Localized Sensor Area Coverage with Low Communication Overhead

Antoine Gallais, Jean Carle, David Simplot-Ryl, and Ivan Stojmenović, *Fellow*, *IEEE*

Abstract—We propose several localized sensor area coverage protocols for heterogeneous sensors, each with arbitrary sensing and transmission radii. The approach has a very small communication overhead since prior knowledge about neighbor existence is not required. Each node selects a random time out and listens to messages sent by other nodes before the time out 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 an *activity* message announcing it. There are four variants in our approach, depending on whether or not *withdrawal* and *retreat* messages are transmitted. Covered nodes decide to sleep, with or without transmitting a withdrawal message to inform neighbors about the status. After hearing from more neighbors, active sensors may observe that they became covered and may decide to alter their original decision and transmit a *retreat* message. Our simulations show a largely reduced message overhead while preserving coverage quality for the ideal MAC/physical layer. Compared to an existing method (based on hello messages followed by retreat ones and where excessive message loss contributed to excessive coverage holes), our approach has shown robustness in a model with collisions and/or a realistic physical layer.

Index Terms—Sensor networks, area coverage, network connectivity, localized algorithms.

✦

1 INTRODUCTION

Once deployed in a field, the battery power of sensors cannot be easily replaced or refilled. Energy is therefore the system's most important resource. Most often, too many nodes are deployed, and only some of them are really needed for monitoring. This redundancy conveniently allows nodes that are not required for the local monitoring task to turn into sleep mode, in order to increase their own life span and the lifetime of the created network. The problem considered in this article is about sensors making decisions whether or not to turn off so that the whole area remains fully covered and the subset of active nodes remains connected. Therefore, the sensor area coverage problem is to determine a small number of active and connected sensors that still cover the same area as the fully deployed set. This enables active sensors to detect any event in the covered area and report it to a monitoring center.

Several centralized and distributed approaches have already been proposed in literature [8], [3]. 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 a decision. A number of distributed protocols relax this full information propagation but use instead a wave type of computation and communication, memorization at nodes, or unbounded delays or have other problems. Localized solutions have significantly lower communication overhead since no global view of the network is required. In a localized protocol, each node makes its activity status decision solely based on the decisions made by its communication neighbors. Moreover, in a fully localized protocol, decisions are not impacted by distant nodes (for example, 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. These solutions are suitable for networks of any size and density. They are needed for dynamic networks such as sensor networks because of changes in activity status or changes due to failures or adding more nodes. In localized solutions, topological changes are limited to the neighborhood of a node.

Our proposed solutions rely on an extremely low communication overhead in order to be suitable also for highly dense networks. No neighbor discovery is needed. Nodes wait for a random time-out duration while receiving decision messages from neighbors. The sensor evaluates its coverage and connectivity by active neighbors and decides whether or not to be active when the time out expires. Active sensors inform neighbors, whereas decisions to sleep may or may not be announced. After making a decision to be active, nodes may hear from more active neighbors, and their sensing area may then become fully covered. Such nodes may then change their minds by sending a retreat message to their neighbors.

The goal of this article is twofold. First, for the ideal MAC and physical layers, our protocols should achieve similar performance, in terms of the ratio of active sensors

---

- *A. Gallais, J. Carle, and D. Simplot-Ryl are with the POPS Research Group, IRCICA/LIFL, CNRS UMR 8022, University of Lille 1, INRIA Futurs, France. E-mail: {antoine.gallais, jean.carle, david.simplot}@lifl.fr.*
- *I. Stojmenović is with the Department of Electronic, Electrical, and Computer Engineering, University of Birmingham, UK, and the School of Information Technology and Engineering, University of Ottawa, Ontario, K1N 6N5, Canada. E-mail: stojmenovic@storm.ca.*

Manuscript received 8 Feb. 2007; revised 26 July 2007; accepted 3 Dec. 2007; published online 17 Mar. 2008.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0045-0207.
Digital Object Identifier no. 10.1109/TMC.2007.70793.

in a given round being the best existing localized solution [18], [12], while reducing significantly the number of messages for making a decision at each node. Our protocols should have comparable area coverage and connectivity preservation along with increasing reliability of the network and a decreasing message cost. More messages lead to more collisions and, consequently, more retransmissions. Second, when collision considerations at medium access or realistic physical layers are added, each message has a certain probability to be received, thus requiring several retransmissions (possibly even acknowledgments). Thus, the actual communication overhead increases more than linearly with respect to node density if the goal is to assure that decisions by neighbors are received. Further, missed messages may cause coverage holes and/or an unnecessary activity status. The goal of our protocols is winning performance simultaneously in all measures: coverage preservation, message overhead, and activity ratio.

After introducing (in Section 2) the notations that will be used in this article, we will describe, in Section 3, the existing localized area coverage algorithms, the coverage evaluation mechanisms, and the details about connectivity preservation. Our protocol will be precisely described in Section 4, and its performance will be compared to its closest competitor, the algorithm TGJD by Tian and Georganas [18], as modified by Jiang and Dou [12] and extended here for fair comparison, in Section 5.

A preliminary conference version of this article appeared in [7].

2 ASSUMPTIONS

We assume nodes to be randomly and densely deployed to create a large-scale wireless sensor network. We consider only sensor networks with static sensors. Most existing algorithms assume that sensor nodes know their respective positions. The same assumption will be made in this paper. Not only the hardware needed to receive signals from satellites, for the purpose of position determination, has become very small (for example hardware of size $7\ \mathrm{mm} \times 7\ \mathrm{mm} \times 2\ \mathrm{mm}$ exists), but the sensor positioning problem has been largely addressed in the literature as well (see [1]). In several solutions, few anchor nodes, including sinks, are sufficient to spread the location information over the whole network. Sensors are distinguished by their position and otherwise have no identities.

We assume that devices are *time synchronized* so that activity decisions can occur in rounds. Synchronization can be achieved by applying some network protocols (see [15] for a survey) or by sending a training signal from the base station or another entity (for example, helicopter) that reaches all sensors. Note that the popular ZigBee (IEEE 802.15.4) medium access protocol, which is now becoming the de facto industry standard, does assume all sensors to be time synchronized. Our time synchronization assumption is therefore justified by the latest sensor developments.

Each sensor has its own sensing radius SR and its own communication radius CR so that $SR \leq CR$, in accordance with the characteristics of existing sensors in applications. The sensor network is therefore *heterogeneous*. The connectivity and coverage criteria are defined with respect to such modeling, with sharp transition from guaranteed sensing or communication inside the circle of the corresponding radius SR or CR to the inability to do so outside these circles. The communication graphs may be asymmetric (a graph can be restricted to symmetric links if acknowledgments are part of the protocols). The modeling can however be modified. For instance, in our experiments, we considered also message collisions and the realistic physical lognormal shadowing model for communication. We address the area coverage problem in sensor networks with the requirement of maintaining both connectivity and full area coverage. The connectivity of active sensors is needed to enable reporting the acquired information to sinks.

Although our protocols work for heterogeneous sensor networks, we frequently, in our analysis and simulation, refer to the important special case of *homogeneous* sensor networks, where all sensors have an equal sensing range SR, and also, all sensors have an equal transmission range CR. This special case is called the unit disk graph (UDG) model (for sensing or communication). We then analyze further the criteria and performance for homogeneous sensor networks for an arbitrary $\frac{CR}{SR} \geq 1$ ratio. In particular, the connectivity follows from area coverage whenever $2SR \leq CR$.

The (communication) degree of a node is the number of neighbors it has within its transmission range. The average value of it (here, it is increased by one to account for the node itself) is referred to as the network density. Similarly, the sensing degree of a node is the average number of neighboring sensors whose sensing areas overlap with its own sensing area.

For simplicity, the discussion in this paper normally assumes homogeneous sensor networks. However, equivalent statements can be made also for heterogeneous networks.

A sensor is aware of the partial coverage of its sensing area by another sensor only if they are communication neighbors (at distance $\leq CR$). For example, in case $CR < 2SR$, it is possible that the sensing area of two nodes overlap, but they are not aware of that when the distance between them is $< 2SR$ but $> CR$.

Existing protocols normally assume that the communicating range is at least twice the sensing range (for example, [22]), and therefore, the connectivity is ensured for sensors with partially overlapping sensing areas. If $CR < 2SR$, then a simple connectivity test is added in the coverage evaluation process. The knowledge of the positions and transmission ranges of neighbors is sufficient to learn their connectivity graph, and Dijkstra's shortest path algorithm can be applied to test their connectivity.

In summary, we have two relevant ratios, which are $1 \leq \frac{CR}{SR} < 2$ and $\frac{CR}{SR} \geq 2$. They only differ in that the connectivity criterion is mandatory in the former case but not in the latter. Heterogeneity may introduce a variety of new cases, and then, it may be simpler to always run this test to guarantee that the active node set is connected.

3 RELATED WORK

A comprehensive literature review of existing solutions for the sensor area coverage problem, including centralized, distributed, and localized solutions, is described in [16]. Here, we describe several of the algorithms that are closest in their assumptions, especially on a localized approach. For instance, some interesting results such as those provided in [3] are not detailed here, as a global knowledge is required from each node in order to ensure the Voronoi neighbor computation. Concerning localized algorithms, several asynchronous solutions have already been proposed in [21], [4], and [9]. A more complete description of these protocols can be read in [7].

3.1 Hexagonal and Square Grid Coverage

Zhang and Hou [22] described an efficient algorithm for selecting covering sensors in a 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 the 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 the full coverage of the originally covered area.

The algorithm presented in [20] 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 the 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 a cooperative manner is nontrivial.

3.2 Coordinated Area Coverage

Hsin and Liu [10] investigated random and coordinated area coverage algorithms. Each sensor covers a circle of radius R. 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 another 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. The authors suggest that nodes collect information about residual energy from neighboring sensors. Sensors with low residual energy are more likely to enter the sleep state than sensors with high residual energy. Each sensor maintains its own delay counter, which is used for role alteration. Coordinated schemes performed better in their experiments. Although the coordinated scheme of Hsin and Liu [10] 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.

Sheu et al. [17] propose the following protocol. First, each sensor A sends or routes its priority to all sensing neighbors. Then, it considers the perimeter of its sensing circle and portions of perimeters of sensing neighbors with higher priority that are inside its own sensing circle. If all these perimeters are fully covered by other sensing neighbors with higher priorities, then A may sleep. To decide about some neighboring active sensors, each of the considered perimeters is subdivided into segments, based on the intersections with other considered circles. For each such segment, the sensor with the highest priority, among nodes covering this segment, is active. Note that some neighboring active sensors may not be discovered. However, those discovered suffice to construct a connected query tree for reporting from A to the sink. This elegant localized protocol requires one message per node for $CR > 2SR$, but for other ratios CR/SR, the routing overhead and complexity (for example, if greedy routing fails) may become excessive.

Carle et al. [2] proposed a localized scheme based on a relay selection phase. Every node selects a set of relays among its one-hop neighbors. The relays cover an area as large as the area covered by the whole neighborhood. Then, an activity decision is made based on a unique key. Any node that has the smallest priority in its neighborhood or that has been selected as a 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 one-hop neighbors and sending messages informing neighbors about relays (the latter messages are even of extended size), which is a considerably higher communication overhead than that of the methods studied and proposed in this article.

3.3 Low-Communication-Overhead Schemes

Tian and Georganas [18] proposed a solution for sensor area coverage in synchronous homogeneous networks where the sensing range is equal to the transmission range. 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 sees that neighbors (that have not yet sent a *retreat* message) together cover its monitoring area, the node transmits a *retreat* message to all its neighbors and moves into the sleep mode. Otherwise, the node remains active but does not transmit any message. The process repeats periodically to allow for changes in the 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 [12]. 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 [12] describe several improvements to the algorithm in [18]. They assume that $CR \geq 2SR$ and apply the criterion that a circle C is covered completely if the perimeters of the 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 [12], at the beginning of each round, each node sends a hello message to inform about its position. The algorithm in [18] is then applied (which relies on node retreat messages).

This algorithm is the closest competitor to our new protocols. For fairness, we modify it in several ways. First, the perimeter coverage criterion was replaced here by a computationally more efficient criterion about covering intersection points of two circles inside a given circle, described here in Section 3.4. This does not change any activity decision. Next, we consider the protocol for general ratio CR/SR by adding a similar connectivity criterion when $CR < 2SR$. The experimental data in [12] show that this algorithm outperforms PEAS [21] with respect to the number of nodes needed in the coverage while completely preserving the sensing coverage of the original network. This modified protocol will be referred to as the TGJD protocol.

3.4 Coverage Evaluation

In the rest of the article, we will assume that each node is able to evaluate the coverage provided by its communication neighbors. Several mechanisms have already been used in protocols such as TGJD. Each node can decide to enter the passive mode if its sensing area is fully covered by the set of neighbors. Let us now 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 [18] cannot be applied for heterogeneous networks and/or when $CR > SR$. We decided to apply a well-known geometric theorem, which is generally applicable. 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 [12], [22], and [19]. It efficiently confirms whether or not a sensing region is fully covered by other sensing regions. It is applied on the borders of the sensing areas of each sensor. In our case, these borders are normally circles, and we will express the theorem first in circle terminology. This criterion, expressed in the following theorem, is fast to compute and works for any CR/SR ratio and for heterogeneous networks.

Theorem 3.1. *If there are at least two covering circles and any intersection point of the 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, the intersection points of any other disk $d1$ with d must also be fully covered (be inside) a third circle $d2$. Note that circles are not required to have the same radii. Note that a complete proof was given in [7]. This theorem was also used in the protocol given in [19].

This criterion can be modified to avoid the problem of border node effects. As elaborated in [18] and [2], the nodes located near the borders of the monitored area may be active in each round since they are the only ones able to monitor certain parts of their own sensing coverage, which could be outside the area to be monitored. We therefore intersect the original joint sensing coverage and the monitored area. The monitored area is a geometrical figure such as a rectangle, inside which sensors are deployed. To remedy the problem, we assume that nodes are aware of the field they have to monitor and can adjust the covering criterion in order to consider the portions of sensing regions located inside the deployment area only (see also [22]). Nodes simply find the intersections of sensing and monitoring areas and revise their sensing areas (see [7] for an illustration).

4 NEW PROTOCOLS

Our approach is fully localized and can be applied in networks composed of time-synchronized devices knowing their positions. The main goal of our protocols is to have a very low communication overhead with a low number of active nodes in each round, thus increasing the network lifetime. Indeed, no neighbor discovery phase is needed here. Neighbor knowledge is brought by activity messages. Nodes wait and listen to activity messages to see, once their time outs 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 that they are not really needed after some new neighbors were discovered. This method permits us to reduce the number of active nodes with a small number of added messages.

4.1 Delaying Decision

Our protocols are based on a time-out scheme. When a round starts, every node selects a time out and evaluates its coverage once its time out expires. While waiting for time out to expire, nodes receive decisions made by neighbors with shorter time outs. Note that the later a node decides, the higher the probability for it to become fully covered.

The positions and decisions of these neighbors are memorized and used at the end of the time out to evaluate the coverage in order to make an appropriate decision. In dense networks, this may result in accumulating a number of decisions. To address this issue, nodes may evaluate the coverage upon receiving a certain number of messages. 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 the sleep mode may be made before the time out expires and simply wait for the time to announce it, without the need to memorize the remaining received messages.

The time-out duration can be determined at the beginning of a round and possibly revised by reacting to messages received by neighbors in various ways. We describe here three specific ways (denoted as RT, ERT, and AAT, respectively, standing for Random Time out,

Extendable Random Time out, and Activity-Aware Time out), which will be compared and evaluated in our experiments (Section 5.1).

The simplest way for deciding the time-out duration is to select a random number in a fixed interval [0, TOW], where TOW is the integer number of slots denoting the maximum possible duration for any node (assumed to be fixed and the same for all nodes). This method will be referred to as the RT method.

It is anticipated that the need to be active will decrease after receiving activity decisions from sensing neighbors. Such messages can be incorporated into a time-out function to modify it "on the fly" in various ways. One way is to observe the amount of sensing area overlap. Since exact calculation is time consuming for an actual implementation, we considered a simplified version where the node chooses a number of random points inside its own sensing area. It then evaluates if any of its already known neighbors is able to sense it. Nodes initially select random time outs in [0, M]. After each received activity message from its sensing neighbors, a node calculates the ratio R (thus, R is in [0, 1]) of the random points that becomes covered by them. The time-out duration is then modified to $M + R(TOW - M)$. Thus, the time out is extended after each received message. This method is referred to as method ERT. We used $M = \frac{TOW}{2}$ in our experiments, which appeared best after trying several options.

Our third method (AAT) is motivated by the expected energy balance of previously active nodes when the new round starts. In the first round, nodes select random time outs in [0, TOW], as in method RT. Nodes active in the previous round should attempt to avoid repeating this by selecting longer time-out durations. A node u that has been active during $a > 0$ consecutive rounds will pick up its time out within the last $\frac{1}{a+1}$ portion of the time-out window. The node selects $Random$ within [0, 1] and then, as time duration, selects the rounded value of

$$timeout_u = TOW - Random \times \frac{TOW}{a+1}.$$

In a symmetric manner, the longer nodes are passive, the shorter their time outs should be in order to incite them to get active. Then, a node u that has been passive during p consecutive rounds will pick up its time out within the first $\frac{1}{p+1}$ portion of the time-out window. In this case, u will compute its time out as follows:

$$timeout_u = Random \times \frac{TOW}{p+1}.$$

The AAT method is illustrated in Fig. 1. We used several approaches for the activity counter evolution. Basically, when a node changes its status, it can reset every counter or not. We opted for a variant with an overall counter, which is incremented when the node is active and decremented when the node decides to be passive. Therefore, in our implementation, we actually had one variable, whose absolute value was used as a value for a or p, according to its sign.

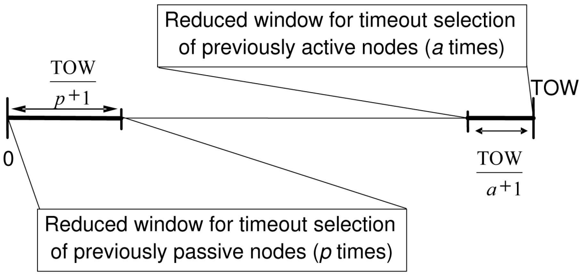


Fig. 1. Past-activity-aware time-out selection.

4.2 Decision Announcement

After verifying the coverage condition, each node decides whether or not to send a message. We propose four new protocols, which may use three types of messages. In all four proposed variants, at the end of the time out, if a node decides to be active, it sends an activity acknowledgment so that neighboring nodes with higher time-out values can consider it for their coverage evaluation. We consider variants with or without sending *withdrawal* messages. Both activity and withdrawal messages bring the same information on coverage. That is, withdrawal messages inform nodes that the area they normally cover is already covered by other sensors and may be eliminated from their coverage areas.

We found out that there was significant redundancy among the set of active nodes, since decisions are made only once during a round. We therefore introduce the capability for a node to change its decision. This must be announced by a message. Such message will be called a *retreat* message. We propose the following four new protocols:

- **Activity only** (AO). Only nodes that decide to be active send a (exactly one) message.
- **Activity and withdrawal** (AW). Every node sends exactly one message, which corresponds to its decision, an activity acknowledgment for an active status or a withdrawal before entering a sleep mode. Note that for $CR > 2SR$, withdrawal messages do not help if ideal communication is assumed since active neighboring sensors that cover the area are already able to inform the sensor directly (by activity messages) about their sensing overlap. They could still help, however, in case of message losses from active sensors.
- **Activity and retreat** (AR). Nodes that decide to be active send their decisions, whereas nodes that decide to sleep do not. Active nodes, however, can later on learn that they are covered with the help of newly announced active nodes and may decide to "change their minds" and to enter the sleep mode; such nodes send also one retreat message.
- **Activity, withdrawal, and retreat,** (AWR). All decisions by all nodes are transmitted; thus, each node sends one message corresponding to the original decision on the active or sleep status. Nodes with an originally active decision may reconsider it later on and switch to the sleep mode and possibly send one retreat message. Note that a node u cannot

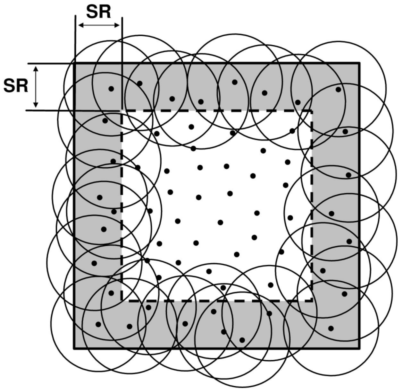


Fig. 2. Nodes within the gray strip have infinite energy.

consider (for the purpose of retreating) withdrawal messages from neighbors with originally longer time outs since these neighbors may have decided to be passive thanks to u itself.

Each message contains the position of a given node and its decision, to be active or to sleep. Thus, the decision requires only 1 bit in the message (except in the AO variant when no bit is needed). If a node sends a retreat message, 1 bit still suffices, since the message can be simply sent the same way as a withdrawal announcement, and neighboring nodes will alter the status of the neighbor that sent another message from the same position. For simplicity, we assumed initially that messages are transmitted after randomly applied delay durations without simultaneous message transmissions and message collisions. In reality, time-synchronized sensors apply the ZigBee protocol, where although they are all awake with communication devices, they apply an IEEE 802.11-like medium access protocol, with delays taken as random integers to decide how many slots without sensed transmissions in the neighborhood are to wait before its own transmission. This can obviously lead to collisions and delays in announcing decisions. Note that these delays grow more than linearly with the number of messages attempted for transmission. This is yet another reason to make activity decisions with a low communication overhead.

5 PERFORMANCE EVALUATION

We have simulated only homogeneous sensor networks. Experimental results were obtained from randomly generated connected networks with a discrete-event simulator. Nodes are deployed over a 50 × 50 rectangle area. The communication range (CR) is fixed at 10, whereas the sensing radius (SR) varies to observe the compared algorithms under two conditions: $SR \leq CR < 2SR\ (SR = 10)$ and $2SR \leq CR\ (SR = 4)$. Simulations were launched over densities varying from 40 to 90, with a step of 10. At density 40, the simulated networks are composed of 380 nodes, whereas there are roughly 860 nodes at density 90. For each density, the number of performed iterations is adjusted so that 95 percent of the results are in a sufficiently tight confidence interval. Each iteration consists of rounds. Nodes compute time-outs at the beginning of each round. Then, a round starts, and every node decides about its activity status as per the corresponding protocol. Comparison was made with the protocol called TGJD, which is our modification of the protocols proposed in [18] and [12]. We modified the protocol to provide fair comparison with our new protocols AO, AW, AR, and AWR, making all protocols the same (regarding connectivity and coverage evaluations) except for the main difference in the messages being sent. We measure the percentage of active nodes, average number of messages per node, and percentage of preserved coverage of the original area. We also measure the network lifetime defined as the number of rounds during which every node has still some power left.

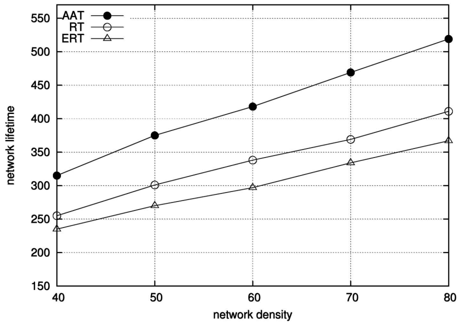


Fig. 3. Benefiting from activity-aware time outs.

5.1 The Impact of Time-Out Duration

We first study the impact of the time-out function on protocol efficiency. We define the network lifetime as the number of rounds during which every node has still some power left. In order to test the impact of particular time-out functions, we used our variant AO as a reference protocol and run for options RT, ERT, and AAT described earlier.

We used a simplistic energy consumption model, solely based on the nodes' activity. Each sensor initially has an energy of 100 battery points and remains silent and passive after reaching zero point. An active node looses one battery point, whereas passive sensors remain with an unchanged energy level. Activity rounds are simulated, and the remaining battery power of the active sensors is decremented after each round.

In order to really observe the impact of the time-out function, we had to modify our network conception. Indeed, when keeping the model of sensor nodes deployed over a square area, even with an area-aware coverage evaluation scheme, nodes located near the borders are more prone to be active. We therefore considered that every node located at a distance of at most SR from one of the borders would have an infinite energy and would be the first to announce their activity (see Fig. 2).

This allows us to focus on nodes located in the real interior of the sensor network (see the black nodes inside the white centered square in Fig. 2; for the sake of clarity, their sensing areas are not drawn). The network lifetime then equals the lifetime of the set constituted by these sensors, which is more pertinent than observing the whole set of nodes.

The comparison between our three methods is given in Fig. 3. Surprisingly, the time-out function has a much lower impact on the proportion of active nodes than anticipated.

TABLE 1
Proportions of Active Nodes Induced by the AO Variant, with Various Time-Out Durations

Density	30	40	50	60	70	80
RT	33.6	27.7	23.2	20.3	17.8	16.2
ERT	31.7	26	22.2	19.5	17.4	15.9
AAT	33.7	27.4	23.2	20.3	18.3	16.3

TABLE 2
Message Loss in the TGJD Protocol

Density	Communication degree		Retreats	
($2SR \leq CR$)	Expected	Real	Sent	Received
40	40	14	5.7	4
60	60	21	16.8	9.1
80	80	28.2	30.3	13.5

We even observe that random time outs, possibly extended upon message receptions (that is, ERT method), lead to the lowest network lifetime extensions. Actually, this method reduces the number of active nodes by allowing those that are nearly covered to extend their time outs, expecting for new neighbors to show up (see Table 1). Nevertheless, the energy consumption is not well balanced over the set of nodes, as the network lifetimes are lower than those with the RT function.

This is the reason why we considered an activity-aware time-out function whose main benefit is the load balancing it induces. Table 1 shows that as many active nodes as with RT are required, whereas the network lifetime is extended by roughly 25 percent (see Fig. 3). Results could be applied easily to any other protocol. In the rest of simulations, time-out duration RT was applied to all protocols for fairness and simplicity.

5.2 Message Overhead

We now discuss the communication overhead induced by each proposed solution. The neighbor discovery phase used in TGJD [18], [12] induces one message per node. Then, activity messages need to be compared. Fig. 4 shows the average total number of messages used by a node during one round when $2SR \leq CR$. Protocol names are given in Section 5.4. These results are in accordance with the number of active nodes. We can observe that an excessive number of messages are needed by protocol TGJD since it relies on a retreating phase. The more the nodes that decide to sleep, the more messages are sent. Intuitively, the higher the density, the more the passive nodes are, since the number of required active sensors does not vary much. Therefore, we observe the increase of messages sent by TGJD along with the density (up to 1.8 at density 70, whereas AO is at 0.3). Note that our approaches have a very low communication overhead; the higher the density, the lower the number of messages per node, thanks to both the absence of neighbor discovery and the nonsystematic transmissions of withdrawal and retreat messages.

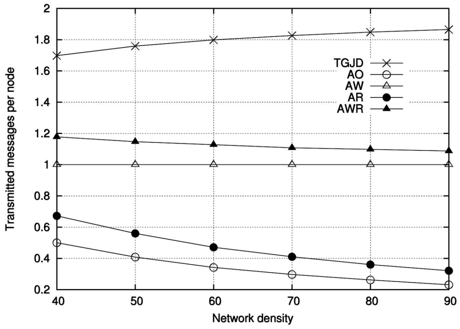


Fig. 4. Very low communication overhead of AO, AW, AR, and AWR compared to TGJD ($2SR \leq CR$).

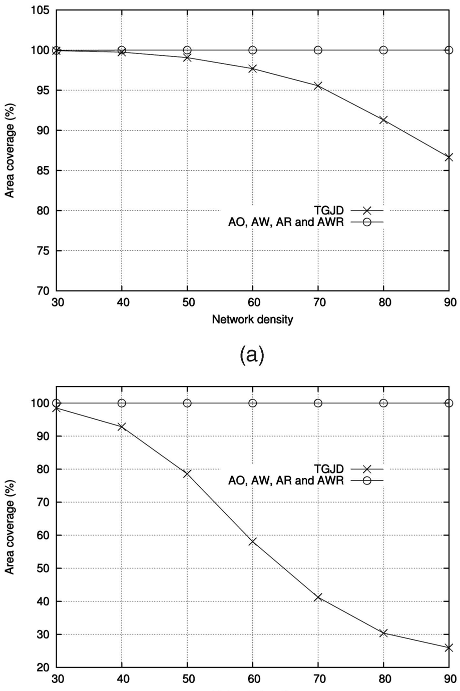
We have also simulated a MAC layer where collisions are modeled. We have added a contention window of size CW and a time out for each node before it can send any message. Any node randomly picks up an integer value between zero and CW. A node can neither receive two messages correctly nor receive a message while transmitting during the same time slot. We selected a contention window whose size is equal to the network density, which allows us to have a reasonable amount of potential collisions. This simple model of a more realistic MAC layer allows us to measure the impact of collisions and message failures on the studied protocols.

Collisions were considered on both neighbor discovery and retreat messages of the TGJD protocol. Table 2 shows the impact of using a contention window with the 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 (collisions occurred). During the retreating phase, we have counted the number of retreat messages sent to a given node (sent retreats) and the number of retreat messages this node has effectively received (received retreats). The loss of messages is in accordance with the size of the contention window we have used.

For instance, at density 40, less than half of the sent *hello* messages are received. We have counted every sent message so that the nodes that have not been discovered during the *hello* phase can send a retreat 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 nodes have more chances to turn off with increased density, the number of transmitted retreat messages increases (from 5.7 at density 40 to 30.3 at density 80). Furthermore, the proportion of received retreat messages decreases, and it has a negative impact on the performance of the TGJD protocol. Therefore, we conclude that an approach with fewer messages should be preferred in environments with considerable loss of messages.

5.3 Area Coverage

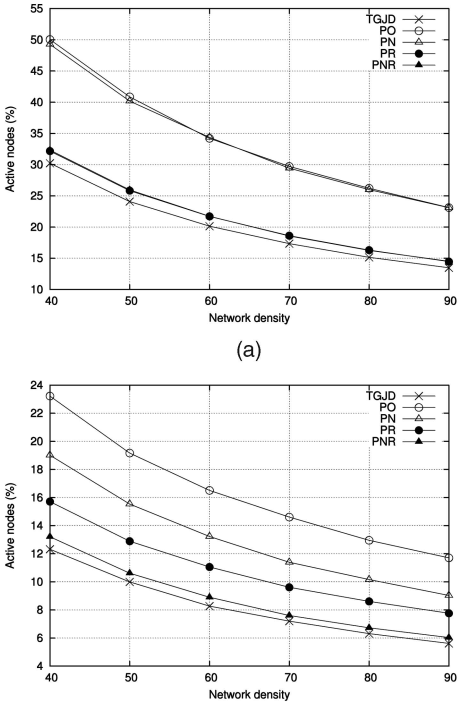
We have measured how much coverage the considered algorithms were able to provide with respect to the deployment area. Due to the relatively high densities considered, sensors are able to cover fully (100 percent) the deployment area. The results presented here refer to this coverage, which allows border 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.



(a)

(b)

Fig. 5. Area coverage with message loss. (a) $2SR \leq CR$. (b) $SR = CR$.



(a)

(b)

Fig. 6. Active nodes without message loss. (a) $2SR \leq CR$. (b) $SR = CR$.

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 a distance of 1/10 of the sensing range. The reported percentages refer to the percentage of these grid points being covered. When every sent message is received (in the case of the ideal MAC layer), all considered schemes are efficient since the deployment area remains fully covered.

We then evaluated the impact of message losses on studied protocols. Fig. 5 shows the area coverage versus the network density. When $SR = CR$ and the TGJD protocol respectively provides 92.8 percent, 58.1 percent, and as low as 30.4 percent of area coverage at densities 40, 60, and 80, all our variants maintain full coverage.

AO and AW protocols preserve full coverage and support to the monitoring application since they rely on correct information. The worst case is that no message is ever received, leading to empty neighbor tables. This would impose every node to be active. In TGJD, if no retreat message is ever received, tables remain as they were constructed during neighbor discovery, whereas some nodes might have decided to retreat. This is why TGJD suffers so much from the introduction of message loss in the communication model.

Despite the use of retreat messages, our variants AR and AWR can preserve full coverage. Indeed, the number of retreat messages is very low compared to those required by TGJD (when eight nodes out of 10 decide to be passive, as many nodes have to send a retreat message). In our protocol, retreat messages can be considered as update messages whose potential loss has far less impact than in protocol TGJD.

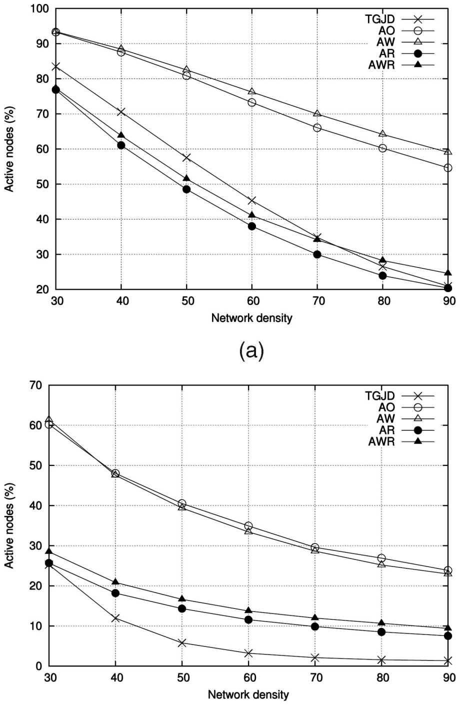
These results demonstrate the robustness of our variants compared to protocol TGJD when the loss of messages is considered. Indeed, in protocol TGJD, if a retreat message is not correctly received, it means that some nodes may decide to sleep while leaving some uncovered areas. Therefore, every protocol using hello and retreat messages is expected to be unable to provide full coverage. Instead of being beneficial, their application appears wasted and counterproductive.

5.4 Active Nodes

It is important to have as few active nodes as possible while not inducing much communication overhead. Fig. 6 shows the average percentage of active nodes induced by TGJD and our four variants when the ideal MAC layer is used.

For $2SR \leq CR$, we can observe in Fig. 6a that at density 60, TGJD has 20.1 percent of active nodes, followed by AWR and AR with 21.7 percent and AO and AW with 34.3 percent. Note that withdrawal messages do not decrease the number of active nodes. In fact, when $2SR \leq CR$, neighbors with partial sensing area overlap are communication neighbors, and their activity messages are received. This means that withdrawal messages are then wasted. However, they help to reduce the percentage of active nodes when $CR < 2SR$. For instance, once $SR = CR$ (see Fig. 6b), we can notice that at density 50, TGJD has 10 percent of active nodes, followed by AWR with 10.6 percent, AR with 12.9 percent, AW with 15.5 percent, and AO with 19.2 percent.

Notably, retreat messages show more benefits than withdrawal acknowledgments. Also, two of our protocols are very competitive with TGJD whose low number of active nodes is due to the full neighborhood knowledge used to make decisions. Nevertheless, excessive messaging is induced to gain such knowledge as we have already shown in Section 5.2. Note that the percentages of



(a)

(b)

Fig. 7. Active nodes with message loss. (a) $2SR \leq CR$. (b) $SR = CR$.

active nodes decrease as density increases. Intuitively, fewer active nodes are needed when SR is increased.

When collisions are simulated, the number of active nodes is significantly modified. Fig. 7 shows the measured percentage of active nodes. As previously explained, the coverage provided by the TGJD protocol suffers considerably from the loss of messages. Too many nodes decide to sleep since retreat notifications are not correctly received. This leads to only 3.2 percent of active nodes when the density is 60 and $SR = CR$ (see Fig. 7b). Our variants behave differently. Indeed, as activity notification messages get lost, more nodes decide to be active, believing that they are needed for coverage. When $2SR \leq CR$ (Fig. 7a) and when the density is equal to 60, this leads to nearly 73.2 percent of active nodes for AO, whereas it generated less than the half (34.5 percent) under ideal conditions. This can also be observed on AW. AR and AWR help this increased redundancy to be reduced thanks to the retreat process. When $SR = CR$, they respectively generate 9.8 percent and 12 percent active nodes at density 70 when messages collide, which is a small increase compared to the results observed under ideal conditions (respectively, 9.6 percent and 7.6 percent for AR and AWR). This shows that these variants produce small excess of active nodes rather than large coverage holes as in the case of protocol TGJD.

5.5 Connectivity Preservation

Our protocols aim at preserving the connectivity of the set of active nodes by enhancing the decision rule. Any node can decide to be passive provided that its sensing area is fully covered by a connected set of neighbors. Otherwise, it should remain active in order to still guarantee the multihop communication among its disconnected neighbors. We have observed as to what extent the studied protocols were able to preserve connectivity. We computed the percentage of connected active node topologies.

TABLE 3
Percentage of Connected Active Node Sets
with Message Loss ($SR = CR$)

	$Density = 40$	$Density = 60$	$Density = 80$
TGJD	0	0	0
AO	100	100	100
AW	100	100	100
AR	98	97	99
AWR	92	88	88

If we consider ideal conditions where no message is ever lost, TGJD and our four variants perfectly preserve the connectivity of the set of active nodes. Indeed, as soon as a node has a disconnected set of neighbors, it decides to remain active.

We have looked at how well the connectivity was preserved if messages could collide and thus prevent nodes from gathering all the information concerning their neighborhood. As already observed for area coverage, connectivity likewise suffers from the retreat process of TGJD. Indeed, a node u may decide to retreat thanks to a connected subset of covering neighbors, but this set may actually contain nodes that have already decided to retreat. Nevertheless, corresponding retreat messages were not received due to some collisions. Our experiments indicate that protocol TGJD never builds a connected active node set when collisions are modeled (see Table 3).

Our variants AO and AW do not suffer from message losses since they still preserve the connectivity of the active node sets. This is due to the reduced information that activity decisions rely on. Lost messages only increase the number of active nodes, and every topology remains connected. Table 3 shows that these two variants always build connected active node sets.

However, when we simulated retreat-based variants (AR and AWR), we suffer from the same problems as protocol TGJD. Retreat messages might get lost, thus leading to wrong information and wrong decisions. This is why AR and AWR fail to maintain all topologies connected when messages could collide. Still, as the retreat process is simply an update here, used by few nodes, protocols AR and AWR still preserve connectivity in nearly 90 percent of connected topologies (see Table 3), compared to none for protocol TGJD.

5.6 Impact of a Realistic Physical Layer

All results have been obtained from simulated graphs, each being denoted as $G(V, E)$. The set of sensor nodes is V, whereas the set of edges (E) is obtained from the communication model, which was the UDG. This model induces many assumptions on radio links that are not realistic. Indeed, due to the nature of radio waves, the signal strength greatly decreases with the distance. We selected the *lognormal shadowing* model, described by Quin and Kunz in [14], and transformed our UDGs into weighted graphs. The weight of each edge $(u, v) \in E$ is equal to the probability of correct reception $\mathrm{p}(\mathrm{dist}(u, v))$ for the two nodes u and v. In our simulations, we used an

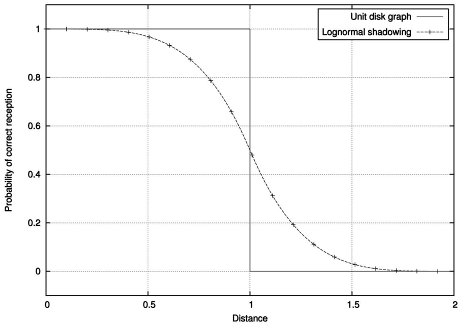


Fig. 8. The two considered physical models ($CR = 1$ and $\alpha = 2$).

approximated function $\mathrm{P}(x)$, described by Kuruvila et al. in [13], as follows:

$$\mathbf{P}(x) = \begin{cases} 1 - \frac{(\frac{x}{CR})^{2\alpha}}{2} & \text{if } 0 < x \le CR, \\ \frac{(\frac{2\times CR - x}{CR})^{2\alpha}}{2} & \text{if } CR < x \le 2 \times CR, \\ 0 & \text{otherwise.} \end{cases} \qquad (1)$$

In this formula, α is the power attenuation factor, x is the considered distance, and CR is the theoretical communication range of a node. This function assumes that the probability of correct reception for the range CR is always equal to $\mathrm{P}(CR) = 0.5$. Fig. 8 illustrates this function for $\alpha = 2$ [13].

We first combined this model with the ideal MAC layer. The obtained data did not differ (in their relative comparison) significantly from the case of MAC with collisions and UDG. This is because message losses occur similarly due to probabilistic loss or collisions. We then evaluated the protocols under both realistic physical and MAC layer assumptions. We have already conducted such experiments with an extended version of the AO variant [6]. We have also tested the three others, AW, AR, and AWR, along with TGJD. We only observed the impact on area coverage and on the number of active nodes.

As shown in Fig. 9, the coverage provided by TGJD is nearly complete only for density 40 and then decreases for higher ones for $2SR \le CR$. The retreat message of a node u may not be received by a node v, which had first received a *hello* message from u. After the retreating phase, nodes using TGJD have wrong neighbor tables. Then, as shown in Fig. 10, many nodes decide to become passive, and the global coverage of the network thus decreases. This phenomenon is less visible in sparser networks, because the probability that a node finds a covering set of neighbors is lower. The decision to become passive is thus taken less frequently. Since these results are obtained for $2SR \le CR$, the probability that two nodes whose sensing areas overall communicate is > 0.5. In case $SR = CR$, the performance of TGJD are strongly negatively impacted.

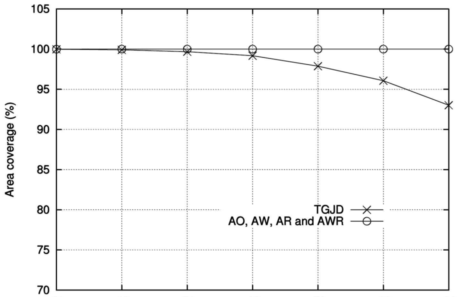


Fig. 9. Area coverage maintenance with realistic physical and MAC layers ($2SR \le CR$ and $\alpha = 2$).

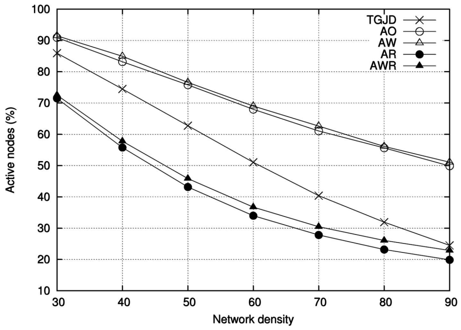


Fig. 10. Both AR and AWR outperform TGJD with realistic physical and MAC layers ($2SR \le CR$ and $\alpha = 2$).

Our AO variant outperforms TGJD but still generates a high percentage of active nodes (Fig. 10). An important fact is that for $2SR \le CR$, protocol AW generates a similar number of active nodes as protocol AO. The gap between AO and AW enlarges with increased density. More collisions occur, and some of the withdrawal messages help nodes to learn about active neighbors whose activity message was lost. Nevertheless, the use of withdrawal messages could be counterproductive as they also participate in raising the communication overhead, thus leading to more collisions. Finally, variants using retreats (AR and AWR) help reduce the percentage of active nodes while preserving the coverage fully. AR should therefore be preferred as it induces a lower communication overhead.

6 CONCLUSION

Experimental results with the ideal MAC layer show that for a similar number of selected active sensors, our methods significantly reduce the number of messages to decide the status compared to an existing localized protocol where nodes send *hello* message followed by retreat messages from nodes before sleeping. We also consider message losses, induced by a MAC layer with collisions and/or a realistic physical layer, and show that the existing compared method, for dense networks, fails to cover the area reasonably with a connected set of active nodes (nodes may decide to sleep since some *retreat* messages are not received, creating coverage *holes*, and connectivity losses). Our methods, however, still remain robust in terms of high area coverage with a reasonable amount of active nodes and connectivity preservation despite message losses.

7 FUTURE WORK

Various improvements to the presented protocols are possible. We have shown here that activity-aware time-out functions could help extend the network lifetime. Improved versions of the time-out functions may also be considered in order to optimize the number of active nodes. The time out could therefore depend on a measure that corresponds to the size of the uncovered portion of a node's monitoring area. Nevertheless, we identified several problems with such computation. First, the exact computation involves computing the surface area of the uncovered portion of the monitoring area. To find this net area, the boundary of it is first found using a polynomial-time algorithm [11]. This algorithm finds, for each neighbor, the portion of its perimeter, located inside a given circle, that is not covered by other sensors. These portions then create the boundary of the region. In the second step [4], the area of each net sensing region can be computed by calculating the area of the polygon formed by its segment sequence. The algorithm has sophisticated details. It is possible to estimate the size of the intersection area in several ways. One simple way is to consider a dense grid of points in the monitoring area and find out how many of them are not covered, using their ratio in time-out function. The problem with this approach is that the computation itself is time consuming; more time is needed for more accuracy. Also, sensors need to memorize uncovered grid points while waiting for more messages from neighbors so that the time out is related properly. Sensors may not have enough memory and time available for performing these computations. When $SR = CR$, the computation can be simplified, since the perimeter of a covering circle may be used instead of the grid. The size of the uncovered portion of that perimeter is simple to compute.

We finally plan to extend our protocols by considering k-coverage rather than simple sensing coverage for the deployment area. This can be done by modifying either the coverage evaluation scheme or the protocol itself [5]. Still, both extensions are based on ideas presented here for maintaining a low communication overhead and correct behaviors under realistic assumptions. We will also consider a more realistic physical layer for sensing as we already initiated in [6].

ACKNOWLEDGMENTS

This research was partially supported by NSERC Canada, UK Royal Society Research Merit Award, and INRIA, France.

REFERENCES

[1] J. Bachrach and C. Taylor, "Localization in Sensor Networks," *Handbook of Sensor Networks,* I. Stojmenovic, ed., John Wiley & Sons, pp. 277-310, 2005.

[2] J. Carle, A. Gallais, and D. Simplot-Ryl, "Preserving Area Coverage in Wireless Sensor Networks by Using Surface Coverage Relay Dominating Sets," *Proc. 10th IEEE Symp. Computers and Comm. (ISCC),* 2005.

[3] B. Carbunar, A. Grama, J. Vitek, and O. Carbunar, "Coverage Preserving Redundancy Elimination in Sensor Networks," *Proc. First Ann. IEEE Conf. Sensor and Ad Hoc Comm. and Networks (SECON),* 2004.

[4] Y. Cai, M. Li, W. Shu, and M.-Y. Wu, "ACOS: A Precise Energy-Aware Coverage Control Protocol for Wireless Sensor Networks," *Ad Hoc & Sensor Wireless Networks,* vol. 3, no. 1, pp. 77-98, 2007.

[5] A. Gallais and J. Carle, "An Adaptive Localized Algorithm for Multiple Sensor Area Coverage," *Proc. IEEE 21st Int'l Conf. Advanced Information Networking and Applications (AINA),* 2007.

[6] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Ensuring K-Coverage in Wireless Sensor Networks under Realistic Physical Layer Assumptions," *Proc. Fifth IEEE Conf. Sensors (Sensors),* 2006.

[7] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Localized Sensor Area Coverage with Low Communication Overhead," *Proc. Fourth IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom),* 2006.

[8] H. Gupta, S. Das, and Q. Gu, "Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution," *Proc. ACM MobiHoc,* 2003.

[9] C. Gui and P. Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks," *Proc. ACM MobiCom,* 2004.

[10] C.-F. Hsin and M. Liu, "Network Coverage Using Low Duty-Cycled Sensors: Random & Coordinated Sleep Algorithms," *Proc. Third Int'l Symp. Information Processing in Sensor Networks (IPSN),* 2004.

[11] C.-F. Huang and Y.-C. Tseng, "The Coverage Problem in a Wireless Sensor Network," *Proc. Second ACM Int'l Conf. Wireless Sensor Networks and Applications (WSNA),* 2003.

[12] J. Jiang and W. Dou, "A Coverage Preserving Density Control Algorithm for Wireless Sensor Networks," *Proc. Third Int'l Conf. Ad-Hoc Networks and Wireless (ADHOC-NOW),* 2004.

[13] J. Kuruvila, A. Nayak, and I. Stojmenović, "Hop Count Optimal Position Based Packet Routing Algorithms for Ad Hoc Wireless Networks with a Realistic Physical Layer," *Proc. First IEEE Int'l Conf. Mobile Ad Hoc and Sensor Systems (MASS),* 2004.

[14] L. Quin and T. Kunz, "On-Demand Routing in Manets: The Impact of a Realistic Physical Layer Model," *Proc. Second Ad-Hoc Networks and Wireless (ADHOC-NOW),* 2003.

[15] K. Romer, P. Blum, and L. Meier, "Time Synchronization and Calibration in Wireless Sensor Networks," *Handbook of Sensor Networks,* I. Stojmenović, ed., John Wiley & Sons, pp. 199-238, 2005.

[16] D. Simplot-Ryl, I. Stojmenović, and J. Wu, "Energy-Efficient Backbone Construction, Broadcasting, and Area Coverage in Sensor Networks," *Handbook of Sensor Networks,* I. Stojmenović, ed., John Wiley & Sons, pp. 343-380, 2005.

[17] J.P. Sheu, C.H. Yu, and S.C. Tu, "A Distributed Protocol for Query Execution in Sensor Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC),* 2005.

[18] D. Tian and N. Georganas, "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks," *Proc. First ACM Int'l Conf. Wireless Sensor Networks and Applications (WSNA '02),* pp. 32-41, 2002.

[19] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks," *ACM Trans. Sensor Networks,* vol. 1, no. 1, pp. 36-72, 2005.

[20] T. Yan, T. He, and J.A. Stankovic, "Differentiated Surveillance Service for Sensor Networks," *Proc. ACM Conf. Embedded Networked Sensor Systems (SenSys),* 2003.

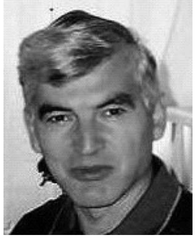
[21] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-Lived Sensor Networks," *Proc. 23rd Int'l Conf. Distributed Computing Systems (ICDCS),* 2003.

[22] H. Zhang and J.C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Ad Hoc & Sensor Wireless Networks,* vol. 1, pp. 89-123, 2005.

Antoine Gallais received the MSc and PhD degrees in computer science from the University of Lille, France, in 2004 and 2007, respectively. Recently, he joined the Fundamental Computer Science Laboratory of Lille (LIFL), as a research assistant. He is involved in the POPS research group, a common project of the University of Lille, CNRS and INRIA. His main interests include ad hoc and sensor networks, distributed computing and self-organization of large populations of wireless devices.

Jean Carle received the MSc and PhD degrees in computer science from the University of Amiens, France, in 1998 and 2000, respectively. In 2001, he joined the University of Lille, France, where he is currently an associate professor. He is involved in the POPS research group, a common project of the University of Lille, CNRS and INRIA. His research interests include distributed computing and data communication in mobile ad hoc and sensor networks. He was a program committee member of a number of international conferences and workshops such as MobiWac 2004, MSN 2005-2006, WWAN 2005, EUC 2005, WirelessCom 2005, PERCOM 2006, ICC 2006, InterSense 2006, IADIS AC 2006-2007, EEWSN 2006, LOCAN 2007, SNDP 2007, ISSNIP 2007, and PPN 2007.

David Simplot-Ryl received the PhD degree in computer science in 1997. In 1998, he joined the Fundamental Computer Science Laboratory of Lille (LIFL), France, where he is currently a professor. He is the scientific leader of the POPS research group, a common project of the University of Lille, CNRS, and INRIA and the scientific coordinator of the national CNRS project RECAP on sensor and self-organizing networks. He has been the research vice chair of INRIA Futurs Research Centre since 2007. His research interests include sensor and mobile ad hoc networks, embedded operating systems, and RFID technologies. He is an associate editor, a member of editorial board, or a guest editor of several journals (*Ad Hoc & Sensor Wireless Networks*, *International Journal of Wireless and Mobile Computing*, *IEEE Network Magazine*, and *IEEE Transactions on Parallel and Distributed Systems*). He was the general chair or program chair of IEEE MASS 2007 and ADHOC-NOW 2008 and also the chair or a cochair for several other international conferences and workshops (ICDCS 2004-2007 for WWAN and ACM MobiCom 2008 for SANET).



Ivan Stojmenović received the PhD degree in mathematics. He held positions in Serbia, Japan, USA, Canada, France, Mexico, Spain, and United Kingdom. He is currently the chair professor in applied computing at the University of Birmingham, United Kingdom. He has published more than 200 different papers and edited four books on wireless, ad hoc, and sensor networks and applied algorithms with John Wiley & Sons or the IEEE. He is currently an editor of more than a dozen journals and the founder and editor in chief of three journals (*Journal of Multiple-Valued Logic and Soft Computing*, *International Journal of Parallel, Emergent and Distributed Systems*, and *Ad Hoc & Sensor Networks*). He is in the top 0.56 percent most cited authors in computer science (Citeseer 2006). One of his articles was recognized as the Fast Breaking Paper in October 2003. He is a recipient of the Royal Society Research Merit Award, United Kingdom. He was recently elected to IEEE fellow status (class of 2008) and is a member of the IEEE Computer Society. He has chaired and/or organized more than 30 workshops and conferences, served in more than 100 program committees since 2004, and presented more than a dozen tutorials.