

# Activities scheduling algorithms based on probabilistic coverage models for wireless sensor networks

Mohammed Boudali<sup>1</sup> · Mustapha Reda Senouci<sup>1</sup> · Mohamed Aissani<sup>1</sup> · Walid-Khaled Hidouci<sup>2</sup>

Received: 31 January 2016 / Accepted: 20 January 2017 / Published online: 1 February 2017  
© Institut Mines-Télécom and Springer-Verlag France 2017

**Abstract** Area monitoring using Internet and barrier coverage is a typical application of wireless sensor networks. The main concerns in this type of applications are coverage efficiency and sensor energy conservation. For that, many activities scheduling algorithms are proposed in the literature. Unlike prior efforts based on an unrealistic binary sensor coverage model, this paper proposes three efficient activities scheduling algorithms based on realistic sensor coverage models. The first algorithm (C1L-PBC) is centralized and it is based on a coverage graph. The second algorithm (D1L-PBC) is distributed and it ensures 1-barrier coverage; whereas, the third one (D2L-PBC) is also distributed and it guarantees 2-barrier coverage. The obtained experimental results show that the proposed algorithms can effectively guarantee the barrier coverage and prolong the sensor network lifetime.

**Keywords** Wireless sensor networks · Barrier coverage · Intrusion detection · Activities scheduling algorithms · Network lifetime

---

✉ Mohammed Boudali  
mohammedboudali@gmail.com

Mustapha Reda Senouci  
mrseouci@gmail.com

Mohamed Aissani  
maissani@gmail.com

Walid-Khaled Hidouci  
wk.hidouci@gmail.com

<sup>1</sup> Ecole Militaire Polytechnique (EMP), Algiers, Algeria

<sup>2</sup> Ecole nationale Supérieure d'Informatique d'Alger (E.S.I), Algiers, Algeria

## 1 Introduction

A wireless sensor network (WSN) is composed of several sensor nodes (or sensors) that communicate and send the captured information to one or more collection points, called sinks. WSNs, formed of hundreds or thousands of sensors that are usually deployed in large-scale environments, are used in various applications, such as environmental monitoring, forest fire detection, battlefield monitoring, and target tracking [1].

Area monitoring using Internet and barrier coverage technique (Fig. 1) is one of the main applications of WSNs. Sensors are deployed along the border of the area of interest (AoI), and they collaborate together to construct a barrier detecting each intruder who tries to enter the monitored area. The end-user could send commands to the barrier (e.g., requesting barrier reconstruction), and receive alerts from the barrier (e.g., barrier failure notification). As energy is the main problem in the barrier coverage and knowing that sensors powered by batteries are not always easy to replace once exhausted, each proposed solution must consider this constraint. Therefore, the major preoccupation is the optimization of barrier coverage reliability while optimizing energy consumption.

To handle the energy problem in WSNs, several activities scheduling algorithms are proposed in the literature. In these algorithms, the nodes that ensure coverage are kept active while the rest of nodes are in sleep state. To the best of our knowledge, all these algorithms are based on the binary sensor coverage model, which does not reflect reality well. Our main objective is to protect an AoI against intrusions by using WSN barrier coverage. To maximize the lifetime of this barrier, we exploit the activities scheduling concept. More precisely, based on a realistic coverage model, we devise three energy-aware activities scheduling algorithms that build WSN barrier coverage.

The first proposed algorithm, called C1L-PBC (centralized probabilistic one-level barrier coverage), centralizes all calculations in one sink. After calculation, the latter sends messages to activate (or deactivate) the nodes. The C1L-PBC algorithm collects the information about all nodes, constructs the coverage graph, and searches the optimal path in the graph. This path is the set of nodes that will be put in the active state. The second algorithm, named D1L-PBC (distributed probabilistic one-level barrier coverage), uses two sinks to perform the necessary calculations, and each sink is placed in one extremity of the barrier. The first sink triggers the selection process of nodes that remain in the active state by broadcasting a discovery message in the network. When the second sink receives discovery messages, it chooses the nodes to keep in the active state by sending an activation message to all of the chosen nodes. The third algorithm, called D2L-PBC (distributed two-level probabilistic barrier coverage), provides barrier coverage with two levels. This distributed algorithm has two main objectives: (1) provide a double barrier coverage so that if a sensor node of the barrier fails then there will be at least one neighbor node in the active state who always ensure coverage and (2) increase the coverage probability so that every point of the barrier is covered by at least three sensors.

The rest of this paper is organized as follows. The concept of barrier coverage and the coverage models are introduced in Section 2. Section 3 discusses related work. Section 4 highlights our assumptions. Our proposals are described in Section 5, and their achieved performance are presented and discussed in Section 6. Section 7 concludes the paper and gives opportunities of this work.

## 2 Background

Barrier coverage is a coverage type used when we want to secure an AoI from an intruder (mobile object) that wants to penetrate into it [2]. The AoI in which the sensors are deployed is a belt as shown in Fig. 2.

The coverage models are abstraction models that try to quantify the performance of a sensor when capturing a physical phenomenon in a given location. These models are represented by a coverage function that has as input the distance between a space point  $z$  and the sensor. Several models were proposed in the literature; among them are the following:

- (a) The Boolean disk coverage model (binary model), which is the most used in the literature. The coverage function of this model is given by Eq. (1), where  $d(s, z)$  is the Euclidian distance between a point  $z$  and the sensor  $s$ , and  $R_s$  is the sensing radius. In this

model, the coverage quality is equal to 1 if the distance between  $s$  and  $z$  is inferior or equal to  $R_s$ .

$$P(d(s, z)) = \begin{cases} 1 & \text{If } d(s, z) \leq R_s \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

- (b) The truncated attenuated disk model, which is a probabilistic model. The coverage quality decreases while moving away from the sensor and it becomes negligible when the distance  $d(s, z)$  exceeds a certain threshold. The coverage function given by Eq. (2) represents one of the truncated attenuated models proposed in the literature, where  $R_s$  is the sensing range,  $R_u$  is called the uncertain range, and  $\alpha$  and  $\beta$  are constants. To compute the quality of coverage of  $N$  sensors in a point  $z$ , we use Eq. (3). The probabilistic coverage models are more realistic than the binary ones. Therefore, the use of probabilistic coverage models in our solutions will provide more realistic results.

$$P(d(s, z)) = \begin{cases} 1 & \text{If } d(s, z) \leq R_s - R_u \\ e^{\alpha(d(s, z) - (R_s - R_u))^\beta} & \text{If } (R_s - R_u) < d(s, z) \leq R_s \\ 0 & \text{If } R_s < d(s, z) \end{cases} \quad (2)$$

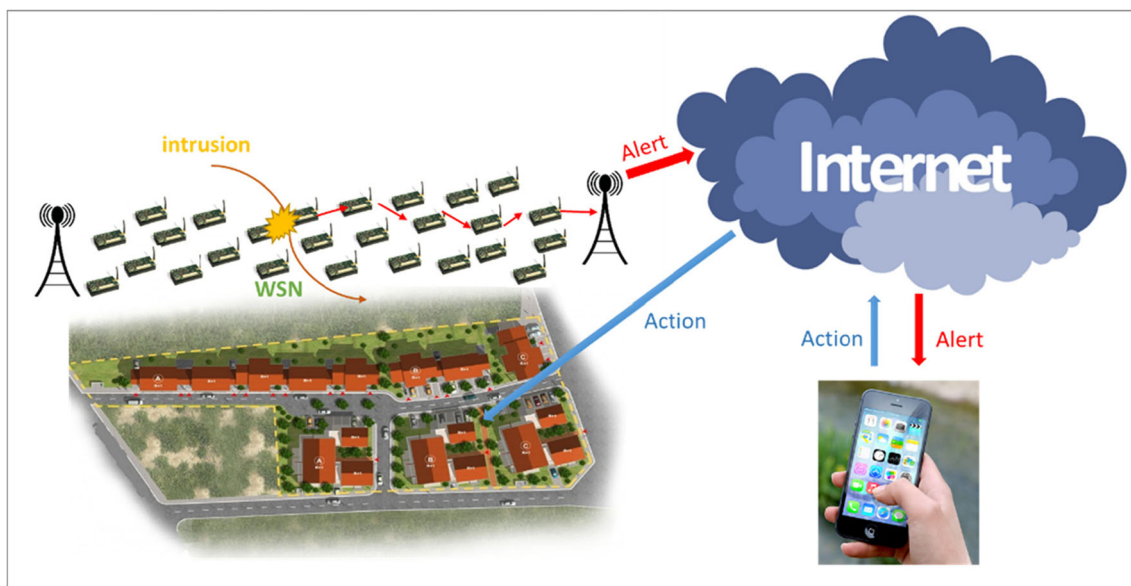
$$P_N(z) = 1 - \prod_{i=1}^N (1 - P(d(i, z))) \quad (3)$$

The coverage graph  $CG = (V, E)$  models the WSN coverage state as shown in Fig. 3. It is composed of a set of vertex  $V$  and a set of edges  $E$ , constructed as follows: the vertexes  $V$  correspond to the sensors of the WSN. An edge exists between two vertexes if and only if the areas of detection of the corresponding sensors overlap [2, 3]. When we use a probabilistic coverage model, like the attenuated truncated model, we put edge between two vertexes if and only if the corresponding sensors ( $u$  and  $v$ ) verify Condition (4), where  $d(u, v)$  is the distance between  $u$  and  $v$ ,  $D = d(u, v)/2$ , and  $P_t$  is the probability threshold. In our work, when we build the coverage graph, we add two virtual nodes  $S$  and  $D$  to indicate the beginning and the end of the barrier, as depicted in Fig. 3.

$$P_{min_{u,v}(D)} = 1 - (1 - P_u(D))(1 - P_v(D)) > P_t \quad (4)$$

## 3 Related work

To the best of our knowledge, all existing scheduling activities algorithms are based on a binary coverage model. Kumar et al. [4] have introduced the barrier coverage problem for the first



**Fig. 1** Area monitoring using Internet and barrier coverage technique

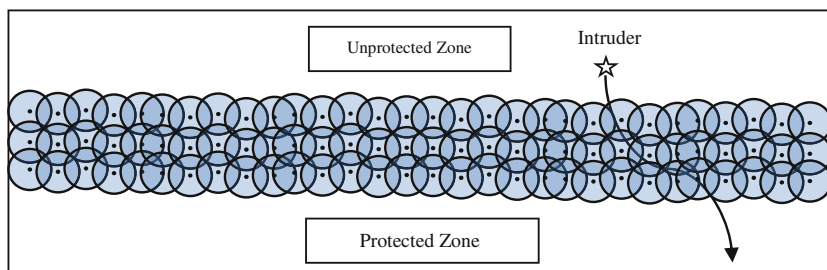
time. They have presented an algorithm to determine whether a WSN provides barrier coverage. In his work, Kumar [5] presented a centralized algorithm, named REPAIR, which consists in adding the minimum of sensors in defined places inside existing barrier coverage to ensure  $k$ -barrier coverage. Kumar et al. [4] also proposed the simplest activities scheduling algorithm, called RIS (random independent sleeping), which is based on a certain probability  $P$ , where each node activates itself during a certain time independently of the other nodes. The barrier coverage in RIS is guaranteed probabilistically. Kumar et al. [3] proposed two algorithms, named PRAHARI and STINT, to schedule the activity of sensors. STINT is dedicated to homogeneous networks, while PRAHARI is dedicated to heterogeneous networks.

Chen et al. [6] defined the local  $k$ -barrier coverage and proposed the localized  $k$ -barrier coverage construction protocol, named LBCP (localized barrier coverage protocol). In LBCP, a node uses its neighbor’s information and schedules locally its activity state. Shen et al. [7] proposed a deterministic centralized algorithm, called CBarrier, for omnidirectional mobile sensors. This algorithm calculates the possible locations to which the mobile sensors will move to form the barrier coverage. Saipulla et al. [8] proposed a centralized algorithm

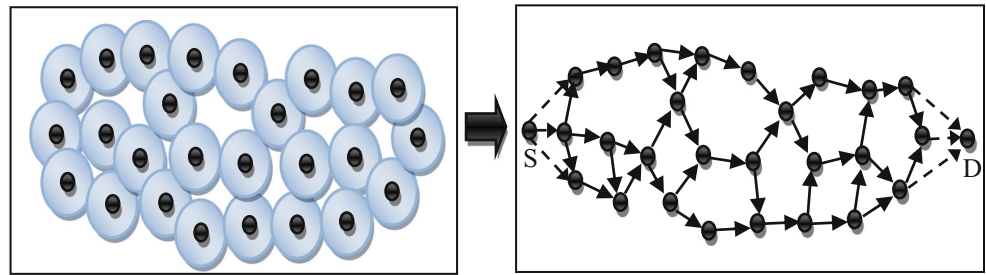
to construct the barrier coverage with sensors deployed from a plane. Bhattacharya et al. [9] proposed several algorithms to determine the best locations and movement modes of mobile sensors to ensure optimal circular barrier coverage. Ssu et al. [10] introduced the  $k$ -barrier coverage with directional sensors, deployed randomly, and then proposed a distributed construction algorithm. Ban et al. [11] proposed a distributed algorithm, called DBCS (distributed barrier coverage scheduling), where each node exchanges its information with its neighbors, and on the basis of this information, it decides if it remains active or goes to the sleep state.

Yang et al. [12] proposed a sliding barrier coverage algorithm based on the principle that the barrier can change length and direction randomly in time. The proposed algorithm subdivides a zone into subzones and ensures the  $k$ -barrier coverage in each one. Yamamoto et al. [13] proposed a distributed algorithm, called SSBC (sleep scheduling barrier coverage), where the static nodes are grouped in clusters. Each cluster ensures local barrier coverage and the coordination among clusters is done by the sink to obtain global barrier coverage. Tao et al. [14] proposed a centralized algorithm, called SBP (strong barrier path), for a strong barrier coverage based on directional sensors. Cao et al. [15] proposed a centralized

**Fig. 2** Illustration of barrier coverage



**Fig. 3** Coverage graph



algorithm, named  $k$ -SBCCS (protocol of strong  $k$ -barrier coverage with coordinated sensors) which ensures the  $k$ -barrier coverage. This algorithm subdivides a zone into subzones and ensures  $k$ -barrier coverage in each subzone.

Chen et al. proposed in [16] an approach based on the theoretical analysis of detection probability. They formulated minimum weight-barrier problem about how to schedule sensor energy efficiently, which is NP-hard. Du et al. [17] focused on maximizing the network lifetime under a novel  $k$ -discrete barrier coverage model, whose goal is to cover some specific discrete points by deploying sensors in  $k$  lines to form barriers.

Deng et al. [18] studied the gap mending problem in a hybrid WSN which consists of both stationary and mobile sensors with adjustable sensing ranges. They proposed two gap mending schemes: the min-max scheme and the max-lifetime scheme. The first one minimizes the maximal energy consumption to move sensors, whereas the second maximizes the lifetime of barrier coverage after mending all gaps. Wang et al. [19] studied how to efficiently use mobile sensors to achieve  $k$ -barrier coverage. In particular, two problems were studied under two scenarios. First, when only the stationary sensors have been deployed, what is the minimum number of mobile sensors required to form  $k$ -barrier coverage? Second, when both the stationary and mobile sensors have been pre-deployed, what is the maximum number of barriers that could be formed? They proposed an optimal solution and a greedy solution for each of the two problems.

Zhang et al. [20] considered a scenario where sensors with adjustable ranges and a few sink nodes are deployed to form a virtual sensor barrier for monitoring a belt-shaped region and gathering incident data. The problem considers three relevant objectives: minimizing power consumption while meeting the barrier coverage requirement, minimizing the number of active sensors (reliability), and minimizing the transmission distances between active sensors and the nearest sink node

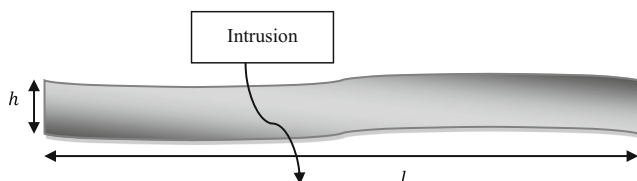
(efficiency of data gathering). Zhao et al. [21] focused on constructing energy efficient strong barrier coverage of a randomly deployed hybrid directional sensor networks on a long irregular strip region. First, they find the minimum gap barrier by modeling a directed graph for the static sensors in the network. Second, for each gap, they compute the repair location for mobile sensor to move to. Finally, the mobile sensor that has the shortest moving distance is chosen to repair the gap. Yu et al. [22] proposed local face-view barrier coverage, a novel concept that achieves statistical barrier coverage in camera sensor networks leveraging intruders' trajectory lengths  $\ell$  along the barrier and head rotation angles  $\delta$ . Using  $(\ell, \delta)$  and other parameters, they derived a rigorous probability bound for intruder detection for local face-view barrier coverage via a feasible deployment pattern.

All of the aforementioned works are based on a binary sensor coverage model. Although this latter simplifies the analysis, it may not be realistic in many cases [23, 24]. To avoid this limitation, in this paper, we consider the probabilistic coverage model that reproduces the probabilistic sensing behavior of sensors and thus captures the behavior of sensors more realistically.

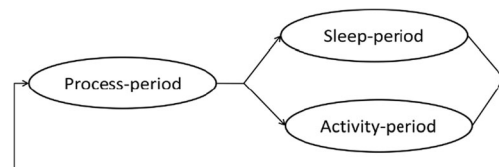
### 4 Assumptions

The proposed algorithms consider a network with  $N$  static sensors that are deployed randomly to supervise a zone. The latter is a belt region (band) of length  $l$  and width  $h$ , as shown in Fig. 4. The band is limited by two parallel curves. The first curve is located in the interior side of the supervised zone and the second curve is located in the external side. We suppose that an intrusion is successful if it crosses the barrier from beginning to end.

The WSN is supposed to be dense so that any point of the barrier is covered by several sensors. For our work, we choose the attenuated truncated coverage model [2], and we assume



**Fig. 4** Barrier schematization



**Fig. 5** System states of the sensors

that communication range is much larger than the sensing range. The choice of the sensors to be kept in the active state is based on a decision function having two parameters:  $E_{\min}$ , which is the minimal energy of the sensors composing the path that conduct from node S to the node  $C_i$ , and  $ND_{NBR}$  which is the number of these sensors. Since we want to maximize  $E_{\min}$  and to minimize  $ND_{NBR}$ , we choose a function given by Eq. (5), where  $\alpha'$  and  $\beta$  are strictly positive constants.

$$F_{\text{decision}}(C_i) = (\alpha' * E_{\min}) \left( \beta' * \frac{1}{ND_{NBR}} \right) \tag{5}$$

### 5 The proposed C-PBC, D1L-PBC, and D2L-PBC algorithms

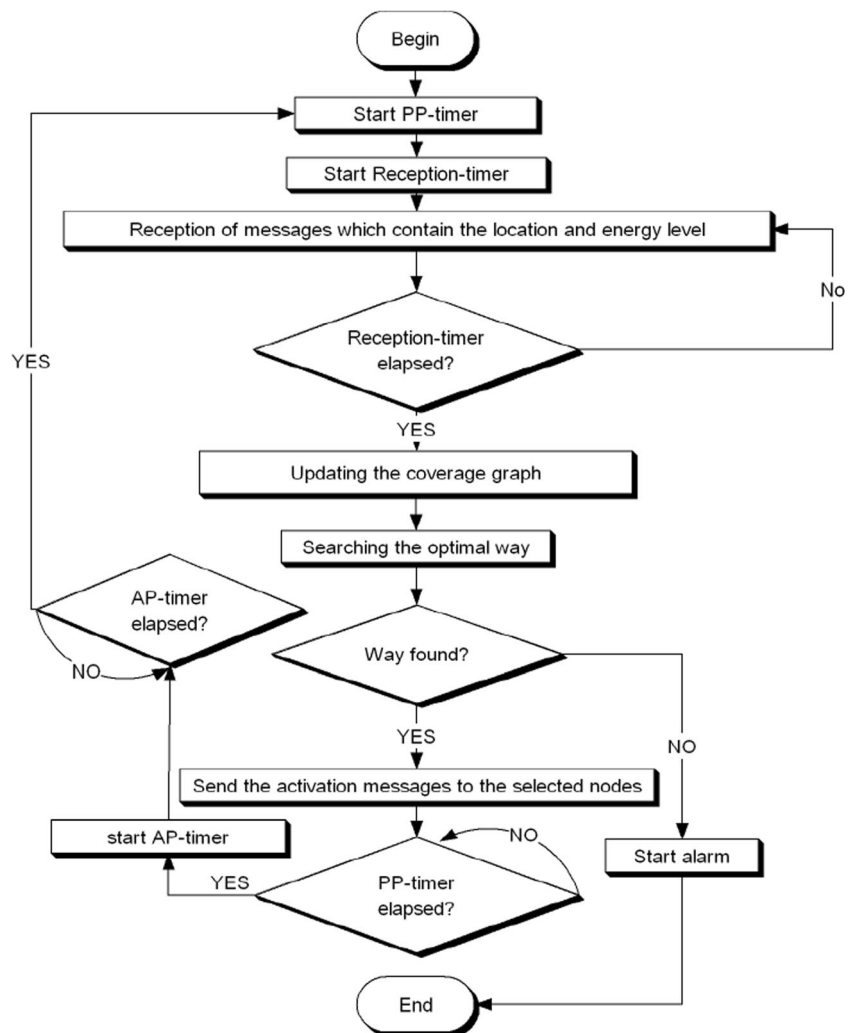
We propose three scheduling activities algorithms that are based on a probabilistic coverage model. The general operating mode of our algorithms is shown in Fig. 5. In the beginning, all the

sensors are in the process-period (PP) and they are all active. In this period, the system makes calculations to choose the sensors that will remain active. When the time allocated to PP is finished, the selected sensors remain active and go to activity-period (AP) in which they will provide the barrier coverage. The other sensors will be deactivated and will go to sleep-period (SP). At the end of AP and SP, all the sensors will be activated and will go to PP to re-execute the operation.

#### 5.1 The C1L-PBC algorithm

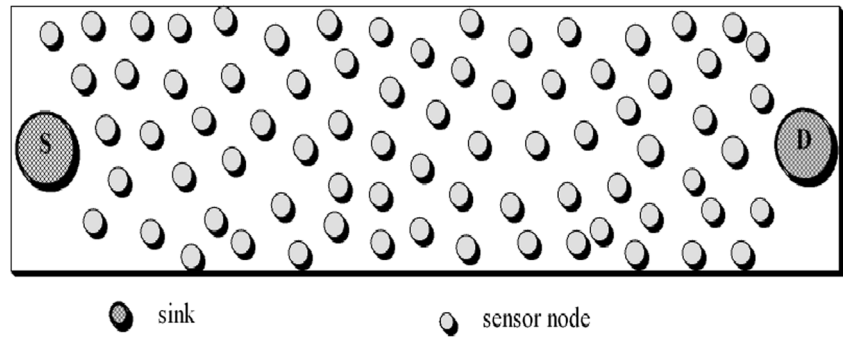
C1L-PBC is a centralized algorithm that provides barrier coverage with one line of sensors. All the calculations are done in one sink located in the center of the band. At the end of these calculations, the sink sends the activation (or deactivation) messages to the sensors of the band. C1L-PBC operates in three phases: information collection, coverage graph construction, and coverage-path selection. The flowchart depicted in Fig. 6

Fig. 6 Flowchart of the C1L-PBC process





**Fig. 7** The network model in the DIL-PBC algorithm



represents the execution of CIL-PBC in one sink. After deployment, the sink collects the information by receiving the messages sent by all the sensors. After that, the sink constructs a coverage graph  $G = (V, E)$  [3], which corresponds to the deployed WSN. Two virtual nodes  $S$  and  $D$  are added at the two ends of the barrier with the respective co-ordinates  $(0, h/2)$  and  $(l, h/2)$ , as shown in Fig. 3. These nodes are defined like *source node* and *destination node*, such as the arcs defined in the graph will be directed from  $S$  towards  $D$ .

After that, the system goes to PP state in which the sensors are active. In this state, the sink starts a timer, called reception-timer, during which it receives the messages from the sensors composing the network. Once reception-timer elapsed, the sink ignores the messages that will arrive after and updates the coverage graph. Furthermore, the sink starts the search procedure of the optimal path using the decision function  $F$  given by Eq. (5). If the path was not found, the sink sends an alarm to inform the network administrator that the coverage was not established. Otherwise, the sink sends an activation message to the selected sensors. Once the PP-timer elapsed, the sensors that receive the activation message go to the AP state (start AP-timer) in which they remain active. The other nodes go to the SP state (start SP-timer) in which they will be deactivated. Once SP-timer and AP-timer elapsed, the sink re-executes the above-described procedure.

### 5.2 The DIL-PBC algorithm

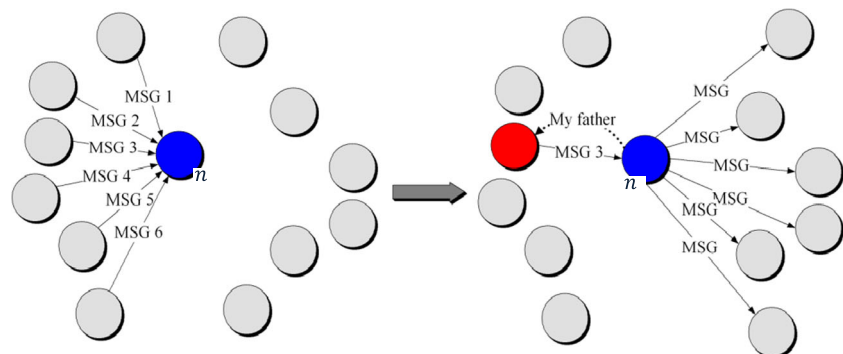
DIL-PBC is a distributed algorithm where the network model is represented in Fig. 7. The sensors are deployed randomly in a band and two real sinks  $S$  and  $D$  are placed at the two ends of the band. DIL-PBC uses discovery and activation messages. At the beginning of the optimal path search process (i.e., the sensors to put in active state), sink  $S$  sends a discovery message to its direct neighbors. Each node that receives this message makes its calculations and sends another message to its neighbors informing them of the beginning of the optimal path search process and proposing its participation in this path. Each node in the network makes the same operation until these discovery messages arrive to sink  $D$ . When sink  $D$  receives the various discovery messages from its direct neighbors, it chooses one of its neighbors and sends an activation message to it. Each node that receives the activation message sends it to the chosen predecessor neighbor until the message arrives at sink  $S$ .

The DIL-PBC algorithm is executed differently in three locations: in the sink  $S$  (Section 5.2.1), in an ordinary sensor (Section 5.2.2), and in the sink  $D$  (Section 5.2.3). Note that a flowchart representing the global execution of DIL-PBC is given in Fig. 13.

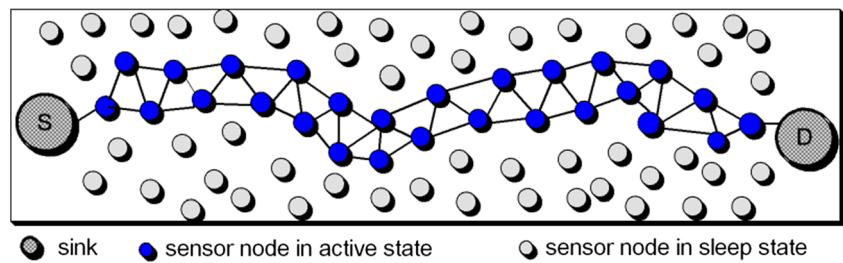
#### 5.2.1 Execution of DIL-PBC in the sink $S$

When a WSN is deployed, sink  $S$  puts itself in the active state and starts a timer called PP-timer, which correspond to the PP

**Fig. 8** Discovery message reception and father-node selection by an ordinary node  $n$  in the DIL-PBC algorithm



**Fig. 9** Barrier coverage ensured by the D2L-PBC algorithm



state in which the sensors to put in active state are selected. Then, sink  $S$  sends a discovery message to its neighbors. The message sent contains the identifier of  $S$ , its coordinate  $(X, Y)$  and numbers of nodes by which the message passed ( $ND_{NBR} = 0$ ). After that,  $S$  awaits the end of PP. Once this period is finished,  $S$  checks if it received an activation message. If the message was not received, then sink  $S$  informs the network administrator by starting an alarm. If not,  $S$  informs the administrator that the barrier was built and starts the AP-timer corresponding to the AP state in which the chosen sensors remain in active state and the others go to in sleep state. Once AP-timer elapsed, sink  $S$  passes to PP state by starting the PP-timer.

5.2.2 Execution of D1L-PBC in a sensor

The execution of the D1L-PBC algorithm in an ordinary sensor of the network is done in the following way. When a WSN is deployed, each sensor  $n$  is put in the active state and starts the PP-timer. Upon node  $n$  receives the first discovery message, it starts a reception-timer. This latter corresponds to the period during which node  $n$  stores in its table of messages the discovery messages received from its predecessor's neighbors.

A message is stored if and only if (a)  $X$ -coordinate of the receiver node is larger than  $X$ -coordinate of the sender node. This means that the receiver node is located after the sender node. (b)  $Pmin_{sender, receiver}$  given by Eq. (4) is less than the probability threshold. Once reception-timer elapsed, node  $n$  checks its table of messages. If this table is empty, then node  $n$  only awaits the end of PP. Otherwise,  $n$  analyzes the received messages and calculates the function  $F$  using Eq. (5) for each message. Further, the node having the greatest value of function  $F$  will be selected as father node. Then, node  $n$  sends its discovery message to its successor neighbors, as shown in Fig. 8.

Once PP-timer elapsed, node  $n$  checks the reception of an activation message. If such message was received, then the node goes to the AP state. If not, then node  $n$  passes in the SP state. Once the AP-timer and SP-timer elapsed, node  $n$  restarts the procedure.

5.2.3 Execution of D1L-PBC in the sink D

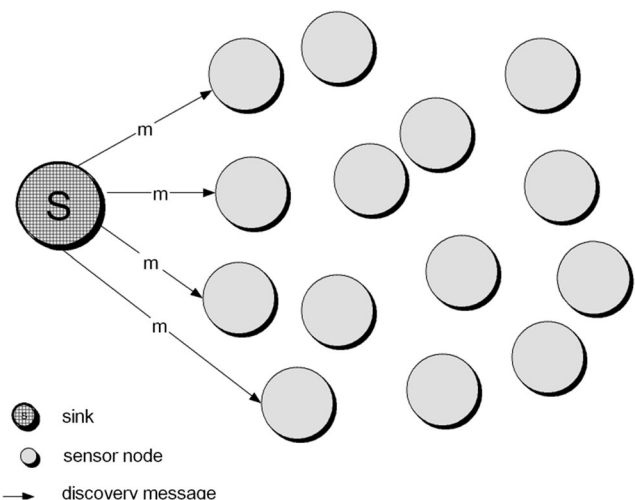
The execution of the D1L-PBC algorithm in the sink  $D$  is identical to its execution in an ordinary sensor of the network, except

that after the stage of messages analyzing and the stage of the choice of the father node,  $D$  does not send the discovery messages received because it is located in the end of the band and it is regarded as the last point of the barrier. After that,  $D$  identifies its father node and sends an activation message to it. Each node that receives an activation message sends it to the node chosen as father node. The nodes that will ensure the barrier coverage for this period are those that have received an activation message. Once the PP-timer elapsed,  $D$  goes to the AP state.

5.3 The D2L-PBC algorithm

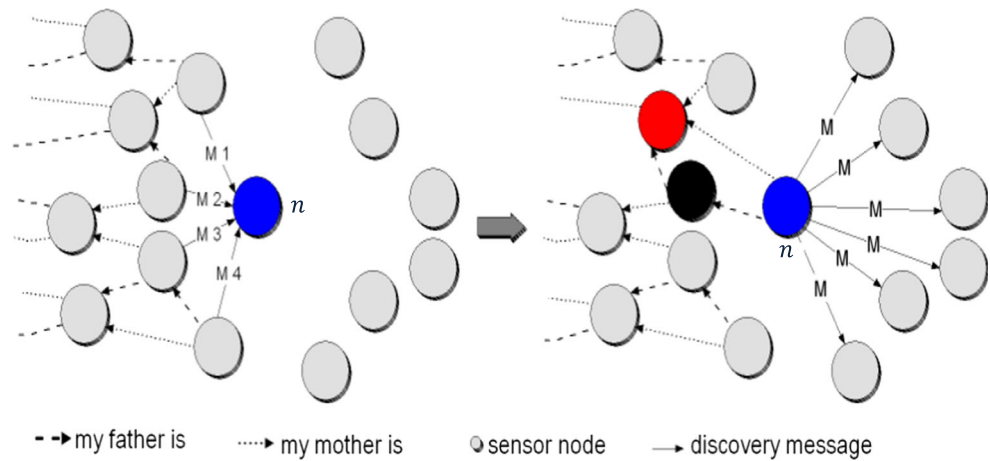
The D2L-PBC algorithm is an improved version of the D1L-PBC algorithm described in the previous section. Note that D1L-PBC ensures the barrier coverage with only one line of sensors, which means that when one sensor fails, it generates a failure in the barrier and then allows intruders to enter to the protected zone without being detected. Consequently, barrier coverage will not be guaranteed in this case by the D1L-PBC algorithm. To overcome this limit, the D2L-PBC algorithm, which is based on the same assumptions of D1L-PBC, comes to ensure two essential points:

- Reinforce the fault-tolerance level of the barrier coverage by leaving two levels of sensors in the active state so that if



**Fig. 10** Discovery messages diffusion by sink  $S$

**Fig. 11** Discovery message reception and father selection in an ordinary node  $n$  in the D2L-PBC algorithm



a sensor fails, there will remain at least another sensor to ensure the coverage, as shown in Fig. 9.

- Increase the probability of coverage so that any point of the barrier will be covered by at least three sensors.

The D2L-PBC process is not different from that of the D1L-PBC algorithm except that some treatments differ in the used discovery messages and in the selection of parents (father and mother) of a sensor. Thus, D2L-PBC is also executed in three different locations in the network: in the sink  $S$  (Section 5.3.1), in an ordinary sensor (Section 5.3.2), and in the sink  $D$  (Section 5.3.3). Note that a flowchart representing the global execution of D2L-PBC is given in Fig. 13.

5.3.1 Execution of D2L-PBC in the sink  $S$

At the beginning, sink  $S$  starts the PP-timer and sends a discovery message to its successor's neighbors (Fig. 10). Once the PP-timer runs out,  $S$  checks if it received an activation message to check if the barrier coverage is construct or not. Thereafter,  $S$  starts AP-timer and once this last runs out, it goes to PP state and re-executes the procedure.

5.3.2 Execution of D2L-PBC in a sensor

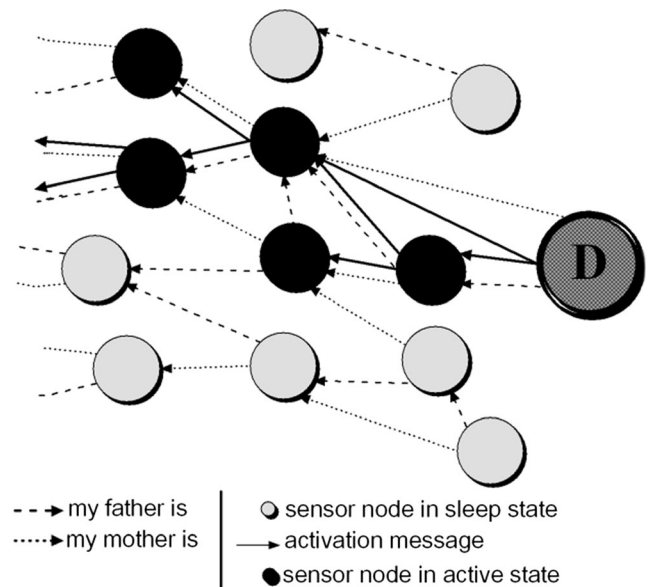
The execution of the D2L-PBC algorithm in any ordinary sensor  $n$  is little different from its execution in sink  $S$ . First, node  $n$  passes in the PP state, put itself in active state, and starts the PP-timer. Once node  $n$  receives the first discovery message, it starts the reception-timer. Until the latter is elapsed, node  $n$  awaits the discovery messages. The received messages are inserted in the table of messages. After that, node  $n$  checks if the table of messages is empty. If yes, then node  $n$  awaits the end of PP without doing anything. Otherwise, node  $n$  treats the discovery messages to elect its parents nodes (father and mother). Then, node  $n$  constructs its

discovery message and sends it to its successor's neighbors, as shown in Figs. 11 and 12.

Once the PP-timer elapsed, if node  $n$  does not receive an activation message, then it is put in the SP state and it starts the SP-timer. In the contrary case, node  $n$  checks the type of the received activation message. If this message is intended for a mother node then node  $n$  remains in the active state and starts the AP-timer. On the other hand, if the activation message is intended for a father node, then node  $n$  remains in the active state, sends an activation message to its father and mother, and starts the AP-timer. Once the timer elapsed, node  $n$  goes again to the PP state.

5.3.3 Execution of D2L-PBC in the sink  $D$

Since sink  $D$  is at the end of the barrier, it will be the last node that will receive the activation messages (Fig. 12). At the



**Fig. 12** Discovery message reception and father selection in sink  $D$  in the D2L-PBC algorithm



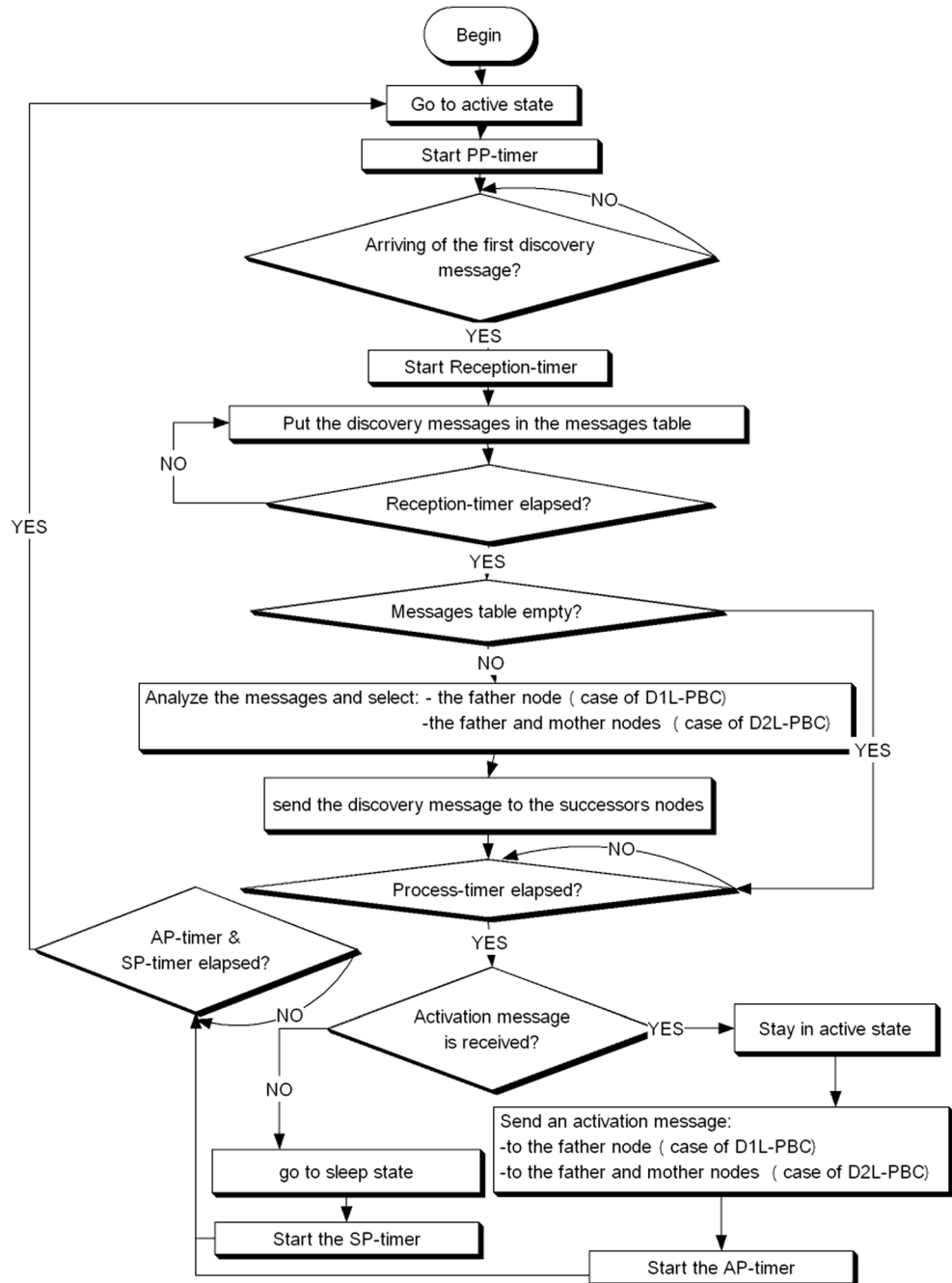
beginning,  $D$  is put in PP state and starts the PP-timer. During this period, execution of the D2L-PBC algorithm will be the same one as that in an ordinary sensor except that after selecting the father and the mother nodes,  $D$  does not send a discovery message. Once reception-timer elapsed, if  $D$  received discovery messages, it would have chosen a father and a mother node and sends an activation message to the father node, another one to the mother node, and it starts the AP-timer. Each node that receives this message will also send activation messages to its parents until it arrives at sink  $S$ . Once AP-timer elapsed,  $D$  restarts the operation (Fig. 13).

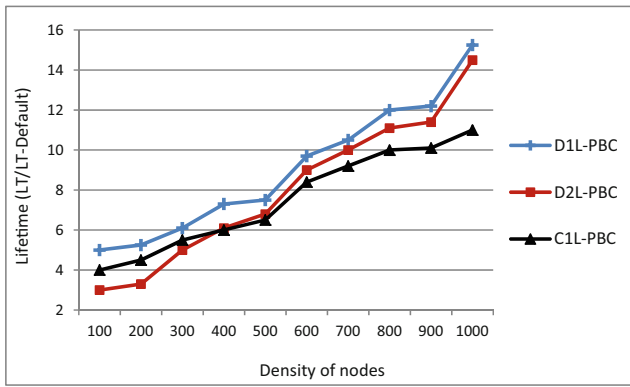
### 5.4 Message complexity analysis

For a WSN of  $n$  nodes, let us assume that, in average,  $n'$  nodes participate in the barrier. In C1L-PBC, the sink broadcasts a first message to ask for the state of the nodes. Each node replies with its state. After that, the sink performs local computation and informs the nodes that should stay in the active state; these nodes send back an acknowledgement message to inform the sink. In total, we have  $O(2n + 2n')$  message complexity.

In D1L-PBC, the sink  $S$  broadcasts a discovery message to its neighbors and each node that receives this message sends it

**Fig. 13** Flowchart of the algorithms D1L-PBC and D2L-PBC





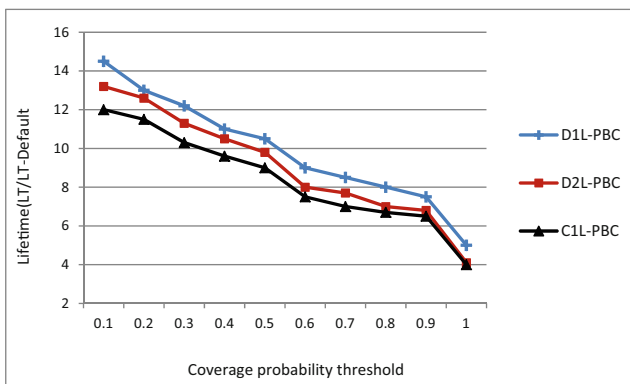
**Fig. 14** Impact of the sensor density on the network lifetime (see online version for colors)

to its neighbors until the message reaches the sink  $D$ . After that,  $D$  sends an activation message to the selected neighbor node, and this latter sends an activation message to its chosen neighbor node until this message arrives to the  $S$ . In total, we have  $O(n + n')$  message complexity. For D2L-PBC, we have also  $O(n + n')$  message complexity.

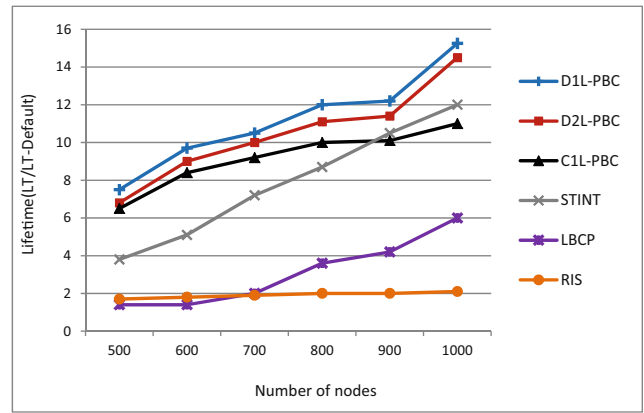
### 6 Performance evaluation

In this section, we present the performance evaluation of our proposed algorithms implemented in TinyOS [25] and evaluated using its embedded simulator TOSSIM [26]. For that, we use a band (1000 m × 100 m) with sensors deployed randomly. The communication radius of these nodes is fixed at 30 m and their initial residual energy is fixed at 20 mAh. The parameters of the coverage model  $\alpha$  and  $\beta$  are set to 1.

To evaluate our algorithms, we study the variation of the network lifetime (i.e., the period during which the network can ensure the barrier coverage) according to the following parameters: the density of sensors and the fault-tolerance level of the barrier (i.e., the probability threshold of coverage). To represent the network lifetime, at the beginning (i.e., after



**Fig. 15** Impact of the coverage probability threshold on the network lifetime (see online version for colors)



**Fig. 16** Comparison between our algorithms and those proposed in the literature

deployment process), we calculate the lifetime  $LT$ -default of the network without applying any algorithm, where all the deployed nodes remain in active state. Then, when we apply one of the previous algorithms and, for each lifetime  $LT$  obtained, we calculate the ratio  $(LT/LT\text{-default})$ . We repeat ten times each simulation with variation of the seed, and report arithmetic means on the graphs. We observed that for a degree of confidence of 95%, the simulations results remain in 2–10% compared to the average.

The first evaluation concerns impact of nodes density variation on the network lifetime. The results obtained are represented in Fig. 14. We notice that by increasing the number of sensors deployed in the band, the network lifetime increases for the three proposed algorithms. This is due to the fact that each time we increase the number of sensors, we increase also the number of sensors that will participate in the barrier coverage. We also notice that D1L-PBC gives better results. This can be explained as follows: D1L-PBC does not use a routing protocol, which minimizes the number of messages circulating in the network, and it uses fewer sensors to ensure the barrier coverage.

Concerning D2L-PBC, at the beginning, it is outperformed by the two other algorithms (C1L-PBC and D1L-PBC). This is explained by the fact that the number of sensors in the network is not important at the beginning and since D2L-PBC uses more significant number of sensors to ensure the barrier coverage that implies that it consumes more energy than the two other algorithms. On the other hand, when the number of sensors increases, the number of nodes used by D2L-PBC to ensure the barrier coverage becomes negligible compared to the total number of sensors. This explains the fact that performance achieved by D2L-PBC approaches that achieved by D1L-PBC.

The second evaluation concerns the impact of the coverage probability threshold on the network lifetime. In our scenario, we used a probability threshold that is fixed by the operator. The obtained results are presented in Fig. 15. They show that if we increase the coverage probability threshold, the network

lifetime decreases for the three proposed algorithms. This can be explained by the fact that the increase of the probability threshold decreases the node coverage field used, which incur more sensors to ensure efficient barrier coverage. Since the probability threshold defines the coverage fault-tolerance level of the established barrier coverage, we deduce that an increase in the fault-tolerance level implies a reduction in the network lifetime.

The third evaluation is a comparison between our algorithms and three existing algorithms: RIS [4], which is the simplest algorithm in the barrier coverage literature; STINT [3], which is a centralized scheduling algorithm; and LBCP [6], which is a distributed one. Figure 16 shows the obtained results. It can be noticed first that RIS shows the worst performance due to the fact that the nodes are activated without any coordination among them. STINT outperforms both LBCP and RIS, whereas our proposals have better performance than all other algorithms. Second, both D1L-PBC and D2L-PBC outperform C1L-PBC due to the higher number of exchanged messages in C1L-PBC (cf. Section 5.4). Finally, it can be observed that D1L-PBC consistently outperforms D2L-PBC. This is due to the fact that D2L-PBC ensures two-level barriers which consume more resources than the one-level barrier generated by D1L-PBC.

## 7 Conclusion

In this paper, we have presented the barrier coverage in wireless sensor networks as an appropriate concept of coverage when sensors are deployed to detect objects penetrating in a protected area. Since the energy of these nodes is a critical resource to take into account during the development of solutions, we have proposed three energy-aware activities scheduling algorithms to ensure efficient barrier coverage without negatively affecting the network lifetime.

The first proposed algorithm (C1L-PBC) is centralized, where all the calculations are performed in one sink, which incur important energy consumption. To deal with the important number of messages generated by C1L-PBC, the second proposed algorithm (D1L-PBC) decentralizes the management process of the barrier coverage. D1L-PBC operates with two sinks placed at the ends of the deployed network (i.e., the band). The third proposed algorithm (D2L-PBC) is an improved version of D1L-PBC. D2L-PBC is also distributed and aims to increase the fault-tolerance level of the barrier coverage. The obtained results show that our algorithms are efficient and increase considerably the network lifetime by using probabilistic coverage models.

In the future work, we plan to deploy our developed protocols in the Imote2 sensors for testbed experiments, in order to consolidate the simulation results presented in this paper.

## References

1. Mirsadeghi M, Mahani A (2015) Energy efficient fast predictor for WSN-based target tracking. *Ann Telecommun* 70(1–2):63–71
2. Wang B (2010) *Coverage control in sensor networks*. Springer, London
3. Kumar S et al. (2007) Optimal sleep-wake-up algorithms for barriers of wireless sensors. in *Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. Fourth International Conference on*. IEEE
4. Kumar S, Lai TH, Arora A (2005) Barrier coverage with wireless sensors. In *Proceedings of the ACM 11th annual international conference on Mobile computing and networking*
5. Kumar S (2006) *Foundations of coverage in wireless sensor networks*. The Ohio State University
6. Chen A, Kumar S, Lai TH (2007) Designing localized algorithms for barrier coverage. in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM
7. Shen C et al. (2008) Barrier coverage with mobile sensors. in *Parallel Architectures, Algorithms, and Networks, 2008. I-SPAN 2008. International Symposium on*. IEEE
8. Saipulla A, Liu B, Wang J (2008) Barrier coverage with airdropped wireless sensors. in *Military Communications Conference, 2008. MILCOM 2008*. IEEE
9. Bhattacharya B et al (2009) Optimal movement of mobile sensors for barrier coverage of a planar region. *Theor Comput Sci* 410(52): 5515–5528
10. Ssu K-F et al (2009) K-barrier coverage with a directional sensing model. *International Journal on Smart Sensing and Intelligent Systems* 2(1):75–93
11. Ban D et al. (2011) Distributed scheduling algorithm for barrier coverage in wireless sensor networks. in *Communications and Mobile Computing (CMC), 2011 Third International Conference on*. IEEE
12. Yang T, Fan P, Mu D (2011) Sliding the barriers in wireless sensor networks. in *Computing, Control and Industrial Engineering (CCIE), 2011 I.E. 2nd International Conference on*. IEEE
13. Yamamoto K et al. (2011) Barrier Coverage Constructions for Border Security Systems Using Wireless Sensors. in *Parallel Processing Workshops (ICPPW), 2011 40th International Conference on*. IEEE
14. Tao D et al. (2011) Strong Barrier Coverage Using Directional Sensors with Arbitrarily Tunable Orientations. in *Mobile Ad-hoc and Sensor Networks (MSN), 2011 Seventh International Conference on*. IEEE
15. Cao Y et al. (2011) Local maximum lifetime algorithms for strong k-barrier coverage with coordinated sensors. in *Communication Software and Networks (ICCSN), 2011 I.E. 3rd International Conference on*. IEEE
16. Chen J, Li J, Lai TH (2013) Energy-efficient intrusion detection with a barrier of probabilistic sensors: global and local. *Wireless Communications, IEEE Transactions on* 12(9):4742–4755
17. Du J et al (2013) Maximizing the lifetime of k-discrete barrier coverage using mobile sensors. *Sensors Journal, IEEE* 13(12): 4690–4701
18. Deng X et al. (2013) Mending barrier gaps via mobile sensor nodes with adjustable sensing ranges. in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE
19. Wang Z et al (2014) Achieving k-barrier coverage in hybrid directional sensor networks. *Mobile Computing, IEEE Transactions on* 13(7):1443–1455
20. Zhang X et al. (2015) Multi-objective Optimization of Barrier Coverage with Wireless Sensors. in *Evolutionary Multi-Criterion Optimization*. Springer

21. Zhao L et al. (2015) Energy efficient barrier coverage in hybrid directional sensor networks. in *Wireless Communications & Signal Processing (WCSP)*, 2015 International Conference on. IEEE
22. Yu Z et al. (2015) Local face-view barrier coverage in camera sensor networks. In *Computer Communications (INFOCOM)*, 2015 I. E. Conference on. IEEE. doi:[10.1109/INFOCOM.2015.7218437](https://doi.org/10.1109/INFOCOM.2015.7218437)
23. Senouci MR, Mellouk A, Oukhellou L, Aissani A (2012) An evidence-based sensor coverage model. *IEEE Commun Lett* 16(9):1462–1465
24. Senouci MR, Mellouk A, Senouci MA, Oukhellou L (2014) Belief functions in telecommunications and network technologies: an overview. *Ann Telecommun* 69(3–4):135–145
25. Levis P, Gay D (2009) *TinyOS programming*. Cambridge University Press
26. Levis P et al. (2003) TOSSIM: Accurate and scalable simulation of entire TinyOS applications. in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM