Localized Sensor Area Coverage With Low Communication Overhead

Antoine Gallais, Jean Carle, David Simplot-Ryl, Ivan Stojmenovic

Localized Sensor Area Coverage With Low Communication Overhead

Antoine Gallais, Jean Carle and David Simplot-Ryl
IRCICA/LIFL, Univ. Lille 1,
INRIA Futurs, France
Email: {gallais,carle,simplot}@lifl.fr

Ivan Stojmenović
Computer Sciences, SITE,
University of Ottawa, Canada
Email: ivan@site.uottawa.ca

Abstract

We propose several localized sensor area coverage protocols, for arbitrary ratio of sensing and transmission radii. Sensors are assumed to be time synchronized, and active sensors are determined at the beginning of each round. The approach has a very small communication overhead since prior knowledge about neighbor existence is not required. Each node selects a random timeout and listens to messages sent by other nodes before the timeout expires. Sensor nodes whose sensing area is not fully covered (or fully covered but with a disconnected set of active sensors) when the deadline expires decide to remain active for the considered round, and transmit a message announcing their activity status. There are four variants in our approach, depending on whether or not negative and retreat messages are transmitted. Experimental results with ideal MAC layer show that, for a similar number of selected active sensors, our methods significantly reduce number of messages to decide activity compared to existing localized protocol, We also consider a MAC layer with collisions, and show that existing compared method, for dense networks, fails to cover the area reasonably. Our methods, however, still remain robust in terms of high area coverage with reasonable amount of active nodes, despite some message collisions.

1 Introduction

Acquiring numerical pieces of information straightly from our environment has become possible and affordable since recent advances in micro-electro-mechanical systems (MEMS), digital electronics, and wireless communications have enabled the development of lowcost, lowpower, multi functional sensor devices [10]. These devices can operate autonomously to gather, process and transmit information about the area which they are deployed on.

A sensor network is a set of nodes in which a battery, a sensing and a wireless communication device are embedded [1]. Densely deployed over hostile or remote environments, their self-organization should provide full monitoring and pertinent data collection so that further heavy computation and analysis tasks could be achieved by better equipped machines. Energy is the most critical resource considering the irreplaceable batteries of the sensor nodes. In order to increase their lifespan, these objects are allowed to turn into sleep mode as soon as they are not required for the local monitoring task. Indeed, monitoring redundancy can be used to switch off some nodes. The ensuing issue consists in these nodes deciding themselves whether to turn off or not while preserving full area coverage by a connected subset of nodes.

Several centralized and distributed approaches have already been proposed in literature. In centralized solutions, the information about topological changes in dynamic networks must be propagated throughout the network, to maintain the information needed for each node to make decision. A number of distributed protocols relax this full information propagation but use instead a wave type of computation and communication, memorization at nodes, unbounded delays or have other problems. Localized solutions have significantly lower communication overhead since no global view of the network is required. We consider only fully localized protocols so that solutions can be applied in sensor networks of any size and density. In a localized protocol, each node makes its activity status decision solely based on decisions made by its communication neighbors. Moreover, in fully localized protocol, decisions are not impacted by decisions made at distant nodes (e.g. in clustering type protocol, where nodes wait until some decisions arrive and unblock the decision making criterion). In fact, we restrict the impact to only nodes whose sensing ranges overlap. Further, we require each node to send only a small number of messages to its neighbors, to make the protocol reliable and bandwidth and power efficient. Localized protocols are needed for dynamic networks, whose topology changes due to mobility, changes in activity status, or changes due to failures or adding more nodes. In localized solutions, topological changes simply imply some modifications in the neighborhood of a node.

There exist several solutions that provide full coverage of considered area by active sensors. However, the connectivity among the set of nodes is often compromised or provided only for some ratios of sensing and communication ranges. Yet, the fully covering set should enable nodes to report their acquired sensing information. Ensuring such reports at least requires the set to be connected.

We addressed the area coverage problem in sensor networks with the idea of maintaining both connectivity and full area coverage, whatever the ratio between sensing and communicating ranges is. Sensors are assumed to be time synchronized. Synchronization can be achieved by applying some network protocols (see [8] for a survey) or by sending a training signal from the base station or another entity (e.g. helicopter) which reaches all sensors (see [12] for details).

Existing localized solutions [14] [7] rely on heavy initialization phases (using *hello* messages) which leads to increased communication overhead.

Our proposed solutions rely on low communication overhead in order to be suitable also for highly dense networks. No neighbor discovery is needed. Nodes wait a random timeout while receiving activity messages. Once timeout ends, the neighbor table of a node contains every node with shorter timeout and already made decision. The node then evaluates its coverage and connectivity, decides to be active or not and may announce its decision to its neighbors. After deciding to be active, nodes may hear from more active neighbors, and may then become fully covered. Such nodes may then change their mind by sending retreat message to their neighbors.

The primary goal of this article is to achieve similar performance in terms of ratio of active sensors in a given round as the best existing localized solution, while reducing significantly the number of messages for making decision at each node. Therefore, our protocols should have comparable network lifetimes along with increasing reliability of the network and decreasing message cost. Thus, they also decreases energy consumption. More messages lead to more collisions, consequently more retransmissions and so higher communication costs.

After introducing (in section 2) the notations that are used in this article, we will describe, in section 3, the existing localized algorithms that provide full coverage with a low message complexity. Our protocol will be precisely described in section 4 and its performances will be compared to the closest competitor ([14], as modified by Jiang and Dou [7]), in section 5. We show competitive ratios of active nodes with considerably fewer messages being sent in our protocols.

2 Foreword

2.1 Communication and sensing models

Each sensor has communication capacity. We assume equal communication ranges for each sensor, denoted CR. Therefore two sensors are communication neighbors, or simply neighbors, if and only if the distance between them is at most CR. Consequently, the communication is symmetric: if a node u can communicate with a node v, then v can also communicate with u. The degree of a node is the number of neighbors it has. The density of the network will be the average degree of a node in the network.

Nodes monitor an area using their various embedded sensing modules. The sensing radius of a node is denoted by SR. It is assumed that all sensors have the same sensing radius. Monitored area of a sensor is modeled as a disk of radius SR, centered on the node itself.

We consider three different ratios between communication CR and sensing SR radii, that are $SR = CR$, $SR < CR < 2SR$, or $2SR \leq CR$. Differences between these conditions will be detailed further in the article. Although theoretically $CR < SR$ is an option, such system does not exist in practice and were not studied so far.

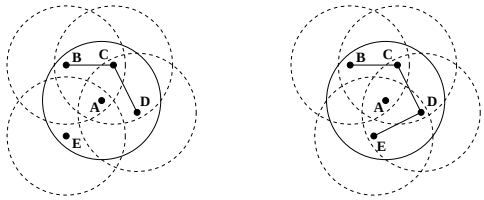
2.2 Assumptions

Before any activity, sensor networks must be deployed. The deployment can be either deterministic or random, depending on the application. We assume nodes to be randomly deployed. We assume that all nodes are static and have the same computation capabilities. We also assume that devices are time-synchronized [8] so that activity decisions can occur in rounds. Most existing algorithms assume that sensor nodes know their respective positions. The same assumption will be made in this paper since positioning issue has already been addressed in literature (see [9]). Sensors are distinguished by their position, and otherwise have no identities. They can be distinguished by assuming that they have a random number generator, and that neighboring sensors always select distinct random numbers.

2.3 Covering and connectivity

We assume that a node is aware of partial coverage of its sensing area by another node only if that node is its communication neighbor. For example, in case $CR = SR$, it is possible that sensing area of two nodes overlap, but they are not aware of that since they may not communicate. In other words, when $CR = SR$, node A is aware of overlap of its sensing area by a node B only when the distance between A and B is at most $CR = SR$. In Figure 1(a), sensing areas of nodes B, C, D and E fully cover the sensing area of node

A. Therefore, the monitoring disk of A is fully covered. Meanwhile, assuming $SR = CR$, this set is not connected since E is disconnected from B, C and D. Indeed, if A is allowed to switch off then the connectivity of the original network could be compromised. On Figure 1(b), covering nodes B, C, D and E are connected; therefore node A may be allowed to sleep.



(a) Covering by disconnected neighbors

(b) Covering by connected neighbors

Figure 1. Covering by neighbors

3 Related Work

A comprehensive literature review of existing solutions for sensor area coverage problem, including centralized, distributed, and localized solutions, is described in [11]. Here we describe several of algorithms that are closest in their assumptions, especially on localized approach.

A simple localized PEAS algorithm was proposed by Ye et all in [17]. Asynchronous networks are considered. Initially, all nodes are in sleep mode. Periodically, each sensor wakes up and sends a probing message. All active nodes within a given transmission radius R (which is identical for all nodes) receive this message. They then evaluate if their distance from the probing sensor is less than a range P, which is also equal for all nodes. This can be done due to signal strength or time delay measurements. If so, they advise the probing sensor accordingly, to allow it to continue sleeping for another period. Otherwise, no message is sent and the probing node decides to be active. Once it activates, it remains active until the end of its lifetime. This protocol is highly fault tolerant. However it does not ensure full area coverage.

Cai, Li and Wu [2] described a sensor area coverage protocol for asynchronous sensor networks. Their algorithm corrects and completes PECAS protocol [6] (which is an extension of PEAS [17]) which leaves holes in coverage. Each node maintains the portion of its area which is not covered by other sensors (net area). In [2], there are five node states (pre-wake, awake, overdue, pre-sleep and sleep). A node u makes its own decision and sends message to neighbors. Awake neighbors respond (attaching their location and remaining activity times), and u can decide to go back to sleep state or to move to Awake state (and informs neighbors). Nodes change their status by using both information about the net areas and timeout condition (see [2] for further details).

Zhang and Hou [18] described an efficient algorithm for selecting covering sensors in time synchronized network. Sensors periodically make new decisions about their active or sleeping status. In each round, a single sensor starts the decision process, which then propagates to the whole network. New sensors are selected so that the priority is given to sensors located near optimal hexagonal area coverage, obtained when the area is ideally divided into equal regular hexagons. The coverage is indeed quite good, given the distributed nature of the decisions. However, the need for a single sensor to start the process may cause problems in applying it, including increased latency. If several sensors start the process then the decisions at meeting points would be suboptimal. Another problem is that the original sensing area coverage may not be preserved (as shown by experimental results). In this article we consider only protocols that preserve full coverage of the originally covered area.

The algorithm presented in [16] divides the area into small grids, and then covers each grid with a sensor. Each sensor that can cover a grid maintains a list of other sensors that can also cover it, in a priority order. All sensors covering the same grid can communicate with each other. When sensor density is significant, sensors need a lot of memory and processing time to maintain priority lists, plus the communication overhead for making covering decisions in cooperative manner is nontrivial.

Hsin and Liu [5] investigated random and coordinated area-coverage algorithms. In their coordinated-coverage scheme, a sensor may decide to sleep after receiving permission from sponsoring neighbors, for the time such permission is given. A node that sponsors any other node must be active. The decisions are not synchronized, since each sensor can "negotiate" with its sponsors independently, and the scheme allows for several variants with (sophisticated) protocol details. Each sensor maintains its own delay counter, which is used for role alteration. Although the coordinated scheme by Hsin and Liu [5] has some desirable properties, such as localized behavior, it may select too many sponsor nodes to be active, since there is no coordination between nodes for the selection of as many as possible common sponsor nodes.

Carle et al. [3] proposed a localized scheme based on a relay selection phase. Every node selects a set of relays among its 1-hop neighbors. The relays cover an area as large as the area covered by whole neighborhood. Then, an activity decision is made based on a unique key. Any node which has the smallest priority in its neighborhood or which has been selected as relay by its neighbor with the smallest priority will decide to remain active. This decision allows connectivity to be preserved along with full area

coverage. However, the algorithm involves sending hello messages to learn 1-hop neighbors, and sending messages of extended size, informing neighbors about relays, which is considerably higher communication overhead than methods proposed in this article.

Tian and Georganas [14] proposed a solution for sensor area coverage in synchronous networks where sensing and transmission ranges are equal. It requires that every node knows the positions of all its neighbors before making its monitoring decision. At the beginning of each round, each node selects a time-out interval. At the end of the interval, if a node u sees that remaining neighbors (that have not yet sent withdrawal message) together cover its monitoring area, it transmits a "withdrawal" message to its neighbors and moves into sleep mode. Otherwise, u remains active, but does not transmit any message. The process repeats periodically to allow for changes in monitoring status. There are several problems in this protocol. Neighboring active sensors may fail without notice, and neighboring sensors may not activate, believing that the sensor is "alive" and monitoring. This problem can be resolved if neighboring information is exchanged at the beginning of each round [7].The other problem is that covering sensors may not be connected; thus, reporting to a monitoring station may not succeed. The authors also discuss the case of different sensing radii at each sensor.

Jiang and Dou [7] describe several improvements to the algorithm in [14]. They assume that $CR \geq 2SR$, and apply the criterion that a circle C is covered completely if perimeters of other circles covering it (only portions that are inside C) are fully covered by other covering circles. Nodes apply a random backoff before making decisions. In the algorithm presented in [7], at the beginning of each round, each node sends a hello message to inform about its position. The algorithm from [14] is then applied. This algorithm is the closest competitor to our new protocols. For fairness, we modify it in several ways, without any change on nodes decision. First, the perimeter coverage criterion was replaced by computationally more efficient method described in the next section. Next, we consider the protocol for general ratio of CR and SR, by adding similarly connectivity criterion when $CR < 2SR$. Experimental data in [7] show that this algorithm outperforms PEAS [17] with respect to the number of nodes needed in the coverage, while completely preserving sensing coverage of the original network.

4 Our contribution

Our approach is fully localized and can be applied in networks composed of time-synchronized devices knowing their positions. We assume arbitrary ratio of sensing and communicating radii. The main goal of our protocols is to have very low communication overhead. Indeed, no neighbor discovery phase is needed here. Neighbor knowledge is brought by activity messages. Similar to a Neighbor Elimination Scheme [13], nodes *wait* and listen to activity messages to *see*, once their timeouts end, if they will eventually be required for local full area coverage. We also consider adding retreat messages by nodes that first announced their activity status, but later on noticed they are not really needed after some new neighbors were discovered.

4.1 Timeout computation

Our protocols are based on a timeout scheme. When a round starts, every node selects a timeout and evaluates its coverage once its timeout expires. We assume, for simplicity, that any two neighboring nodes would select different random numbers, so that two nodes never attempt to send message at the same time.

While waiting for timeout to expire, nodes receive decisions made by neighbors with shorter timeouts. The positions and decisions of these neighbors are memorized, and used at the end of timeout to evaluate the coverage, and make appropriate decision. In dense networks, this may result in accumulating a number of decisions, especially if negative acknowledgments are also sent. To address this issue, nodes may evaluate the coverage upon receiving certain number of messages. It is clear that the probability that the size of uncovered area is zero increases with the number of received messages, thus a heuristic value can be used to decide the threshold properly. In case of not covering the area fully, the evaluation can be repeated after receiving few more messages. In this way, the decision to go to sleep mode may be made before the timeout expires, and simply wait for the time to announce it, without the need to memorize the remaining received messages.

4.2 Coverage evaluation

Each node can decide to sleep if its sensing area is covered by a set of connected nodes with lower timeout values. In this section, we discuss how to decide whether or not the monitoring area of a sensor is fully covered. Different evaluation schemes have already been proposed in literature. The perimeter based scheme used in [14] can not be applied as soon as communicating and sensing radii are different. We decided to apply a well known geometric theorem, which is applicable for any ratio of sensing and covering radii. Moreover, it is applicable to any shape of monitored region by a sensor, which was used by us to deal with border issues.

The covering criterion has been already applied in [15, 18]. It efficiently confirms whether or not a sensing region is fully covered by other sensing regions. It is applied on the borders of the covering regions. In our case, these borders are normally circles, and we will express the theorem first in

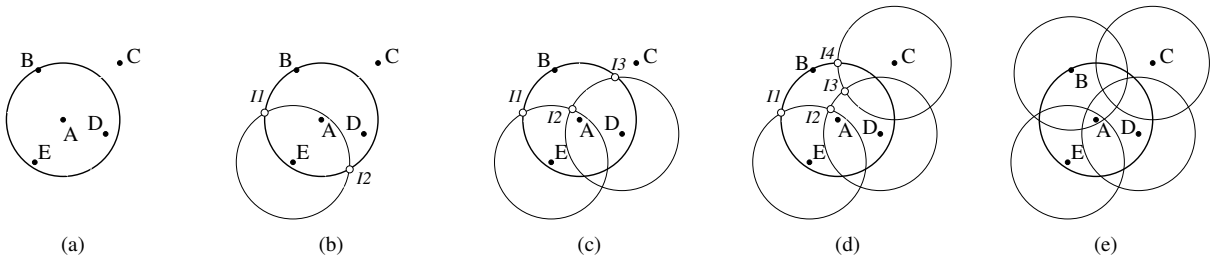


Figure 2. Intersection-based coverage evaluation scheme

circle terminology. The efficient criterion for full coverage is as follows (see [15] for proof):

Theorem 1 *If there are at least two covering circles and any intersection point of two covering circles inside the sensing area is covered by a third covering circle, then the sensing area is fully covered.*

In other words, a disk d is fully covered by other disks if and only if every intersection point of two disks $d1$ and $d2$ inside d is covered by another disk $d3$. In addition, intersection points of any other disk $d1$ with d must also be fully covered (be inside) a third circle $d2$. For instance, in figure 2, node A evaluates its coverage, knowing neighbors B, C, D and E (see figure 2.a). We can observe that the communicating range is greater than the sensing radius since C is a neighbor of A. Figure 2.b shows the two intersections $I1$ and $I2$ of monitoring disk centered at E with the one centered at A. $I2$ is covered by disk centered at D (see figure 2.c) but two other intersection points appear (uncovered intersection points are denoted by $I1$, $I2$ and $I3$ in that figure). Figure 2.d shows that C covers $I3$ but then, two more intersection points are added (denoted by $I3$ and $I4$). At last, B covers these four intersection points, and does not create new one (not covered by any of previous circles). Therefore A is fully covered. This method has many advantages. It is quick to compute and works for any ratio of sensing and communicating ranges.

We have also modified this scheme to avoid the problem of border nodes effect. We distinguish the original sensing coverage and the monitored area. The monitored area is a geometrical figure, such as a rectangle for instance, inside which sensors are deployed. Problems will occur if the border effects are ignored. As exposed in [14, 3], the nodes located near the borders of the monitored area have no other choice but to be active in each round since they are the only ones able to monitor further pieces of their own sensing coverage. To avoid that, we assume that nodes are aware of the field they have to monitor, and adjust the covering criterion as follows.

The coverage criterion is extended to take into account the intersections of sensing areas with the deployment or

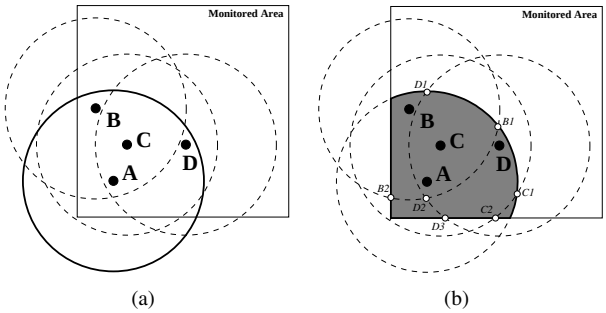


Figure 3. Area aware evaluation scheme

monitoring area. Nodes simply find the intersection of their sensing area with the monitoring area, and consider it as their revised sensing area. For example, in figure 3(b), node A considers shaded area (the mentioned intersection) as its new sensing area. If node A applied the same reduction to its neighbors, special cases, with regions intersecting along line segment rather than in a singe point, would occur. To keep algorithm simple for implementation, node A preserves covering regions of neighbors as circles. This does not impact the accuracy of this criterion. The modified covering region prevents border nodes from being active at every round. For instance, figure 3(a) shows four nodes, A, B, C and D within a square area. For preserving the coverage of the whole region that it is able to cover, A should be active in each round. However, if only the rectangle is to be monitored, then A could get into sleep mode. As observed in figure 3(b), circle centered at C covers all intersection points created by other circles and revised monitoring region of A, while $C1$ and $C2$ are covered by circle centered at D.

This method is computationally very efficient. It allows dynamic evaluation of the criterion, by keeping the list of nodes to be covered by new active neighbors. As soon as the criterion is satisfied, the verification of this condition can be terminated. In this way, the criterion does not require excessive computation time in case of highly dense networks, even if negative acknowledgments are sent by all neighbors.

4.3 Connectivity conservation

The problem of connectivity among the set of active nodes is also addressed by our algorithm. Many existing works assume that connectivity is ensured once the communicating range is at least twice the sensing range [18]. A simple additional criterion is introduced here. If $CR < 2SR$ then a node can decide to turn off if and only if its neighbors fully cover it and are also connected [4].

Figure 1(a) shows a disconnected coverage of node A by neighbors B, C, D and E. Considering $CR = SR$, A is not allowed to turn off. On the contrary, on Fig. 1(b), its neighbors are connected and A can so turn into sleep mode. Connectivity is still preserved.

4.4 Decision announcement

After verifying the coverage condition, each node decides whether or not to send a message. Messages contain geographic position of nodes and the activity status. In our four proposed variants, at the end of the timeout, if a node decides to be active, it sends a positive acknowledgment so that neighboring nodes with higher timeout values can consider it for their coverage evaluation. We consider variants with and without sending negative acknowledgment for nodes that decide to enter sleep mode.

For instance, figure 4(a) shows a configuration of 7 nodes ($CR = SR$), their timeouts corresponding to their numberings. Timeouts are used in the following lines to designate nodes. No identifier is needed since positions of the nodes, which are unique, are included in the activity messages. In figure 4(b), the first five nodes (nodes 1-5) have decided to be active since their sensing disks are uncovered. $6th$ node then decides to switch off since it is fully covered by a connected set of nodes. The last deciding node can benefit from a negative acknowledgment of node 6. Indeed, node 7 has received positive messages from nodes 1 and 2 but is not aware of nodes 3, 4 and 5. Yet, decision of node 6 means that the dashed portion of figure 4(c) is covered by sensors with shorter timeouts. Then, node 7 learns from the negative acknowledgment of node 6 that its monitoring disk is fully covered by 1, 2 and unknown other nodes. Hence, both positive and negative messages bring the same information on coverage. We found out that there was still significant redundancy among the set of active nodes, which worsened with density increase. This is due to the fact that decisions are made only once during a round. The capability for a node to change its mind was so introduced. A message, called retreat, announces such a decision. We combined the three types of messages in different ways and describe four new protocols:

- **Positive-only**, *denoted by PO*: nodes that decide to be active send exactly one message. Nodes that decide to be passive do not send any message.

- **Positive and Negative**, *denoted by PN*: every node sends exactly one message, a positive or a negative acknowledgment respectively for an active or a passive status.

- **Positive and Retreat**, *denoted by PR*: Same as PO except that a node that has already decided to be active can later on learn about newly announced active nodes, and may decide to enter sleep mode; such nodes send one retreat message.

- **Positive, Negative, and Retreat**, *noted as PNR*: all decisions by all nodes are transmitted; thus each node sends one message corresponding to the original decision on active or sleep status. Nodes with originally positive decision may switch to sleep mode later and send one retreat message.

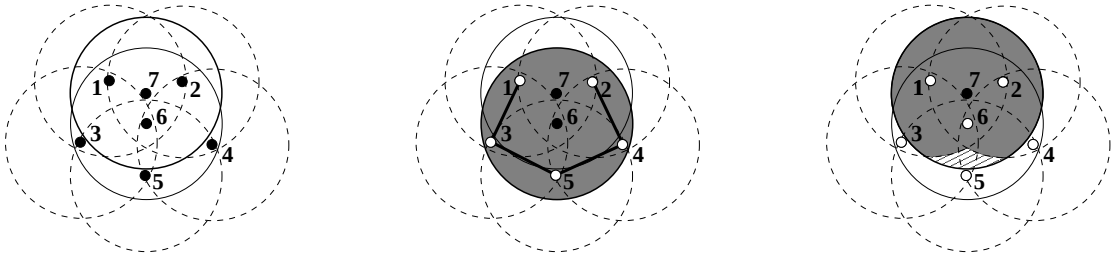
Each message contains the position of given node and its decision.

5 Performance evaluation

Experimental results were obtained from randomly generated connected networks. Nodes are deployed over a 50*50 rectangle area. The communication range (CR) equals to 10 while sensing radius (SR) varies to observe the compared algorithms under two conditions: $SR \leq CR < 2SR$ and $2SR \leq CR$ ($SR = 4$). The first condition is ensured grace to nodes having equal communicating and sensing radii (which equal to 10). The important point is that the two mentioned conditions depend on whether or not the connectivity criterion should be applied so that the set of active nodes can be connected. Simulations were launched over three different densities, 30, 50 and 70 (respectively 240, 400 and 560 nodes). Energy levels are initially fixed at 100. An active node loses 1 battery unit. Energy levels of sleeping nodes remain unchanged. Simulation ends as soon as the set of nodes with remaining energy is no longer connected.

In the first set of experiments, we assume ideal MAC layer (without collisions), so that every emitted message is received correctly by all neighbors. Nodes compute random timeouts at the beginning of each round. Then, a round starts and every node decides of its activity status as per corresponding protocol. Comparison was made with the protocol, called TGJD, which is our modification of protocols proposed in [7] [14]. We modified the protocol to provide fair comparison with our new protocols PO, PN, PR and PNR, making all protocols same except the main difference in messages being sent.

We measure the percentage of active nodes, average number of messages per node, and percentages of original sensing and monitoring area coverage.



(a) 7 nodes with their timeouts

(b) First five have decided and 6 is fully covered by a connected set

(c) Negative acknowledgment from 6 helps 7 to know about dashed area coverage

Figure 4. Importance of Negative acknowledgment

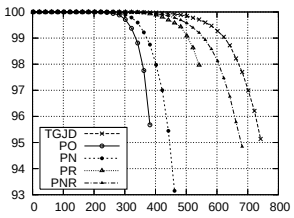


Figure 5. Area coverage versus time (Density=50, SR=CR)

5.1 Area coverage

We have measured how much coverage considered algorithms were able to provide with respect to the deployment area (the rectangle). Due to relatively high densities considered, sensors are able to cover fully (100%) the deployment area. The results presented here refer to this coverage, which allows border sensors to periodically sleep. We therefore leveraged the impact of border regions and obtained results that we believe reflect better the internal dynamics of the network.

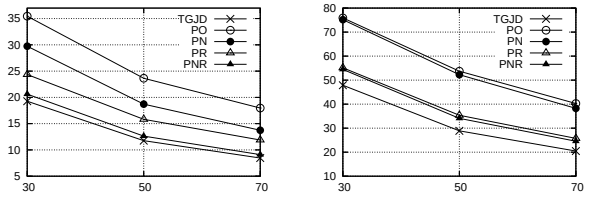
The percentage of coverage is computed at the end of every round. The computation is done by considering grid reference points which are spread at distance of one tenth of the communication range. The reported percentages refer to percentage of these grid points being covered.

All considered schemes are efficient since the deployment area remains fully covered for a long time as shown in Fig. 5. It can be observed that TGJD is best in this respect, followed by PNR, PR, PN and PO. Our protocols, however, are competitive with TGJD and all avoid sudden fall after round 100. That would be the case if our border region handling scheme was not used (all border nodes would be active at every round, loosing one energy point and so all dying about 100 rounds).

5.2 Active nodes

It is important to have as few active nodes as possible while not inducing much communication overhead. Figure 6 shows the average percentage of active nodes, for three different densities and SR over CR ratios, for TGJD and our four variants, in the first round. We can observe that, for $CR = SR$ and density 30, TGJD has 19% of active nodes, followed by PRN with 20%, PR with 24%, PN with 30%, and PO with 35%. Similar results are obtained for other densities and CR over SR ratios. Interestingly, retreat messages show more benefits than negative acknowledgements. Also, two of our protocols are very competitive with TGJD (PR remain within 5% difference while PRN remains within 1% in all cases). TGJD is expected to be the best since it uses full knowledge of neighborhood to make decisions and excessive messaging to gain such knowledge.

Note that percentages of active nodes decrease as density increases. Intuitively, more active nodes are needed when SR is decreased. This is confirmed by Figure 6(b).



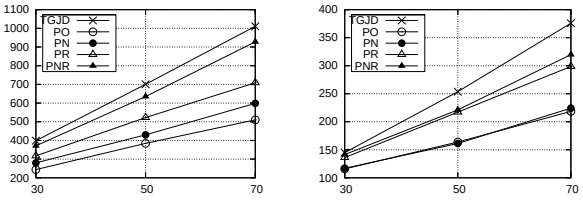
(a) $SR = CR$ (b) $2SR \leq CR$

Figure 6. Percentage of active nodes versus density

5.3 Network lifetime

For a given network, simulation runs until the set of active nodes becomes disconnected. The remaining energy of

nodes depends only on their past activity. The energy cost of messages is not considered. Figure 7 shows the average number of rounds (the lifetime) during which the set of active nodes remains connected. These data are in accordance



(a) $SR = CR$ (b) $2SR \leq CR$

Figure 7. Lifetime versus network density

with the number of active nodes needed by each protocol to achieve full coverage. More active nodes lead to shorter lifetimes.

5.4 Average number of messages per node

Assuming no collisions, we quantify the amount of sent messages in each protocol. Let N be the number of deployed sensor nodes, A be the number of active nodes and S be the number of sleeping nodes. We have $N = A + S$.

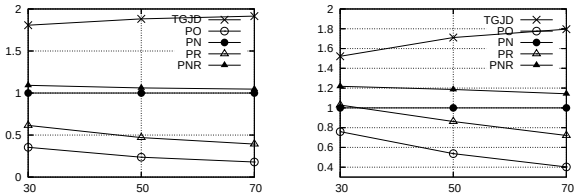
Neighbor discovery phase of [7] requires N *hello* messages. Furthermore, during the evaluating phase, only off-duty nodes emit one withdraw message. Thus the total number of messages in protocol TGJD is equal to $N + S$. Our variants PO and PN respectively induce A and $A + S = N$ messages. Let R be the number of retreat messages sent. PR and PNR variants then require $A + R$ and $N + R$ messages, respectively.

Consider an example. As observed in figure 6(a), for networks whose density is 50, assuming $CR = SR$, the percentage of sleeping nodes is nearly 88% for TGJD algorithm. Thus we have $S = 0.88 * N$. The total number of messages in protocol TGJD is approximately $N + S = 1.88 * N$. In protocol PO, the number of sent messages is $A = 0.24 * N$, which is 7 times less than in TGJD. Protocol PN requires N messages. Let us now consider variants using retreat messages. In the considered case, it was measured that $R = 0.08 * N$. Protocol PR then requires $A + R = 0.24 * N + 0.08 * N = 0.32 * N$ messages. Finally, protocol PNR uses $N + R = 1.08 * N$ messages. Our protocols PO, PN, PR, PNR therefore use 13%, 53%, 17% and 57% of messages used by protocol TGJD, respectively. Among our variants, PR strongly reduces the communication overhead (nearly 6 times) while requiring only 5% more active nodes, compared to TGJD. Diagrams in Fig. 8(a) and Table 1 show that the more dense the network is, the more messages are generated by TGJD protocol. At the same

$SR < CR < 2SR$	$Density = 50$	$Density = 70$
TGJD	1.8	1.9
PO	0.3	0.2
PN	1.0	1.0
PR	0.6	0.5
PNR	1.1	1.1

Table 1. Average number of messages per node

time, our protocols PO, PR and PRN show reduced number of messages with increased density. In the next section, we show the impact of message cost when a MAC layer with collision considerations is introduced.



(a) $SR = CR$ (b) $2SR \leq CR$

Figure 8. Average number of messages versus density

5.5 Impact of a realistic MAC layer

In order to quantify the benefits of our new protocols with respect to existing TGJD approach, we repeated our experiment on a more realistic MAC layer, which considers message collisions as message failures. We want to show that an approach with less messages should be preferred in environments where loss of messages is not negligible. We have added a contention window of size CW and a timeout for each node before it can send any message. Any node randomly picks up an integer value between 0 and CW. Then, a node can neither receive two messages at the same time nor receive a message while transmitting. We selected a contention window of size 32, in accordance to IEEE 802.11 standard.

Collisions were considered in both neighbor discovery and withdrawal messages of TGJD protocol. Table 2 shows the impact of using a contention window with TGJD protocol. The expected degree of a node is the number of received *hello* messages, if collisions were not considered. The real degree of a node is the average number of neighbors a node has really discovered (no collisions occurred). During the withdrawal phase, we have counted the number of withdrawal messages sent to a given node (Sent withdrawals)

Density	Neighbors		Withdrawals	
$(SR = CR)$	Expected	Real	Sent	Received
30	25.1	14.5	11.6	7
50	41.9	20.6	20.5	10.4
70	58.7	26.2	28.9	13.3

Table 2. Collisions in protocol TGJD

Density 50	DA Coverage (%)	Active nodes (%)
$SR = CR$		
TGJD	69.5	4.8
PO	100	39
PN	100	33.9
PR	99.4	10.8
PNR	98.2	9
$SR < CR < 2SR$		
TGJD	70.6	7.6
PO	100	44.6
PN	100	39.9
PR	95.8	12.6
PNR	91.1	10.5

Table 3. Coverage and active nodes

and the number of withdrawal messages this node has effectively received (Received withdrawals). Loss of messages is in accordance with the size of the contention window we have used. For instance, at density 50, only half of sent *hello* messages are received. We have counted every sent message, so that nodes that have not been discovered during the *hello* phase can send a withdrawal message. This message will not be considered by the receiving node (for the activity decision) but it increases the probability of collision within the communication zone. Since node has more chances to turn off with increased density, number of emitted withdrawal messages increases (from 11.6 at density 30 to 28.9 at density 70). Furthermore, the percentage of received withdrawal messages decreases and it has an important impact on TGJD performance.

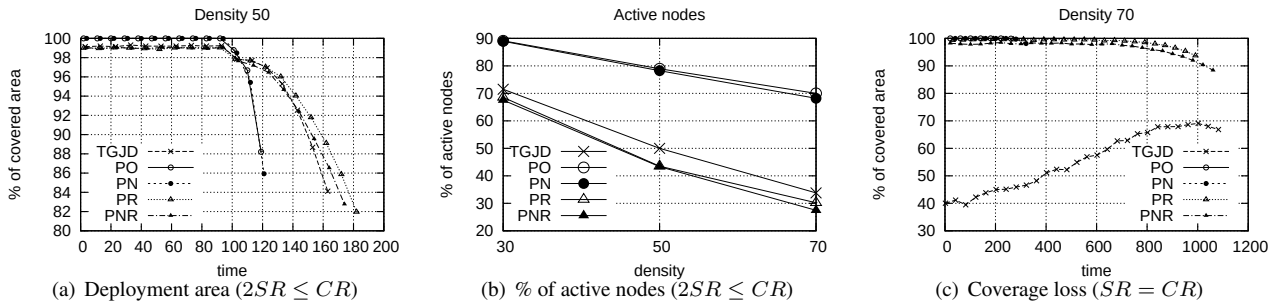
These results are shown in figure 9. For density 50 and $2SR \leq CR$, PR and PNR variants outperform TGJD in terms of number of active nodes (and eventually the lifetime). Fig. 9(b) shows that PR and PNR involve around 43% of active nodes while TGJD requires 50%. Moreover, Fig. 9(a) shows that the coverage provided by every protocol are nearly identical, except for PO and PN that preserve 100% of coverage despite loss of some messages. PR and PNR variants can not preserve full coverage since some of retreat messages are not received, and two simultaneous retreat messages by two neighbors may create a coverage hole. Tab. 3 shows that for other SR/CR ratios and density 50, the coverage of TGJD is worse than any one of our four variants. While TGJD provides about 70% coverage of deployment area (DA), all our variants provide over 91%. This is due to the lack of active nodes with TGJD protocol (only 4.8 when $SR = CR$ compared to 10.8 for PR variant). Note that when $SR = CR$, the four variants of our protocol provide more than 98% of deployment area coverage.

Figure 9(c) shows the evolution of the coverage with time for a network of density 70. It clearly raises the issue of preserving coverage once not all withdrawal messages are received. TGJD performance drops to as low as 40% of area coverage, in the very first round. The fact that the coverage of TGJD increases with time is due to decrease of loss of messages since failed nodes no longer send messages. Withdrawal messages are better received and so TGJD performance is getting closer from its level with lower density. Our variants can maintain nearly full coverage. Moreover, we can notice that average lifetime is between 1000 and 1100, like for TGJD. Therefore, as network density increases, enabling communication between nodes becomes more and more difficult. Our protocols PO and PN can preserve nearly full coverage (100% while PR and PNR cover more than 95% of area) and so maintain the good behavior of the monitoring application. These results demonstrate the robustness of our algorithm compare to TGJD when loss of messages is considered.

6 Conclusion

We have proposed a localized algorithm for maintaining connected area coverage under various ratios of communicating and sensing radii. In addition to providing competitive ratios of active nodes under various conditions, our approach induces very low communication overhead, and shows robustness when message collisions are considered. The main novelty of our approach is to put emphasizes on positive rather than negative acknowledgments. If a positive acknowledgment is missed, neighbors merely increase their chance of remaining active (possibly unnecessarily) and coverage is preserved. Low percentage of active nodes means low chance of collisions, and reduced impact of collisions. The existing approach, termed here TGJD protocol, relied on withdrawal messages (negative acknowledgments) sent by nodes that decide to sleep, and *hello* messages sent in preprocessing step. Collided withdrawal message is interpreted as the corresponding area remaining covered, and sensor may wrongly decide to sleep, leaving coverage holes. The problem worsens with increased node density. *Hello* messages increase the problem since missed neighbors afterward just contribute to more collisions.

Various improvements to presented protocols are possible. One can introduce an energy-consumption model that



(a) Deployment area ($2SR \leq CR$) (b) % of active nodes ($2SR \leq CR$) (c) Coverage loss ($SR = CR$)

Figure 9. Impact of MAC layer

considers cost of messages. That would allow more accurate lifetime improvement measures. Improved versions of timeout functions may also be considered so that nodes with higher energy levels make sooner decisions thus being active more often.

We plan to extend our protocols by considering realistic physical layers for sensing and communications, and by considering k-coverage rather than simple sensor coverage for the deployment area. Both extensions are based on ideas presented here for maintaining low communication overhead.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[2] Y. Cai, M. Li, and M. Wu. Acos: A precise energy-aware coverage control protocol for wireless sensor networks. In *International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, Wuhan, China, Dec. 2005.

[3] J. Carle, A. Gallais, and D. Simplot-Ryl. Area coverage in wireless sensor networks based on surface coverage relay dominating sets. In *Proc. 10th IEEE Symposium on Computers and Communications (ISCC 2005)*, Cartagena, Spain, June 2005.

[4] J. Carle and D. Simplot. Energy efficient area monitoring by sensor networks. *IEEE Computer Magazine*, 37:40–46, Feb. 2004.

[5] C. fan Hsin and M. Liu. Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms. In *IPSN'04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 433–442, New York, NY, USA, 2004. ACM Press.

[6] C. Gui and P. Mohapatra. Power conservation and quality of survelliance in target tracking sensor networks. In *ACM Mobicom*, pages 129–143, Philadelphia, PA, U.S.A., Sept. 2004.

[7] J. Jiang and W. Dou. A coverage preserving density control algorithm for wireless sensor networks. In *ADHOC-NOW, Vancouver, LNCS 3158*, pages 42–45, July 2004.

[8] P. B. K. Romer and L. Meier. Time synchronization and calibration in wireless sensor networks. *Handbook of Sensor Networks: Algorithms and Architectures (I. Stojmenović, ed.)*, pages 199–238, 2005.

[9] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AOA. In *Proc. IEEE INFOCOM'2003*, San Francisco, USA, 2003.

[10] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[11] D. Simplot-Ryl, I. Stojmenović, and J. Wu. Energy-efficient backbone construction, broadcasting, and area coverage in sensor networks. *Handbook of Sensor Networks: Algorithms and Architectures (I. Stojmenović, ed.)*, pages 343–380, 2005.

[12] I. Stojmenović and S. Olariu. Data centric protocols for wireless sensor networks. *Handbook of Sensor Networks: Algorithms and Architectures (I. Stojmenović, ed.)*, pages 417–456, 2005.

[13] I. Stojmenović, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(12), Dec. 2001.

[14] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proc. First ACM Intl. Workshop on Wireless Sensor Networks and Applications*, pages 32–41, Sept. 2002.

[15] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proc. 1st Intl. Conf. on Embedded Networked Sensor Systems (SenSys)*, pages 28–39, Los Angeles, CA, Nov. 2003. ACM.

[16] T. Yan, T. He, and J. A. Stankovic. Differentiated surveillance service for sensor networks. In *First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, pages 51–62, Los Angeles, California, November 2003.

[17] F. Ye, G. Zhong, S. Lu, and L. Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. In *Proc. 23rd International Conference on Distributed Computing Systems (IEEE ICDCS)*, 2003.

[18] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc and Sensor Wireless Networks, an International Journal, 1 (1-2), 89-124.*, 2005.