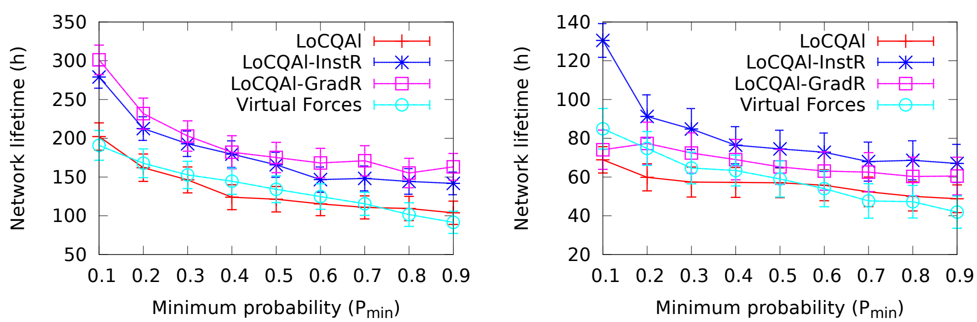

Figure 19: Network lifetime produced for a scenario with high node density (left) and low node density (right)

the previous scenario. The battery of the CDS nodes last for shorter time since they have been moved during the self-deployment phase. The lower the density, the more the energy consumption due to movements and, thus, the higher the gap with the other algorithms. The number of exchange messages is, also, much higher for Virtual Forces, as presented in Figure 21, since Virtual Forces need many iterations to finish and the nodes communicate very often with each other. LoCQAl-InstR and LoCQAl-GradR provide similar performance, while LoCQAl presents the lower number of messages, since it does not include any redeployment mechanism.

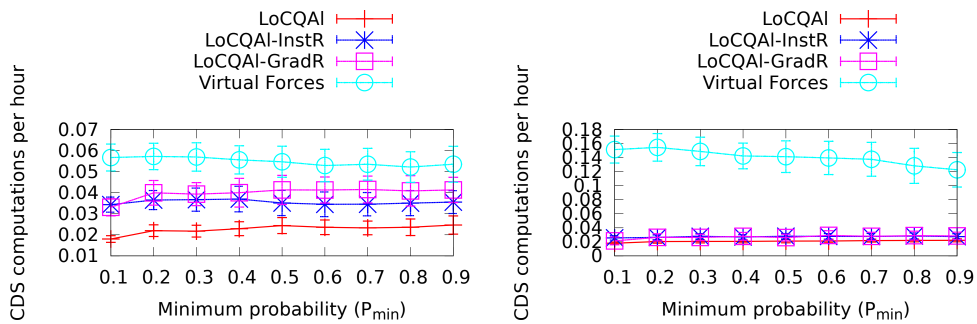

Figure 20: Number of CDS computation per hour for a scenario with high node density (left) and low node density (right)

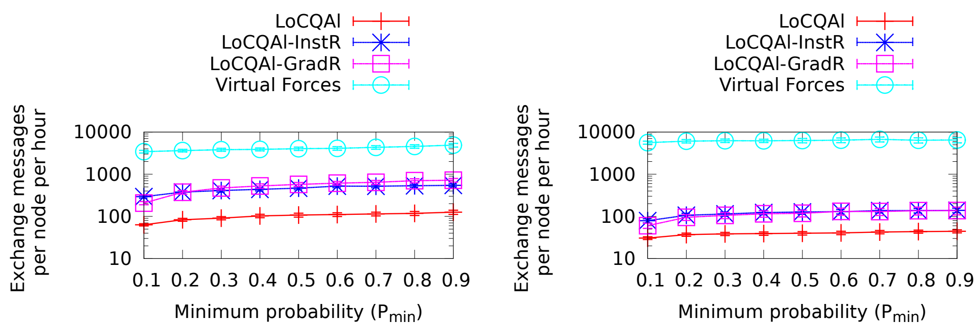

Figure 21: Number of exchange messages per node and per hour for a scenario with high node density (left) and low node density (right)

Finally, we measured the redeployment times of the approaches for the two examined scenarios. Because this measurement is based on produced network lifetime, we present the relative instead of the absolute values. As it is shown in Figures 22 and 23 LoCQAl-GradR decreases the redeployment delay by 40% on average, especially when the detection probability is low.

When P_{min} is low, the moving nodes reach their destination before the end of the monitoring round, since the monitoring rounds last for longer time (the nodes don't need to move and consume energy). The absolute redeployment delay of LoCQAl-InstR and Virtual Forces is constant regardless the value of the detection probability, but since the network lifetime decreases with P_{min} the relative delay looks to increase. For Virtual Forces the redeployment delay is independent of the number of targets as it mainly depends on the number of iterations of the algorithm.

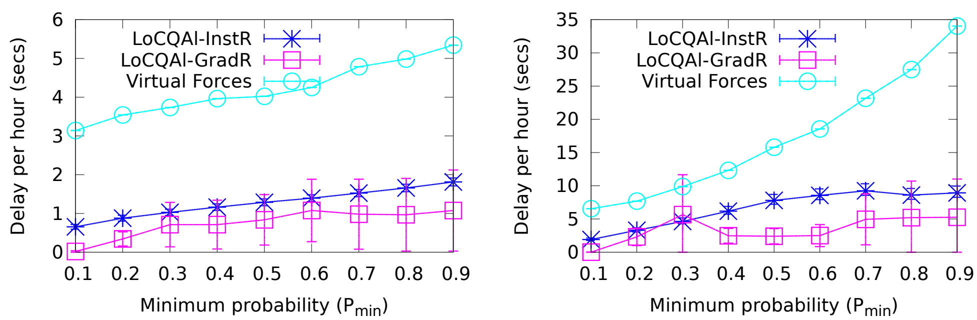

Figure 22: Redeployment delay per monitoring hour for a scenario with 10 targets (left) and 30 targets (right)

7.3. Encountering packet loss

In this section we evaluate the performance of our algorithms when a 50% packet loss occurs. Assuming that high packet loss we assess the algorithms' behavior to very noisy environments when many contentions occur. Particularly, the 50% packet loss stands for transmission or reception errors in each step of the algorithm (broadcasts, node discoveries, point-to-point communications etc.). For point-to-point communication, a node can retransmit a

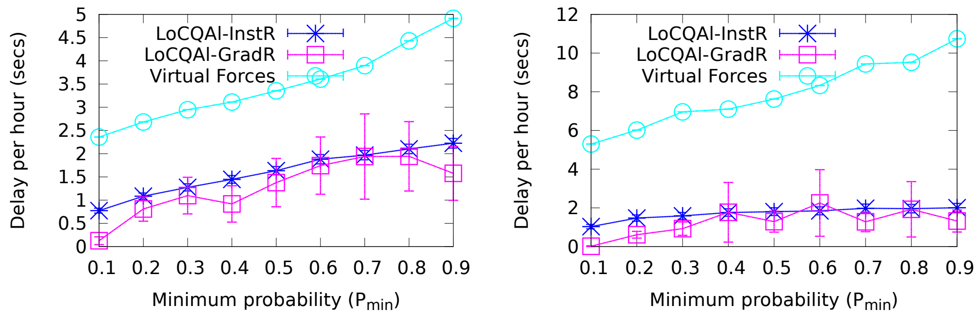

Figure 23: Redeployment delay per monitoring hour for a scenario with high node density (left) and low node density (right)

package if the first transmission was not successful. The second transmission has a 50% success rate as well. We compare our algorithms against the approach that considers static nodes assuming that no errors occur.

The results in Figures 24 and 25 show that the produced network lifetime has been reduced, but it is still higher than the static approach, especially when the detection probability is high. The algorithms are not able to move all the available nodes to achieve P_{min}, when this huge packet loss takes place. First of all, they cannot build correctly the cardinality sets since they cannot communicate with all their neighbors. Secondly, some of the selected nodes fail to move, since they are not able to receive a message from the corresponding head-node. LoCQAl-GradR looks to be more resilient to message fails when the target areas are populated by many sensors (i.e. scenarios with few targets and scenarios with high node density). In these cases the targets run out of sensors after many rounds giving the opportunity to LoCQAl-GradR to move some sensors at the end of each round. On the contrary, LoCQAl-InstR moves all the nodes before the first monitoring round. If the initial computation of the cardinality sets fails, no other improvement will be done.

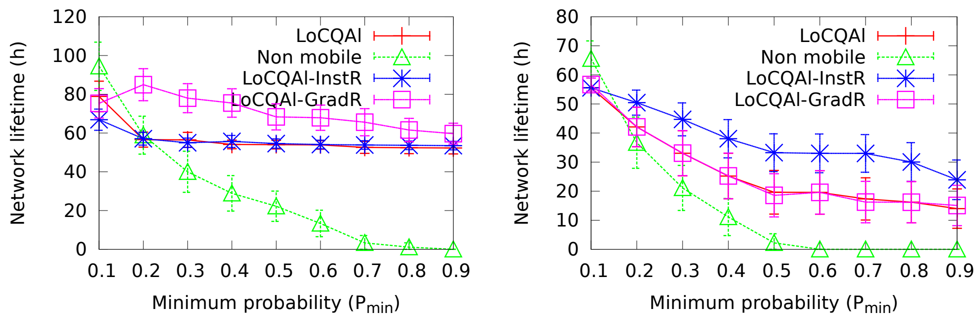

Figure 24: Network lifetime produced for a scenario with 10 targets (left) and 30 targets (right) and 50% packet loss

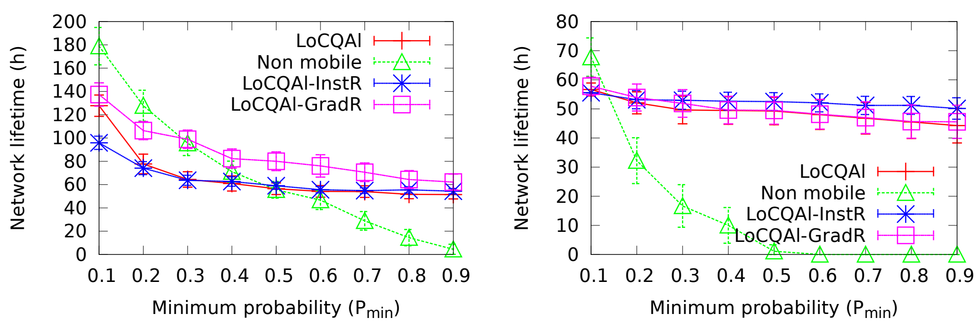

Figure 25: Network lifetime produced for a scenario with high node density (left) and low node density (right) and 50% packet loss

8. Discussion

In this section we make a discussion about how our algorithms would behave or what should be changed to deal with similar problems such as the presence of mobile targets instead of static ones, the presence of obstacles, CDS disconnections and relay node redeployment.

In presence of moving targets, the nodes covering the target should follow it all over the way keeping connectivity with at least one CDS node. In this case the CDS network has to be consisted of more nodes in order to avoid disconnections. The redeployment from target-to-target can be done taking into account the direction and the speed of the moving target. This is one of the applications that we still work on and it is part of our future work.

In presence of obstacles, the node should follow another path in order to reach the destination. We can distinguish two cases here: in the first case the nodes know the shape and the size of the obstacles, so the route and the total cost can be calculated in advance. In the second case where the nodes have no knowledge of the obstacles, obstacle avoidance algorithms from robotics can be used. Apparently, this would increase the redeployment delay and energy consumption.

If the network is disconnected or if a node does not receive a message from a neighbor, this means that the neighbor has failed and a procedure for partition detection and reconnection can be used such as the one described in [23].

In case of static targets, the network lifetime can be further extended by allowing a number of relay nodes to be moved towards poorly covered targets. Our LoCQAl redeployment algorithms can be applied for this purpose after the first CDS computation. The CDS nodes can attract or push non-CDS nodes from neighboring areas, balancing the number of nodes between the backbone network and the targets. An instance of this approach is illustrated in Figure 26. The left image shows a round of LoCQAl-InstR, while the right image depicts the same round when the relay nodes are allowed to move. All the nodes that do not cover any target and they are not parts of the backbone network, have been moved towards CDS nodes or target areas. Moreover, this scheme can be adapted according to the energy consumption model that has been assumed. For example, CDS nodes that are closer to the BS may have more nodes in their vicinity.

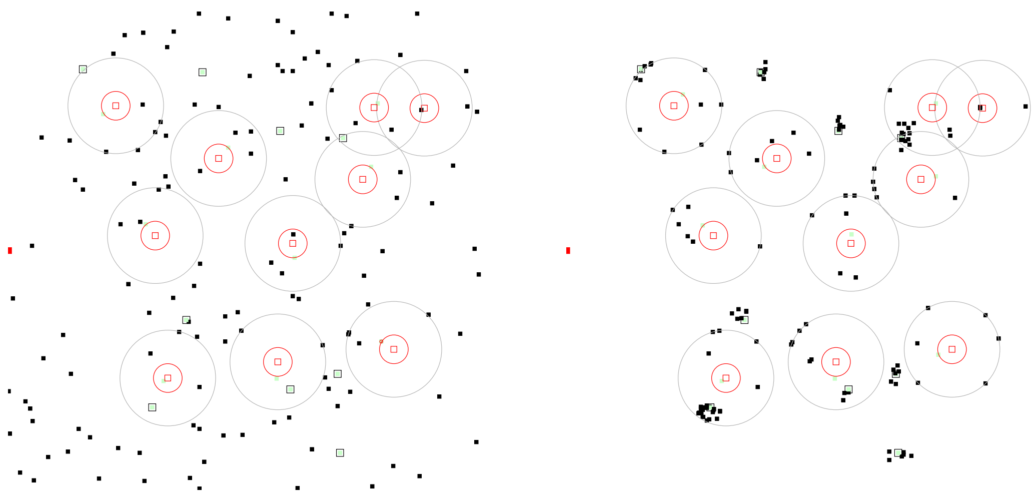

Figure 26: A depiction of the same round of LoCQAl-InstR and Full redeployment for a scenario with $P_{min} = 0.9$, 150 sensors, 10 targets and 10K m^2 terrain size

The effectiveness of the full redeployment approach is shown in Figure 27. In the case where the density is low, the network lifetime can be almost doubled since a large number of relay nodes is moved towards targets. The improvement is low for the high density scenario, where a few only relay nodes can be moved.

9. Conclusions & future work

In this paper, we introduced the minimum sampling quality problem for a set of targets in the network that are covered by mobile sensor nodes.

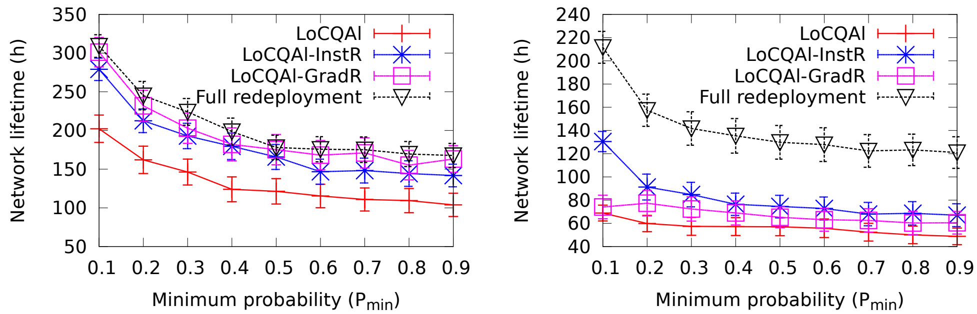

Figure 27: Network lifetime produced for a scenario with high node density (left) and low node density (right). A comparison with the full redeployment approach

We used a probabilistic coverage model in order to ideally detect an event happening in a close area. We presented analytical results describing that the probability of covering a single target by a number of static nodes is quite low, when the required probability of detection is high. For this reason we presented a localized solution that takes into account the capability of the nodes to move in order to prolong the network lifetime. The proposed approach achieves the desired coverage probability and satisfies connectivity with the base station. In order to enhance the performance of our approach, we proposed two redeployment schemes that balance the number of nodes between the target areas. The first approach exhibits high performance in terms of network lifetime, but it implies a long delay at the beginning of the operation. The second approach provides a good combination of performance and fast convergence time. Finally, we provided a discussion of how our algorithm could be adapted to satisfy different needs. A part of our future work is, also, to assess our approaches using a non-uniform node deployment and different detection needs per target.

References

[1] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Computer Networks 52 (12) (2008) 2292 – 2330.

[2] M. Cardei, M. Thai, Y. Li, W. Wu, Energy-efficient target coverage in wireless sensor networks, in: Proc. of INFOCOM 05, Vol. 3, IEEE, 2005, pp. 1976–1984.

[3] M. Kutylowski, Algorithmic challenges for sensor networks: foreword to algosensors 2007, in: Proceedings of the 3rd international conference on Algorithmic aspects of wireless sensor networks, ALGOSENSORS'07, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 1–5.

[4] N. Ahmed, S. S. Kanhere, S. Jha, Probabilistic coverage in wireless sensor networks, in: Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary, LCN '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 672–681.

[5] T. Rappaport, Wireless Communications: Principles and Practice, 2nd Edition, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[6] D. Zorbas, C. Douligeris, Connected coverage in wsns based on critical targets, Computer Networks 55 (6) (2011) 1412 – 1425.

[7] M. Hefeeda, H. Ahmadi, A probabilistic coverage protocol for wireless sensor networks, in: Network Protocols, 2007. ICNP 2007. IEEE International Conference on, 2007, pp. 41 –50.

[8] S. A. R. Zaidi, M. Hafeez, D. C. McLernon, M. Ghogho, A probabilistic model of k-coverage in minimum cost wireless sensor networks, in: Proceedings of the 2008 ACM CoNEXT Conference, CoNEXT '08, ACM, New York, NY, USA, 2008, pp. 33:1–33:2.

[9] G. Xing, X. Chang, C. Lu, J. Wang, K. Shen, R. Pless, J. O'Sullivan, Efficient coverage maintenance based on probabilistic distributed detection, Mobile Computing, IEEE Transactions on 9 (9) (2010) 1346 –1360.

[10] B. Carter, R. Ragade, A probabilistic model for the deployment of sensors, in: Sensors Applications Symposium, 2009. SAS 2009. IEEE, 2009, pp. 7 –12.

[11] I. Cardei, M. Cardei, Energy efficient connected coverage in wireless sensor networks, Int. J. Sen. Netw. 3 (3) (2008) 201–210.

[12] M. Chaudhary, A. K. Pujari, Q-coverage problem in wireless sensor networks, in: ICDCN '09: Proceedings of the 10th International Conference on Distributed Computing and Networking, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 325–330.

[13] Y. Gu, Y. Ji, J. Li, B. Zhao, Qos-aware target coverage in wireless sensor networks, Wirel. Commun. Mob. Comput. 9 (12) (2009) 1645–1659.

[14] D. Zorbas, T. Razafindralambo, Wireless sensor network redeployment under the target coverage constraint, Research Report RR-7818, INRIA (Nov 2011).
URL `http://hal.inria.fr/hal-00646342/en`

[15] R. Falcon, X. Li, A. Nayak, I. Stojmenovic, The one-commodity traveling salesman problem with selective pickup and delivery: An ant colony approach, in: Evolutionary Computation (CEC), 2010 IEEE Congress on, 2010, pp. 1 –8.

[16] G. Wang, G. Cao, T. F. La Porta, Movement-assisted sensor deployment, IEEE Transactions on Mobile Computing 5 (2006) 640–652.

[17] M. Garetto, M. Gribaudo, C.-F. Chiasserini, E. Leonardi, A distributed sensor relocatlon scheme for environmental control, in: Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on, 2007, pp. 1 –10.

[18] F. Kribi, P. Minet, A. Laouiti, Redeploying mobile wireless sensor networks with virtual forces, in: Proceedings of the 2nd IFIP conference on Wireless days, WD'09, IEEE Press, Piscataway, NJ, USA, 2009, pp. 254–259.

[19] P. Bergamo, G. Mazzini, Localization in sensor networks with fading and mobility, in: Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on, Vol. 2, 2002, pp. 750 – 754 vol.2.

[20] J. Carle, D. Simplot-Ryl, Energy-efficient area monitoring for sensor networks, Computer 37 (2) (2004) 40 – 46.

[21] Wifibots, `http://www.wifibot.com/`.

[22] Battery energy storage, `http://www.allaboutbatteries.com/Energy-tables.html`.

[23] K. Akkaya, F. Senel, Detecting and connecting disjoint sub-networks in wireless sensor and actor networks, Ad Hoc Networks 7 (7) (2009) 1330 – 1346.