MobiBar: Barrier Coverage with Mobile Sensors

Simone Silvestri
Department of Computer Science
Sapienza University of Rome
Email: simone.silvestri@di.uniroma1.it

***Abstract*—Critical homeland security applications such as monitoring zones contaminated by chemical or biological attacks and monitoring the spread of forest fires, require the timely creation of barrier of sensors along the border to be monitored. The strict time requirements and the hazardous nature of these contexts impede manual sensor positioning. Mobile Wireless Sensor Networks have the potential to meet the desired coverage requirements, by exploiting the device locomotion capabilities.**

In this paper we propose MOBIBAR, a distributed and asynchronous algorithm for *k-barrier coverage* with mobile sensors. We formally prove that MOBIBAR terminates in a finite time and that the final deployment provides the maximum level of barrier coverage with the available sensors.

We compare MOBIBAR to a recent virtual force-based approach by means of simulations, which show the superiority of our solution. Furthermore, we show the self-healing capability of MOBIBAR to quickly recover from sudden sensor faults.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are the reference technology for monitoring large scale environments thanks to their flexibility, scalability and relatively low cost. Critical homeland security applications can benefit from the use of WSNs. As an example, wireless sensors can be used for monitoring zones contaminated by chemical or biological attacks, the spread of forest fires, or international borders to detect illegal intrusions [1], [2], [3].

The *Barrier coverage* model has been recognized to be an appropriate coverage model for such applications [1]. According to this model, a network is said to provide a *strong k-barrier coverage* over a belt region if every path completely crossing the width of the belt intersects the sensing range of k distinct sensors[1]. Barrier coverage allows the desired detection capabilities while requiring much fewer sensors with respect to full coverage [1].

The hazardous contexts typical of barrier coverage applications do not allow the manual positioning of sensors. Thus they are typically dropped from an aircraft or thrown from a distance. Mobile Wireless Sensor Networks (MWSNs) [4], [5], [6], can improve the initial deployment by exploiting the locomotion capabilities of the devices. In particular, mobile sensors can adapt their position in order to increase the level of barrier coverage provided by the network or to recover the maximum level of barrier coverage after the occurrence of device faults.

[1]In this paper we only consider strong barrier coverage. It will be shortly referred to as *barrier coverage* in the following.

Although mobile sensors present several good potentialities which can be exploited to meet the desired coverage requirements, the design of proper algorithms is challenging. A good solution should be able to meet the strict time requirements of the applications, should achieve the required level of barrier coverage while consuming as little energy as possible and should be able to self-heal after sensor faults, just to mention some of the most important properties. Moreover, it should be distributed and rely only on local information while showing a good scalability with the number of sensors.

Centralized solutions [7], [8], [3] are often inapplicable as they would require a high message overhead and the central unit would represent a single point of failure of the entire network. Therefore, the device movements should be regulated by means of a local coordination protocol. This makes it even more difficult to achieve the required performance objectives, since it is not possible to locally verify if the network provides k-barrier coverage [1]. Existing distributed approaches [9], [10], [11], [12], [13] often fail to meet the required performance objectives in terms of deployment time, energy consumption and scalability with the network size.

In this paper we introduce an asynchronous and fully distributed algorithm, called MOBIBAR, for achieving k-barrier coverage with mobile sensors over a belt Area of Interest (AoI). The goal of our algorithm is to achieve a final deployment which provides the maximum barrier coverage achievable with the available sensors. In particular, our main contributions are:

- We provide a fully distributed and asynchronous algorithm for k-barrier coverage formation with mobile sensors.
- We formally prove the termination and correctness of MOBIBAR, i.e. all sensors stop moving in a finite time and the final deployment provides the maximum level of k-barrier coverage achievable with the available sensors.
- We study the performance of MOBIBAR by means of simulations showing that it outperforms one of the most recently proposed distributed solution in terms of several performance metrics. Furthermore, we show the self-healing capability of MOBIBAR to recover after several sudden sensor faults.

II. RELATED WORK

The barrier coverage problem has been widely studied in the literature for static WSNs [1], [2], [14], [15], just to mention some of the most representative works. In this section we only consider previous works on barrier coverage with mobile sensors due to space limitations.

In [7] Saipulla et al. propose a centralized approach to relocate mobile sensors with limited mobility, i.e. sensors have an upper limit on the distance they can traverse in order to save energy. Bhattacharya et al. propose in [8] an optimization problem to calculate an optimal movement strategy for barrier coverage on a circular region. In [3] Saipulla et al. study the problem of relocating mobile sensors after an initial non uniform deployment. The above mentioned papers introduce *centralized approaches* to deploy mobile sensors. On the one hand, a centralized solution requires the central unit to be aware of the position of all sensors in the network which requires a high message overhead and could not be obtained if the initial deployment results in a disconnected network. On the other hand, the central unit would constitute a single point of failure of the entire network.

Distributed approaches have been proposed in [9], [10], [11], [12], [13]. In [11] Yang et al. propose a continuous movement strategy for monitoring a border. According to this strategy, sensors patrol the border so as to detect intrusions. Ban et al. introduce in [12] a distributed algorithm for k-barrier coverage. This algorithm partitions the AoI into rectangular cells and constructs a predefined pattern in each cell by locally calculating an optimal movement strategy.

The authors of [9], [10] introduce distributed algorithms based on virtual forces to determine sensor movements. Both solutions only aim at achieving a line segment deployment, i.e. a 1-barrier coverage between two given points. The proposal of Kong et al. in [13] is also based on virtual forces. They introduce a distributed algorithm for k-barrier coverage. We used this algorithm for performance comparisons with our solution, it is described in Section VI.

III. Network model and problem formulation

We assume a rectangular belt AoI of size $L \times W$, with $L >> W$, where N mobile sensors are initially deployed. We assume a binary sensing and communication model, i.e. two sensors are able to communicate if their distance is less than R_{tx} and each sensor is able to monitor a circular area of radius R_s centred at the sensor itself. We assume $R_{tx} \geq 2R_s$. The sensors are equipped with a positioning system such as a low cost GPS. The sensors are able to move over the AoI with a maximum speed of v_{max}. We assume that MobiBar is implemented over a communication protocol stack which handles possible transmission errors and message losses by means of timeout and retransmission mechanisms.

The goal of MobiBar is to reposition the movable sensors so as to achieve a *strong k-barrier coverage* [1] over the AoI. In particular, we aim at constructing k barriers of sensors such that every *crossing path* from a long edge of the AoI to the other, intersects the sensing ranges of at least k different sensors.

According to [1] a deployment with the maximum number of barriers can be obtained by placing sensors along lines, named *barriers*, which are parallel to the long edges of the

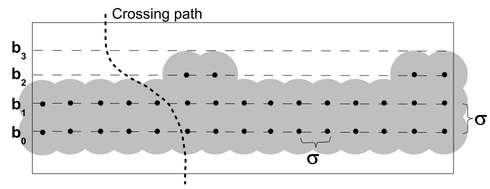


Fig. 1. A connected barrier component providing 2-barrier coverage

AoI. The sensors of the same barrier lie along a common line, equally spaced at a distance $2R_s$. The first sensor of each barrier is placed at a distance R_s from a short edge of the AoI. This deployment can achieve a maximum number of barriers $B^*_{max} = \lfloor \frac{N}{m} \rfloor$ where $m = \lceil \frac{L}{2R_s} \rceil$.

We argue that it may be difficult to achieve such a pattern in the context of mobile sensors as it requires a precise sensor placement. Indeed, even modern positioning systems such as GPS may induce non negligible positioning errors, which may practically impede such a precise placement. As a result, there could be some crossing paths which are less covered by the network and such weaknesses could be exploited by an attacker to reduce the chance of being detected.

In order to deal with the possible inaccuracies of the positioning systems, we propose to shrink the previous pattern by deploying sensors more densely on each barrier, i.e. at a distance $\sigma \leq 2R_s$. A similar approach has been proposed in [16] for full coverage. The value of sigma can be tuned according to the precision of the positioning system. We also do not assume an exact placement of the first sensor of a barrier with respect to an AoI short edge. As a result, the maximum number of barriers that can be constructed is given by $B_{max} = \lfloor \frac{N}{m} \rfloor$ where $m = \lceil \frac{L}{\sigma} \rceil + 1$.

Notice that, MobiBar can be configured to work with any value of σ. In the rest of the paper we will refer to *the maximum number of barriers* that can be constructed placing sensors at a distance σ as the value of B_{max} defined above.

The goal of MobiBar is the distributed construction of the maximum number of barriers B_{max} by only relying on local information. Since it is not possible to locally verify if the network is k-barrier covered [1], we require the barriers to form a *connected barrier component*. In particular, the barriers are created on parallel lines placed at a distance σ starting from a reference line, hereby called *base line*. Such lines are also parallel to the AoI long edges. In the following we will enumerate the barriers $b_0, b_1, \ldots$, where b_0 is the base line and the index of the other barriers increases with the distance from b_0. The sensors are placed on the barriers so as to ensure that nearby sensors on adjacent barriers are able to communicate. This is possible since we assume that $R_{tx} \geq 2R_s \geq \sigma$.

An example of the deployment described above is given in Figure 1, black dots represent sensors and grey zones their sensing ranges; dashed lines represent the parallel lines on which the barriers are created. Such a deployment enables a sensor detecting that its barrier is not completed to collaborate with other sensors in the component to fill the vacant positions.

IV. THE ALGORITHM MOBIBAR

In order to construct a connected barrier component with the maximum number of barriers, MOBIBAR provides the interleaved execution of four main activities, namely the *fix activity*, the *node search activity*, the *reposition activity* and the *combine components* activity. Before starting the execution of these activities each node moves from the initial location to the nearest point of the base line. We assume that the base line is known to the sensors before the deployment. We also assume that each sensor knows the number N of sensors initially deployed as well as the AoI coordinates.

We introduce the notion of *barrier priority* in the design of the algorithm activities. In particular, given a barrier b, the barrier priority is a function $\mathcal{P}(b)$ which reaches its maximum value on the base line and decreases with the distance from the base line of the other barriers. In the following, for a sensor s_i and a position p in the AoI, we will use the notation $\mathcal{P}(s_i)$ and $\mathcal{P}(p)$ to indicate the priority of the barrier on which s_i and p lie, respectively.

In the following we refer to *fixed sensors* as the sensors positioned on a barrier by MOBIBAR and to *movable sensors* as to the sensors not already placed on a barrier.

A. Fix activity

The goal of the fix activity is to give start to the creation of a connected barrier component and to expand the component created so far. After a sensor has reached the base line, if it has not received any message from its neighbours, it can give start to the fix activity in a random time instant in the interval $(0, T_{start}]$ becoming a fixed sensor.

According to the fix activity, every fixed sensor extends the connected barrier component created so far by making other sensors fix in the adjacent barrier positions, according to the hereby described protocol. Let us consider a fixed sensor s_{fix} on a barrier. At the beginning of the fix activity, s_{fix} performs a neighbour discovery in order to gather information on the sensors in radio proximity. On the basis of such information, s_{fix} is able to determine the set $VP(s_{\text{fix}})$, that is the set of adjacent vacant barrier positions to be filled in order to extend the barrier formation. For each vacant position $p \in VP(s_{\text{fix}})$, the sensor s_{fix} selects a movable sensor and sends it a FIX message, giving precedence to the vacant positions with higher priority. If a movable sensor is not available in radio proximity of s_{fix}, a fixed sensor q can be chosen if $\mathcal{P}(q) < \mathcal{P}(p)$. If there are not enough sensors in radio proximity to fill all the vacant positions, s_{fix} starts the node search activity.

When a sensor s receives a FIX message for a position p, if it has not been contacted for another barrier position with higher priority, it sends an ACKFIX message and starts moving towards p, otherwise it sends a NACKFIX message and the sensor s_{fix} will select another sensor.

Before fixing on p, the selected sensor s performs a contention resolution protocol in order to solve possible conflicts arising by the distributed algorithm execution when more sensors are assigned to the same position. The details of the contention resolution protocol are omitted due to space limitation. The sensor s that wins the competition for the position p, fixes there and starts the fix activity itself. Other sensors losing the competition for that position become movable and advertise the neighbourhood of their status.

We conclude the description of the fix activity by mentioning that if s_{fix} detects an adjacent vacant position p such that $\mathcal{P}(p) > \mathcal{P}(s_{\text{fix}})$ it stops the execution of the fix activity and tries to reposition itself according to the reposition activity.

B. Node search activity

The node search activity is started whenever a fixed sensor has not enough available sensors in radio proximity to fill all its vacant positions. The activity is performed by means of a limited broadcast of an NS (Node Search) message which is forwarded over the network by fixed sensors. A maximum hop counter h determines the forwarding horizon of an NS message. According to the node search activity, each node s, fixed or movable, maintains a value $\mathcal{P}_{\text{res}}(s)$ indicating its reservation priority. This value is initially set to $\mathcal{P}(s)$ if s is fixed or to zero if s is movable.

A fixed node that starts the node search activity, hereby called *inviter*, initially broadcasts a NS message with h set to 1. Such a message also contains the priority $\mathcal{P}$ of its vacant position with the highest priority.

When a sensor s receives an NS message with priority $\mathcal{P}$ and maximum hop counter h, it compares its reservation priority $\mathcal{P}_{\text{res}}(s)$ to $\mathcal{P}$. If $\mathcal{P}_{\text{res}}(s) < \mathcal{P}$, the sensor replies to the inviter proposing itself as a candidate to fill the vacant position and set $\mathcal{P}_{\text{res}}(s)$ to $\mathcal{P}$. From now on the sensor s considers itself *reserved* and does not reply to NS messages for positions with priority lower than $\mathcal{P}_{\text{res}}(s)$. Notice that, fixed sensors stop performing the node search activity when reserved. If on the contrary, $\mathcal{P}_{\text{res}}(s) \geq \mathcal{P}$ and s is fixed, s forwards the NS message if $h > 0$, with a decreased maximum hop counter. Each fixed sensors keeps track of the NS messages sent so far to avoid multiple retransmissions.

The inviter sends a FIX message for each vacant position to one of the sensors that have replied to the NS message. If such sensors are not enough to fill all the vacant positions, the inviter extends the search by sending an NS message with an increased maximum hop counter.

The sensors receiving a FIX messages from the inviter acts according to the fix activity. The other reserved sensors reset their reservation priority and do not consider themselves reserved any more after T_{res}, if they did not receive other NS messages with higher priority.

Notice that the inviter extends the search until all the vacant positions are filled or h has reached the maximum network diameter. In this latter case, the inviter needs to determine if there are some sensors that have not replied to the NS messages because they were reserved for a position with higher priority. To this aim, the inviter broadcasts an RS (Reserved Search) message which is forwarded over the entire component by fixed sensors as for NS messages. Only reserved sensors reply to the RS message. If no reserved sensor is found, the inviter stops the node search activity. Otherwise it waits a random time interval and starts the node search activity again.

C. Reposition activity

The reposition activity is introduced in order to let the fixed sensors relocate themselves whenever they locally detect a vacant position in an adjacent barrier with higher priority. In particular, after the neighbour discovery if a fixed node s detects a vacant position p such that $\mathcal{P}(p) > \mathcal{P}(s)$, it tries to fix itself on p as if it had received a FIX message.

D. Combine components activity

The random and distributed start of the fix activity may result in the creation of several misaligned barriers. Since we aim at achieving the maximum number of barriers with the available sensors, we want to obtain a single connected barrier component with equally spaced sensors on each barrier. The combine components activity achieves this goal by merging two connected components when they are in radio proximity. In particular, let us consider two connected components C_{new} and C_{old}. When a sensor s in C_{old} detects a sensor in C_{new}[2], it behaves depending on its role. If s is a fixed sensor, it tries to fix in the nearest barrier position of C_{new} as if it had received a FIX message for that position. If otherwise s is a movable sensor, it advertises its status to the sensors in C_{new}, resets its reservation priority and behaves as a movable of C_{new}. In both cases, s ignores further messages coming from C_{old}.

V. Algorithm properties

In this section we formally prove some properties of MobiBar. In particular, we prove that the algorithm has a guaranteed termination, i.e. all sensors stop moving after a finite time, and we prove its correctness, that is the final deployment provides the maximum barrier coverage level B_{max} with the available sensors. Due to space limitations only proof sketches are provided.

Definition V.1. *[Network state] Given a network with N sensors deployed over an AoI, a* network state *is a vector $S = [N_0, \ldots, N_{(N-1)}, M]$ where N_i is the number of fixed sensors currently placed on the barrier b_i and M is the number of movable sensor in the network. We say that $S' \succ S$ according to the lexicographic order.*

Theorem V.1. *[Termination] Given N sensors deployed over the AoI, the algorithm* MobiBar *terminates in a finite time.*

Proof sketch: MobiBar terminates when the four activities terminate for each sensor. We first consider the termination of the combine components activity.

MobiBar provides an initial movement on the base line. Then, each starter of the fix activity creates a connected barrier component which keeps on growing until no more sensors can be found. Since MobiBar privileges the completion of barriers with highest priority these components eventually arrive in radio proximity while constructing the barrier on the base line, giving start to the combine components activity. Such an activity can be performed only a finite number of times and naturally terminates as soon as a single connected barrier component is created or some disconnected barrier fragments are deployed on the base line if not enough sensor are available to complete a single barrier.

In order to prove the termination of the fix activity, the node search activity and the reposition activity, let us consider the algorithm execution after the termination of the combine components activity. All of the three activities provide to fix a sensor in a previously vacant position. As a result, the execution of these activities implies a transition from a state S to a state S'. We now show that every state transition from a state S to a state S' is such that $S' \succ S$.

Let us consider the position p on the barrier b_i that is vacant in S and taken in S'. All the sensors contending for p are either movable or fixed sensors with a priority lower than $\mathcal{P}(p)$, i.e. each fixed sensor reserved for the position p belongs to a barrier b_j with $j > i$. Therefore, $\forall\ l < i\ S'[l] = S[l]$, $S'[i] = S[i]+1$ and $S'[j] \leq S[j]$ for some $j > i$. Thus $S' \succ S$.

Since the network state space is finite for any fixed N and AoI size, the value of a network state is upper bounded. As any state change increases the value of the network state, we can conclude that the algorithm terminates in a finite time. ■

Theorem V.2. *[Correctness] Let N sensor with radius R_s deployed over an AoI of size $L \times W$ and let the algorithm* MobiBar *be configured to deploy sensors at a distance $\sigma \leq 2R_s$. The final deployment achieved by* MobiBar *provides a k-barrier coverage of the AoI with $k = B_{max}$.*

Proof sketch: Notice that, if there are not enough sensors to construct a single barrier, then $B_{max} = 0$ and the thesis is trivially verified. In order to prove the correctness when a sufficient number of sensors for a single barrier is available, let us consider the final deployment achieved by MobiBar. According to Theorem V.1, every fixed sensor has terminated the four activities. In particular, the termination of the combine components activity implies that there is a single connected barrier component in the final deployment. Furthermore, for every fixed sensor s, either it has filled all its vacant positions or there are some uncovered vacant positions with priority lower than or equal to $\mathcal{P}(s)$, otherwise s would have repositioned itself. Let us consider the fixed sensor s having the vacant position p with the lowest priority. It has terminated the node search activity, thus it has looked over the whole network for available sensors and it has not found any available nor any reserved sensor. As a consequence, there are no movable sensors in the network and every fixed sensor has a priority higher than or equal to $\mathcal{P}(p)$. This implies that there can be at most one incomplete barrier, that is the barrier on which p lies, and all the barriers with higher priority than p are completed. Since MobiBar equally spaces sensors on each barrier at a distance σ, the number of sensors placed on a complete barrier is $m = \lceil \frac{L}{\sigma} \rceil + 1$. Hence, the number of complete barriers constructed is $B_{max} = \lfloor \frac{N}{m} \rfloor$. ■

[2] We assume that each message contains the time-stamp of the creation of the barrier component to which the sender belongs to.

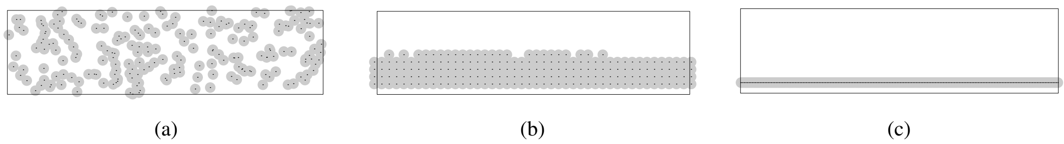


Fig. 2. Deployment examples of 200 sensors: initial random deployment (a), final deployment with MOBIBAR (b), final deployment with VF (c).

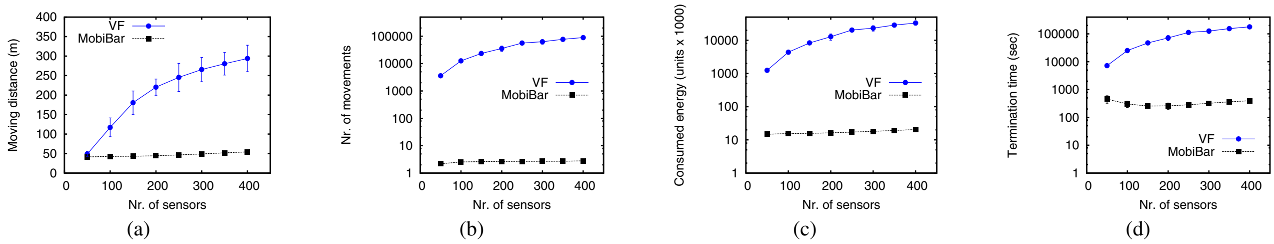


Fig. 3. Average traversed distance (a), average number of movements (b), average consumed energy (c), termination time (d).

VI. A RECENT VIRTUAL FORCE BASED ALGORITHM

In this section we describe the virtual force based algorithm proposed in [13], hereby referred to as VF, that we use for performance comparison with MOBIBAR. VF aims at achieving k-barrier coverage by equally spacing sensors on a single line.

VF has two phases. During the first phase each sensors performs a spiral movement centred at the initial sensor position in order to find the line where the barrier should be constructed. After reaching such a line, a sensor starts the second phase. Such a phase is structured in rounds, thus it requires the sensors to be synchronized. At each round, the sensors send their position information to the neighbourhood. Then, each sensor calculates the virtual force acting on it by considering the nearest sensor on the right and on the left of its position on the line. The algorithm terminates as soon as all forces are balanced. We refer the reader to [13] for more details. Notice that, in order to have fair comparison with our algorithm, the line on which VF constructs the barriers is the base line.

VII. PERFORMANCE EVALUATION

In this section we evaluate the performance of MOBIBAR by simulations. We develop a simulator on the basis of the wireless module of the OPNET simulation environment [17]. The simulation parameters are the following. The AoI is a rectangle of size $300 \times 80m^2$, the communication range R_{tx} is set to $15m$, the sensing range R_s to $5m$, the sensor movement speed v_{max} to $1m/s$. MOBIBAR deploys sensors at a distance $\sigma = \sqrt{2}R_s$ on each barrier. The results described in this section are averaged over 30 simulation runs. The error bars in the figures show the obtained confidence interval.

A. First set of experiments

In this set of experiments we compare the performance of MOBIBAR and VF. Before describing the results, we show an example of the final deployment achieved by MOBIBAR and VF. The initial random deployment of 200 sensors is shown in Figure 2 (a), while Figure 2 (b) and (c) depict the final deployment achieved by MOBIBAR and VF, respectively. Both of the algorithms achieve several levels of barrier coverage over the AoI. Nevertheless, we claim that the final deployment achieved under VF may be vulnerable to possible malicious attacks. Indeed, it is sufficient to compromise some nodes located nearby to significantly reduce the level of k-barrier coverage provided by the network. On the contrary, by equally deploying sensors on several geographically separated barriers, the final deployment achieved by MOBIBAR is more robust, because any crossing path intersects each barrier in different geographical locations.

Figure 3(a) shows the average traversed distance under MOBIBAR and VF. VF shows an increasing traversed distance with the number of sensors. This is due to the difficulty of finding a configuration where all the virtual forces are balanced, which is harder when more sensors are available. By performing precise movements, MOBIBAR achieves a lower traversed distance and shows a good scalability.

Figure 3(b) shows the number of movements performed by sensors on average. This is an important metric to evaluate mobile sensor deployment algorithms since a sensor consumes a significant amount of energy to start and stop a movement [4]. As the figure points out, VF requires a number of movements to terminate which is 2-4 orders of magnitude higher with respect to MOBIBAR (notice the logarithmic scale on the y-axes). As for the solutions based on virtual forces for complete coverage, the execution of VF is characterized by a high number of small movements necessary to find a final stable deployment [5]. On the contrary, MOBIBAR performs a lower number of movements since the sensors move only when necessary and movements are more efficient as they are performed on longer distances.

In Figure 3(c) we show the overall average energy consumption per sensor. Sensors consume energy for communications (sending and receiving messages), and movements (starting/stopping movements and traversing the desired distance). The figure shows a cumulative energy consumption metric expressed in energy units calculated on the energy cost model adopted in [4], [18], [5], [19]. In particular, receiving a message costs 1 unit, sending a message costs 1.125 units, 1m movement and starting/stopping a movement cost the equivalent of sending 300 messages. The above considerations made for the traversed distance and number of movements

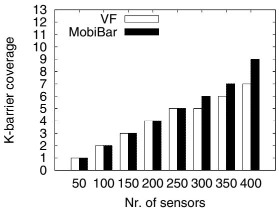


Fig. 4. k-barrier coverage of the final deployment.

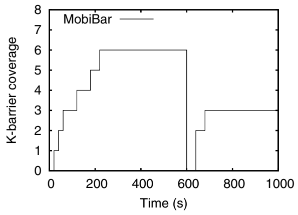
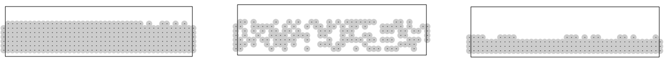


Fig. 5. Self-healing capabilities: k-barrier coverage over time.



(a) (b) (c)

Fig. 6. Self-healing capabilities: t = 220s (a), t = 600s (b), t = 680s (c).

reflect in the resulting consumed energy. The VF algorithm consumes 2-3 orders of magnitude more energy with respect to MOBIBAR.

The termination time of the two algorithms, i.e. the time at which the final deployment is achieved, is shown in Figure 3 (d). MOBIBAR outperforms VF also in this case, showing a termination time that is orders of magnitude shorter. Since the deployment time is of primary importance for mission critical applications where barrier coverage is needed, these results highlights the suitability of MOBIBAR in such contexts.

We conclude this set of experiments by showing the k-barrier coverage level achieved by the final deployment of MOBIBAR and VF. The results are shown in Figure 4. Both algorithms are able to construct several barriers over the AoI. Nevertheless, while VF only aims at equally spacing sensors on the border, MOBIBAR specifically targets the completion of each barrier with the minimum number of sensors required. As a result, MOBIBAR exploits the available sensors better than VF resulting in a higher barrier coverage level when the number of sensor increases.

B. Second set of experiments

In the third set of experiments we study the self-healing capability of MOBIBAR. In particular, we analyse the critical scenario where half of the available sensors suddenly cease to work due to an attack. We consider a scenario with 300 sensors randomly deployed over the AoI. We assume that the sensors are able to detect faults in the neighbourhood by using a periodic polling mechanism. When a fault is detected the sensors restart the algorithm activities accordingly.

Figure 6 (a) shows the final deployment achieved by MOBIBAR before the attack, this deployment terminates at the instant $t = 220$s. At $t = 600$s the network is attacked and half of the network cease to work, resulting in the deployment shown in Figure 6 (b). Notice that, after the attack the network does not provide any level of barrier coverage. Figure 6 (c) shows the final deployment achieved after the network reconfiguration. MOBIBAR has successfully self-healed and has reconfigured the network, providing the maximum number of k-barrier coverage (k = 3) with the available sensors.

Figure 5 shows the level of k-barrier coverage provided by the network over time. The initial random deployment ($t = 0$) provides no barrier coverage. In a short time MOBIBAR is able to build up the maximum number of barriers with the available sensors, achieving a $k = 6$ barrier coverage. At instant $t = 600$ the attack abruptly brings the barrier coverage level to zero. MOBIBAR quickly reconfigures the network, achieving a 3-barrier coverage in only 100s.

VIII. CONCLUSIONS

We proposed an original algorithm for k-barrier coverage with mobile sensors. According to our algorithm, sensors autonomously coordinate their movements in order to achieve a final stable deployment with the highest level of barrier coverage with the available sensors. We formally prove important properties of the algorithm and show its superiority with respect to a recent proposal in the literature by simulations. Furthermore, we show the capability of MOBIBAR to self-heal even after several sensor faults have compromised the coverage provided by the network.

REFERENCES

[1] S. Kumar, T. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *ACM Mobicom*, 2005.

[2] A. Chen, S. Kumar, and T. Lai, "Local Barrier Coverage in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 8, 2010.

[3] A. Saipulla, C. Westphal, B. Liu, and J. Wang, "Barrier Coverage with Line-Based Deployed Mobile Sensors," *Ad Hoc Networks*, 2011.

[4] G. Wang, G. Cao, and T. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, 2006.

[5] N. Bartolini, T. Calamoneri, T. F. L. Porta, and S. Silvestri, "Autonomous deployment of heterogeneous mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 10, no. 6, 2011.

[6] K. Ma, Y. Zhang, and W. Trappe, "Managing the mobility of a mobile sensor network using network dynamics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 1, 2007.

[7] A. Saipulla, B. Liu, G. Xing, X. Fu, and J. Wang, "Barrier coverage with sensors of limited mobility," in *ACM Mobihoc*, 2010.

[8] B. Bhattacharya, B. Burmester, Y. Hu, E. Kranakis, Q. Shi, and A. Wiese, "Optimal movement of mobile sensors for barrier coverage of a planar region," *Combinatorial Optimization and Applications*, 2008.

[9] C. Shen, W. Cheng, X. Liao, and S. Peng, "Barrier coverage with mobile sensors," in *IEEE I-SPAN*, 2008.

[10] T. Cheng and A. Savkin, "A problem of decentralized self-deployment for mobile sensor networks: Barrier coverage between landmarks ," in *IEEE ICCA*, 2009.

[11] G. Yang, W. Zhou, and D. Qiao, "Defending against barrier intrusions with mobile sensors," in *IEEE WASA*, 2007.

[12] D. Ban, W. Yang, J. Jiang, J. Wen, and W. Dou, "Energy-Efficient Algorithms for k-Barrier Coverage In Mobile Sensor Networks ," *International Journal of Computers Communications & Control*, vol. 5, no. 5, 2010.

[13] L. Kong, X. Liu, Z. Li, and M. Wu, "Automatic Barrier Coverage Formation with Mobile Sensor Networks," in *IEEE ICC*, 2010.

[14] G. Yang and D. Qiao, "Multi-round sensor deployment for guaranteed barrier coverage," in *IEEE INFOCOM*, 2010.

[15] S. Kumar, T. Lai, M. Posner, and P. Sinha, "Maximizing the Lifetime of a Barrier of Wireless Sensors," *IEEE Transactions on Mobile Computing*, vol. 9, no. 8, 2010.

[16] M. B. Johnson, D. Sariöz, A. Bar-Noy, T. Brown, D. Verma, and C. W. Wu, "More is more: the benefits of denser sensor deployment," *IEEE INFOCOM*, 2009.

[17] OPNET Technologies, "http://www.opnet.com/."

[18] G. Sibley, M. Rahimi, and G. Sukhatme, "Mobile robot platform for large-scale sensor networks," *IEEE ICRA*, 2002.

[19] N. Bartolini, T. Calamoneri, E. Fusco, A. Massini, and S. Silvestri, "Push & Pull: autonomous deployment of mobile sensors for a complete coverage," *Wireless Networks*, vol. 16, no. 3, 2010.