Detect smart intruders in sensor networks by creating network dynamics

Jie Tian \(^a\), Guiling Wang \(^a\), Tan Yan \(^a\), Wensheng Zhang \(^b,1\)

\(^a\) Department of Computer Science, New Jersey Institute of Technology, United States
\(^b\) Department of Computer Science, Iowa State University, United States

**A R T I C L E   I N F O**

Article history:
Received 27 February 2013
Received in revised form 7 October 2013
Accepted 21 November 2013
Available online 9 December 2013

Keywords:
Wireless sensor network
Mobile sensor network
Intruder detection

**A B S T R A C T**

Intruder detection is an important application in wireless sensor networks. This paper focuses on detecting smart intruders whose purpose is to cross the monitored area and who can calculate the minimum exposure path with the location information sensors. A mixture deployment of mobile and static sensors are proposed in this paper. The central idea is that mobile sensors move smartly to dynamically change the topology and minimum exposure path of whole network. Thus, even the smart intruders can be detected with a high probability. Simulation results show that the proposed scheme can achieve a high detection probability with small energy consumption of mobile sensors.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Intruder detection \([4,15,16,24,26]\) is a very important application of wireless sensor networks. This paper focuses on detecting moving intruders whose objective is to traverse a monitored area. For example, in a battlefield, an enemy may try to traverse a protected area to conduct some malicious tasks, the successful detection of which before he passes through is important. In this paper, we assume powerful intruders who are capable of scrutinizing the deployment of monitoring sensors by certain methods and have a full knowledge of current locations of sensors before their departures. Given the locations of sensors in the monitored field, an intruder can calculate a safest path to follow, following which, the intruder has a least probability of being caught. The path is called Minimum Exposure Path (MEP). The characteristics of MEP have been well studied in \([1,25]\) and thus the intruder is aware of the path. The MEP is the bottleneck of the monitoring sensor networks. The key to improve the detection rate is to make the MEP less safe for intruders. One way to guard the MEP is to deploy a lot more sensors. However, this increases cost. More importantly, when precise manual deployment is not feasible, some areas are not likely to be covered by sensors because of obstacles, etc. and the uncovered areas of the monitored field remain uncovered even more sensors are deployed. Once the sensors are deployed, the MEP is fixed and intruders can calculate it and always move along the MEP.

Mobile sensors have been employed to increase the network coverage and detection rate in \([2,27]\). The authors in \([2]\) studied the case when all sensors are mobile. The performance of these mobile sensor networks are least sensitive to the initial sensor placement, but the networks have a relatively high cost. To get a tradeoff between cost and performance, previous researches have proposed deploying a mix of mobile and static sensors and studied the collaboration between static sensors and mobile sensors for better coverage and detection rate \([21,29]\). For example, reactive sensing, a two-phase detection scheme, was proposed in \([29]\). In the first phase, static sensors detect and make a rough decision that an intruder may appear in a place and in the second phase, static sensors notify mobile
sensors to move to the place to confirm the appearance of an intruder. The performance of reactive sensing is sensitive to sensor distribution and static sensor failure. If some areas are not covered by static sensors or static sensors in the area fail, and initially there is no mobile sensors around, it is very likely that the area will never be covered and a smart intruder can always take this advantage. In this paper, a mix of mobile and static sensors are deployed for intruder detection, same as some previous work. But we utilize the mobility from a different and brand new perspective.

Our central idea is that mobile sensors move smartly to change the topology and MEP dynamically. The original safest path an intruder follows may become a dangerous one as it follows the path. Even though the smart intruder can change its direction based on the newly obtained local information, the intruder may already enter an area that is surrounded by many sensors.

The general design principle of our scheme is mobile sensors move around in the network, visiting less covered areas to improve the most vulnerable part of the monitored field and visiting more covered areas once in a while in an unpredictable way to improve the unpredictability of the MEP. In this way, the MEP changes dynamically and randomly and it becomes difficult for intruders to explore the MEP. The objectives are: (1) to improve the detection rate and early detection of powerful and smart intruders; (2) to keep a low movement and communication cost; and (3) to improve the fault tolerance of the network, i.e., if sensors fail, at a cost of more dynamic movement, required detection rate and detection time can still be maintained.

To achieve the above three objectives, a distributed scheme is proposed to direct the movement of mobile sensors for better detecting intruders. The scheme divides the monitored field into different grids, and thus the sensors are in different groups. The method to divide grids changes over time. Therefore, throughout the lifetime of the network, in different running rounds, sensors are associated with different groups and move within different grids. We introduce two algorithms for this distributed scheme: one is group division algorithm and another is intra-group motion planning algorithm. The group division algorithm guides the network to generate grids dynamically. The algorithm is controlled by two parameters, one of which determines how a division in one round is different from that in the next round, and another determines how frequently the division changes. The intra-group motion planning algorithm directs mobile sensors to move to the least covered area in its grid and generate high dynamics for the network to improve the original MEP at the same time.

1.1. Paper organization

Section 2 introduces the detection model and intruder model adopted in this paper. The group division algorithm and intra-group motion planning algorithm are presented in Section 3. Section 4 shows performance evaluation results. Finally, the related work is in Section 5 and conclusion of this paper is in Section 6.

2. System model

2.1. Target detection model

Sensors are deployed in an area to detect targets by measuring the energy of signals emitted from them. There are two commonly used models in wireless sensor network. One is the 0/1 model, which usually adopts voronoi diagram to evaluate the detection possibility. Another is the fusion model, which uses the energy measurements of targets by sensors to calculate the probability of detection. Our distributed scheme does not depend on the types of models. But for the high detection accuracy [1,21,29] of the fusion model and to align with related work, this paper adopts the multi-sensor group fusion model as the detection model. Once every period T, a group header collects the energy measurements from its group members.

In terms of a group member $n_i$, it samples the energy signal value from a point $v$ based on a built-in sensing frequency of the hardware for a duration $T$ and reports the averaged value to the group header. The group header further averages all the collected measurements and compares the average of all measurements to a detection threshold $\eta$. The probability of collaboratively detecting an intruder at position $v$ in one group of $m$ sensors is denoted by $p_v$. $p_v$ is formulated as follows, in which $e_i(v)$ represents the signal power measured by sensor $i$ for a duration $T$ from position $v$ and $F_{\chi^2}$ represents the Chi-square distribution with $m$ degree of freedom. More details can be found in [29].

$$p_v = 1 - F_{\chi^2}(mn - \sum_{i=1}^{m} e_i^2(v))$$  \hspace{1cm} (1)

2.2. Intruder model

This paper assumes powerful and smart intruders who have assistance and can obtain the sensors' location before they depart for the field monitored by the sensor network. However, they have limited vision and after they enter the field, they can only see sensors close to them. They cannot see a mobile sensor moving towards them if the sensor is out of their vision range. We also assume intruders have a limited speed such that they will not be able to flee the field fast enough without being detected by any sensor close to them.

The intruders are assumed rational and smart. They take one of the two strategies when traverse the field. One is to choose the MEP in its visible local area. The intruder taking this strategy is called Local-optimal Intruder (LI). A LI calculates the local MEP of the area that it can reach in one time unit at its maximum speed, and follows the path. The path is dynamically adjusted as it moves. Another is to choose the MEP of the whole area. The intruder taking such a strategy is called Global-optimal Intruder (GI). A GI calculates the MEP of the area between its current position and the field boundary where it tries to pass. Note that a global-optimal strategy may not necessarily be better than the local one, considering the positions of mobile sensors are unpredictable and an intruder only knows the positions...
of mobile sensors in its vision range and the positions of static sensors. In the performance evaluation, intruders follow random walk are also considered, in addition to GIs and LIs.

2.3. Assumptions

We assume sensors are aware of their locations through GPS or other localization methods [5,9,17,30]. Since our protocol goes round by round, we also assume that the network is loosely synchronized such that mobile sensors can start moving at approximately the same time in each round. Considering that current synchronization algorithms [6,31,32] can achieve very tight synchronization at low overhead, our requirement for movement synchronization is easy to be satisfied without incurring much overhead. We assume the communication among sensors is secure, which can be easily taking care of with various protocols [22,23].

3. Motion planning

The objective of sensors’ movement is to frequently visit areas less covered by static sensors to improve the most vulnerable area of the monitored field and to visit other areas once a while to create network dynamics and the unpredictability of the MEP. At the same time, the movement of mobile sensors should be minimized to save cost. Specifically, in some areas where static sensors are sparsely deployed, there should always be some mobile sensors to provide a desired coverage and the mobile sensors should move around locally in the area to continuously change the local topology; in other areas, some mobile sensors should move across once a while to provide global dynamic topology.

Considering the whole lifetime of a sensor network, generally there are two phases. One is the start period, when sensors self initialize for normal functions. The other is the running phase, when sensors conduct the instructed detection tasks. In the beginning of the start period, the sensors are initially deployed and the distribution may be very irregular. The motion planning algorithm should be able to push a large portion of mobile sensors to the sparse area to improve the most vulnerable part in the monitored field. During this phase, a roughly regular distribution of sensor nodes can be formed such that even the MEP is reasonably covered. After that, in the running phase, the mobile sensors move to create network topology dynamics, which is to move around between the currently most vulnerable area and the best covered area in the network. In this way, the MEP is dynamically changed in an unpredictable way.

To realize this design philosophy, we propose a distributed motion planning protocol to direct the movement of mobile sensors. In the protocol, the monitored field is divided into grids and sensors are divided into groups. The protocol goes round by round. In different rounds, the group division methods are different. For example, shown in Fig. 1, in even rounds, the network is divided by the black solid lines and in odd rounds, the network is divided by the blue dotted lines. The groups in even rounds have intersection with groups in odd rounds. Thus, in different rounds, each sensor is associated with different groups.

In a round, the movement of a mobile sensor is always kept within its grid. Inside a grid, the group header collects the location information of all its mobile and static members and calculates the moving direction of mobile sensors to the least covered area in the group. After the movement, effectively, sensors’ distribution inside the group is more regular and thus improve the group’s most vulnerable part. Also, the location of the most vulnerable part changes, which means, an originally well-covered area becomes the least covered, while the originally least covered is better monitored. If in the next round, the group division is

![Fig. 1. Dynamic local–global group: a zoomed-out overview of the network.](image)
the same, the mobile sensors still move to dynamically change the local MEP within the same grid in the same way. If in the next round, the group division is different, then the mobile sensors move in the same way in a different grid.

In this way, think about most sensors: Once they enter a sparse area, there is a high probability that they are kept there because the movement in one round is kept local. On the other hand, think about an individual mobile sensor: In one round, it moves inside a grid and in the next round it moves in a different grid to a different location. Thus, with certain small probability, a mobile sensor moves from a sparse area to a dense area.

In the protocol, two parameters are adopted to control the division and movement. Parameter \( \alpha \) is to control how the group division changes over rounds and parameter \( \beta \) is to control the changing frequency of the grids in the network.

In the following, we first describe the group division algorithm and then we present the intra-group motion planning algorithm. We illustrate how the protocol runs using an example after that. Parameters, including \( \alpha \) and \( \beta \), are discussed at the end of the section.

### 3.1. Group division algorithm

Before we present the group division algorithm, we first introduce a term grid frame, which determines how a network is grided differently in different rounds. Shown in the left figure of Fig. 2(a), the dotted lines represents the grid frame, which is size of \( 2 \times 2 \) grids. The network is divided into four grids accordingly based on the frame. In the right figure, the grid frame is in a different position and thus the network is divided into nine grids, one of which is of the same size as before and all the others are of smaller size. A shift of the grid frame makes a different network division.

In different rounds of the protocol, the grid frame is in different positions, and thus the network is grided in different ways. How much the grid frame moves between rounds determines how much the groups are different and this is controlled by parameter \( \alpha \). Suppose the square grid length is one. In each round, the grid frame shifts \( \frac{1}{\alpha} \) in both the horizontal and vertical directions. When \( \alpha = 2 \), the grids divisions in different rounds are shown in Fig. 2(a); when \( \alpha = 3 \), the grids divisions are shown in Fig. 2(b). It is clear to observe, as the \( \alpha \) increases, the change of the grids division becomes smaller and the intersection between groups becomes larger in the consequent rounds. Here we call the very original grid division Original Division, as shown in the left figures in Fig. 2 and all the other divisions Varied Divisions, as shown in the right figures in both Fig. 2(a) and (b), and the middle figure in Fig. 2(b).

We introduce parameter \( \beta \), which controls the frequency of grid frame shifts. More specifically, \( \beta \) is the number of rounds the network stays in a group division before it shifts to another. Shown in Fig. 3(c), for example, \( \alpha = 3 \) and \( \beta = 2 \). There are three ways of divisions. The network stays in each one for two rounds \( (\beta = 2) \) and then shifts to another. As \( \beta \) becomes larger, the change of the grid divisions and thus the change of groups becomes less frequent. In another way, as \( \beta \) becomes larger, a mobile sensor stays in a same group for a longer time.

Fig. 4 shows an example of how the network is grided based on \( \alpha \) and \( \beta \). Here, \( \alpha = 4 \) and \( \beta = 1 \). In the first round, the network stays in the Original Division and is divided by the black solid lines, as shown in Fig. 4(a). In the second round, the network shifts to a Varied Division and the network is divided by the blue dotted lines, also shown in Fig. 4(a). It is clear to see some mobile sensor nodes are in different groups in different rounds, such as mobile nodes 1 and 2, so their motion directions change. From the second round to the fourth round, the grid frame, shown in blue dotted lines in Fig. 4, shifts every one fourth of the square grid length in both horizontal and vertical directions to form the different Varied Divisions in the network and thus sensors are in different groups. This process can be observed in Fig. 4(b) and (c). Finally, in the fifth round, the grid frame is back to the original position to form the Original Division same as in the first round. Because the groups in one round have intersection with groups in the next round, in different times, the sensors have a high probability to be associated with different groups, such as the mobile sensor nodes 3 and 4, shown in Fig. 4(b) and (c).

### 3.2. Intra-group motion planning

A group header has the location information of all the static sensors and mobile sensors in its group. It calculates the target positions of the mobile sensors and notifies them the directions to move. In order to calculate the target positions of mobile sensors, it first needs to determine which positions in the group are the best positions for an intruder to stay, most likely to remain undetected. The group header uses a discretization method to determine the positions. An X–Y coordinate axis system is automatically established in each sensors, when they are deployed.
in the area. Only the coordinates of integer value are considered. Then each grid is divided into a finite number of points (small squares shown in Fig. 5). According to Eq. (1), for each point $v$, the probability that an intruder can be detected is expressed as

$$p_v = 1 - F_{F_{\text{mm}}} \left( \left( n_m + n_s \right) \eta - \sum_{i=1}^{n_i} e_i(v) \right),$$

where $n_s$ is the number of static sensors and $n_m$ is the number of mobile sensors.

The safest point $v$ for an intruder to stay is such a point: at point $v$, considering its limited speed an intruder can only reach a number of points, the summation of whose
detection probability is minimum. We call point $v$ the 
weakest point (WP). Note that in many cases, the WP is 
not unique and there are many points which are equiva-
ently weak. In this case, we choose one WP as the target 
position using the following rule: with a high probability 
$p$, the WP farthest to any mobile sensor is chosen with 
probability $1 - p$, one of the remaining WPs is randomly 
chosen. The philosophy behind is to spread mobile sensors 
as well as to generate unpredictable dynamics.

After determining the WP, a group header first virtually 
moves the closest mobile sensor to the WP. Shown in Fig.
5, group header $s_4$ calculates the position of WP and virtually 
moves mobile sensor $m_1$ to it. After that, it assumes $m_1$ is in 
the new position and recalculates a new WP, then virtually 
moves $m_3$ there. The procedure continues until all mobile 
sensors in the group have a virtual position. Then $s_4$ notifies 
all the mobile sensors in its group to move to their virtual 
positions. In case a mobile sensor $m_i$ cannot reach its target 
position by the end of the round because of its speed limit, 
it will virtually move to the closest position to its target 
location. The subsequent calculation of the new WP is based 
on real virtual position of $m_i$, instead of the calculated one. 
The detailed algorithm is described in Algorithm 1.

**Algorithm 1.** Intra-group motion planning algorithm in 
group header

```
for $i = 1$ to $n_m$ do
    Calculate $p_v$ for all the points.
    $WP = v$ with smallest $p_v$.
    Move the node, which has not been moved once, 
closest to $WP$ to $v$.
end for
```

Note that given the positions of static sensors, there is 
an optimal positioning of mobile sensors for the best cov-
erage and highest exposure of the MEP. A group header 
could calculate it and let mobile sensors move there. How-
ever, mobile sensors become static after that, and the MEP 
calculated in this way becomes a fixed one and thus re-
mains the bottleneck of the network. That is why in our 
protocol a group header sequentially calculates the new 
positions of mobile sensors with some randomization. In 
this calculation, coverage holes are constantly covered by 
the coming of mobile sensors and generated by the left 
of them, and thus the places of holes and the MEP are con-
stantly changed in an unpredictable way.

We use an example to illustrate the effectiveness of in-
tra-group motion planning. As shown in Fig. 6, the first 
subfigure presents the positions of all sensors in the begin-
nning of a round at time $t$. The Voronoi polygons regarding 
to these sensors are plotted in blue lines to present a rough 
idea of the MEP. The purpose of the intruder is to follow 
the red dashed line, which is the safest path, to go through 
the monitored area. With the motion planning algorithm, 
mobile sensors move to new positions shown in the second 
subfigure and the time is $t + 1$. The calculation procedure 
of the new positions is illustrated in Fig. 5. It is clear to 
see that mobile sensor $m_1$ is close to the intruder in this 
time unit, which increases the probability of detection. 
Even the intruder is not detected and calculates a new path 
by observing the new position of $m_1$, in the next round, 
time unit $t + 2$, the mobile sensor $m_3$ gets close to the in-
truder again and has a high possibility to detect it.

The movement of one mobile sensor is always kept 
within local grid until the grid shifts. The merit of such mo-
tion plan is that the mobile sensors can always change lo-
cal topology to detect intruders, since intruder’s movement 
is always local. There is a very low probability that
intruders can go through more than one grid in one time unit according to the group division and intruder's speed. Thus the motion plan can largely increase the detection probability. If the mobile sensor nodes are free to move around the network without constraints, their jobs will first become to compensate the areas having low detection probability with spare sensor nodes. The topology's changing will become less in local, which will decrease the detection probability of a local intruder.

3.2.1. Discussion about group header

A group header plays an important role in the intra-group motion planning. As for the selection of a group header, many existing methods with low-overhead [11] can be applied directly. In this paper, we choose the sensor which is closest to the center of the grid and has a relatively high remaining energy to be the group header of a grid. Basically, each sensor in a grid broadcasts its remaining energy and location. If a sensor has the highest remaining energy, it becomes the header; if two sensors have the same energy level, the one closer to the center of the grid becomes the header.

The group header calculates the target moving positions of its mobile members. Suppose it has \( n_m \) mobile sensors. For each one, it needs to calculate the WP among all the points in the grid. Suppose there are \( n_p \) points. For each point \( \nu, p_\nu \) is calculated based on Eq. (2), the complexity of which is close to a constant considering the number of sensors are far less than the number of points. Once the \( p_\nu \) of all the \( n_p \) points are calculated, the smallest one needs to be found, the complexity of which is \( O(n_p) \). Thus the total computational complexity is \( O(n_p) \).

3.3. A snapshot

Fig. 7 is a snapshot showing how the whole motion planning is proceeded. Ten static sensors, shown as blue dot and ten mobile sensors, shown as red dot are randomly deployed. Fig. 7(a) is their initial placement. We consider three types of intruders: a Local-optimal Intruder, represented by blue dot, a Global-optimal Intruder, represented by green dot, and for reference, a randomly moving intruder, represented by red dot. The intruders start their intrusion at the mid-point of the left boundary of the network and go cross the whole monitored area from left to right. In the figure, we have a scale color bar for each subfigure. The darker the color, the higher detection probability in the corresponding position.

The intruders enter the field at \( t = 0 \). From Fig. 7(a), the distribution of sensors are very irregular at \( t = 0 \). There are few sensors in area bounded by cells with x-axis: [30,40] and y-axis: [40,60]. The detection possibility is low in this area. At \( t = 5 \), mobile sensors move to heal this hole and improve the detection probability, as shown in Fig. 7(b). At \( t = 10 \), it is clear that the global-optimal intruder is no longer perfect because of the movement of mobile sensors, as shown in Fig. 7(c). Also at this time instant, both rational intruders move close to some mobile sensors, as a result they can be detected with high probability. As the time goes, the situation goes worse for all the intruders. At \( t = 15 \), mobile sensors move closer to the intruders and force them moving in a zigzagged path to avoid mobile sensors, as shown in Fig. 7(d).

3.4. Discussion of the parameters

Three parameters are involved in the protocol: grid size, \( \alpha \), and \( \beta \). Grid size is determined by several factors. The first one is the physical characteristics of the sensing circuit of the sensor nodes. In the multi-fusion model, a group header averages samples reported by its group members and determines the detection of events of interest. The lower bound of the number of sensors for a group-determination of the
detection is specified by the model of the particular sensor on board. Second, grid size is also affected by communication range. In our protocol, a group header collects the location information of all sensors and notifies mobile sensors the moving directions. If the grid size is too large, and sensors far away from each other are put into one group, the communication overhead will be large. Given the grid size, the number of static and mobile sensors in the network, and the initial distribution of static sensors (note that mobile sensors can move), a rough number of sensors in one group after the initialization phase can be calculated. If the number is smaller than the lower bound determined by the sensing model, the grid size needs to be adjusted proportionally. Third, the grid size is affected by the speed of mobile sensors. Ideally, mobile sensors can reach the target locations calculated by their group header. If the grid size determined by the previous factor is too large, it needs to be adjusted to be smaller accordingly considering the distance that can be reached by a mobile sensor in one round. Certainly the prerequisite, which is the first condition required by the sensing model, is satisfied.

Parameters $a$ and $b$ together determine how the network is grided and how sensors are divided into groups in different rounds. Also they determine how dynamic the network topology changes are. The setting of $a$ and $b$ is different in the start period and the normal running period of the network. In the start period, the distribution of sensors may be very irregular and the primary purpose of the protocol is to push the mobile sensors to the least covered area to reach a relatively even distribution as soon as possible. In this period, $a$ can be set lower and thus the grid changes are larger; $b$ can be set lower and thus the grid changes are more frequently. In the normal operational time of the network, the primary purpose of the protocol is to generate dynamics. $a$ and $b$ are determined to generate the dynamics with low cost. We experimentally determined that $a = 4$ and $b = 2$ provide good results.

4. Performance evaluation

4.1. Simulation methodology

The protocol is implemented on MATLAB. The objectives of conducting the performance evaluation are threefold: first, testing the effectiveness of our protocol...
in detecting intruders; second, comparing our protocol with related work; third, determining the best system parameters $\alpha$ and $\beta$; finally, studying the performance of our protocol under different conditions, including different initial sensor distribution, the maximum speed of mobile sensors, and sensor failure.

We analyze the performance of our protocol from two aspects; detection probability and moving distance of mobile sensors. The detection probability represents the performance of the scheme and the moving distance of mobile sensors represents the cost.

To compare with existing research work, we also implement two other schemes: VFA in [33] and CAM in [14], as a representative of related work, since they are most close to our work. Both the algorithms focus on maximizing sensor coverage and reducing detection delay.

In the simulation, 20 static sensors and 10 mobile sensors are deployed in a 80 m x 50 m rectangular field. This field is discretized into 1 m x 1 m square cells. The speed of intruder $v_m$ is smaller than or equal to 5 cell/time unit. The maximum speed of mobile sensors is $v_m$, which is 3 cell/time unit except in the section to evaluate the impact of sensor speed. The size of a grid is 4 m x 4 m. $\alpha$ is set to four and $\beta$ is set to two except in the section where the two parameters are evaluated. The simulation results are the average of 10 experiments which are on different initial random deployments except in the section to evaluate the impact of initial deployment on the performance.

4.2. Detection performance and cost

To test the effectiveness of our motion planning strategy, we compare it with four other strategies: first, deploying the same number of mobile and static sensors and letting mobile sensors follow random movement to generate dynamics; second, with the same initial distribution of mobile and static sensors, deploying mobile sensors using VFA algorithm; with the same initial distribution of mobile and static sensors, letting mobile sensors move as directed by CAM; deploying a pure static sensor network with a lot more sensors.

To compare with the first strategy, both detection probability and moving distance are compared since in both our protocol and random movement, mobile sensors move constantly to generate network dynamics. The result is shown in Fig. 8. From Fig. 8(a), we can see that our protocol can effectively achieve a detection probability over 80% while random movement can only achieve about 62%. At the cost side, the total moving distance in our protocol is much shorter than the random movement. For example, when $t = 17$, the moving distance of our protocol is about 9 times shorter than the random movement.

We compare the detection probability of our protocol with two other schemes CAM and VFA, which deploy mobile sensors to maximize coverage for a better detection of events. We do not compare the moving distance since mobile sensors in the two schemes stop moving after the deployment phase.

Fig. 9 shows the sensor distribution after deployment using the two schemes. In the figures, the black dot circles represent mobile sensors and the black circles represent static sensors. The Voronoi diagram is also plotted in the figure, and the black straight line represents the MEP. Fig. 10 is the detection probability of the three different types of intruders under the three algorithms. From Fig. 10, it can be clearly observed that our scheme achieves a much higher detection probability than the other two for all the three kinds of intruders. Note that the comparison is not to show VFA and CAM are not effective in maximizing coverage. On the contrary, they achieve their design goals successfully. The comparison is to show the effectiveness of our protocol in increasing the detection probability and thus it shows the protocol has accomplished our design goals through generating network dynamics.

4.3. Fault tolerance

After running for a long time, sensors may fail due to energy depletion or physical damages, especially when mobile sensors are moving. So this section studies the fault tolerance of our motion planning scheme. 10 mobile sensors and 20 static sensors are deployed randomly in the network. Among the 10 mobile sensors, one mobile sensor
is randomly picked and eliminated from the network for each round and algorithm is re-run to collect data. This procedure is repeated one by one until the number of mobile sensors decreases from 10 to 7. The detection probability and moving distances under different number of sensors are plotted in Fig. 11(a) and (b). It is clear to see that the detection probability keeps the same for all the intruders, which shows that our scheme is fault tolerant. Correspondingly, the average moving distance of each mobile sensor increases from 27 to 43 as a cost.

Fig. 12 is the comparison among the three schemes under different types of intruders. It is clear to see our scheme is more fault tolerant than the other two schemes. The detection probability of the other two schemes drops significantly as mobile sensors die.

4.4. Impact of mobile sensors’ maximum speed

To evaluate the impact of mobile sensors’ speed on the performance and cost of our motion planning...
scheme, the maximum speed of mobile sensors is varied from 1 cell/unit to 10 cell/unit. The simulated results are shown in Fig. 13(a). The figure shows that total detection probability increases very slightly as the maximum speed of mobile sensors increases from 1 cell/unit to 10 cell/unit.
As shown in Fig. 13(b), if sensors always move at full speed, the total moving distance of mobile sensors increases from 130 m to 860 m as the maximum speed increases 10 times from 1 cell/unit to 10 cell/unit. This gives us a hint that an appropriate speed setting for mobile sensors can get a good balance between the detection probability and energy conservation even though sensors can move much faster.

4.5. Impact of sensor distribution

In this section, we evaluate the impact of sensor distribution on the performance of our scheme. We adopt Normal distribution to distribute the sensors. Mean of the distribution $\mu$ is set to be the center of the area. The standard deviation $\sigma$ varies from a minimum value of 1/10 of area dimension to the dimension with an increase step of 1/10.

We first study the impact of mobile sensor distribution. The results are shown in Fig. 14. The variance increases from left to right. When $x$-value $= i$, $\sigma$ is $i/10$ of the area dimension. From Fig. 14(a), we can see that the detection probability keeps almost the same and is not affected by the mobile sensors distribution. The moving distance of mobile sensors, shown in Fig. 14(b), decreases as the initial distribution of mobile sensors becomes less concentrated in the area.

Then we study the impact of static sensor distribution. The results are shown in Fig. 15. From the figure, it is clear to see the detection probability does not change much as the distribution of static sensors becomes even. It is very interesting that the detection probability of random intruders is higher when the distribution of static sensors is more dense in the center. After that, the detection probability for all intruders keeps almost the same. Total moving distance shown in Fig. 15(b) decreases as the static sensors become more evenly distributed. The reason is that as the distribution of static sensors becomes even, there are less coverage holes and mobile sensors have less pressure to move to heal the holes.

![Fig. 14. Different distributions of mobile sensors.](image)

![Fig. 15. Different distributions of static sensors.](image)
4.6. Impact of $a$ and $b$

In this section, we evaluate the impact of parameters $a$ and $b$ on the performance of our scheme. In the simulation, with a fixed $b$ value 2, $a$ varies from 2 to 8. With a fixed value of $a$ value 4, $b$ varies from 1 to 5. The results are shown in Figs. 16 and 17. In terms of detection probability, Figs. 16(a) and 17(a) show that the detection probability keeps almost the same and is not affected by either $a$ or $b$ much.

On the contrary, the moving distance of mobile sensors varies with different $a$ and $b$ as shown in Figs. 16(b) and 17(b). In Fig. 16(b), the moving distance has the least value when $b = 2$. In Fig. 17(b), the moving distance has the least value when $b = 4$. This means under the random initial deployment, the optimal value of $a$ and $b$ is 4 and 2 to render the best system performance.

5. Related work

Intrusion detection is one of the most important applications in WSNs. To achieve higher detection performance with a smaller number of sensors, Minimum Exposure Path (MEP) is well studied for designing good topology of the sensor deployment in [1,25]. A centralized algorithm is proposed in [25] to find the MEP in a network given network topology. An optimal framework for determining sensor positions is proposed in [1] to detect mobile target traversing a given area. Coverage properties are studied in [20] as a baseline for how well a monitored area is covered. A system for event detection based on distributed pattern recognition algorithms in WSNs is presented in [28]. The system is fully trainable to recognize different classes of application-specific events. Luo et al. present a solution for intrusion detection of ships in [18]. Using signal
processing techniques and cooperative signal processing techniques, the passing ships can be detected by distinguishing the ship-generated waves and the ocean waves. Prior works using static sensors to detect mobile target are presented in [4,13,15,16,24,26], which mainly focus on effectively detecting the presence of an intruder. Lin et al. [16] develop a logic object-tracking tree to reduce the communication cost. Virtual sensor is introduced in [26] resulting from neighboring sensors’ collaboration based on value fusion, which can improve the coverage performance. Kumar et al. develop solutions for the case when wireless sensors are deployed to form an impenetrable barrier for detecting movements in [13]. The concept of one-way barrier coverage is introduced in [4], which requires that the network reports illegal intruders with certain directions of the movement. Li et al. study the barrier’s detection probability under probabilistic sensing model in [15]. Jie et al. propose a barrier scheduling algorithm to effectively detect intruders and prolong the network lifetime at the same time in [24]. However, these activities focused on the fixed network topology constructed only by the static sensors, where the MEP, the bottleneck of the field coverage, is fixed and can be easily found by an intruder using proposed work.

There are many studies that utilize mobile sensors to assist or improve the target detection performance. The VFA algorithm in [33] attempts to maximize the sensor field coverage by a given number of stationary and mobile sensors. The CAM algorithm in [14] uses multiple stationary and mobile sensor nodes which collaborate to improve the area coverage and to detect events of interest earlier. Bisnik et al. study the quality metric of the stochastic event detection using mobile sensors, and propose the required speed and the number of mobile sensors for the bounded event loss probability in [3]. Combining both static and mobile sensors, multi-fusion model is adopted and reaction-based movement is presented in [29] extended from [21]. In this scheme, the authors propose a two-phase detection and use the reaction-based movement to make the mobile sensors collaborate with static sensors to decrease the moving distance and the false positive value of detection.

Another technique to detect intruders is barrier technique [8,12,19]. In [12], mobile sensors are used to provide k-barrier coverage against moving intruders. He et al. [8] use mobile sensors to achieve barrier coverage in sensor scarcity case by dynamic sensor patrolling. Saipulla et al. study how to efficiently improve barrier coverage using mobile sensors with limited mobility in [19]. However, MEP is not considered in the barrier coverage [19]. A 0/1 sensing model is adopted in barriers construction to cover all crossing paths. It is assumed that the intruder can be detected once the intruder crosses the barrier. However, such model is simpler than MEP and it requires a large number of sensors in barrier construction [19] to achieve a satisfied detection performance.

Detection probability and network connectivity are studied in [27] for both homogeneous and heterogeneous WSN, where the authors evaluate the necessary conditions for ensuring the detection probability in single-sensing detection and multiple-sensing detection model. Fang et al. [7] aim to find the most valuable event areas among all the event areas to monitor, subject to resource constraints in sensors. An online integer linear programming algorithm is proposed to solve the problem. Kapnadak and Coyle [10] study the problem of determining the optimal spatial node density for deployment of a Wireless Sensor Network (WSN) for distributed detection of a randomly located target in a sensing field.

6. Conclusion

In this paper, we study the motion planning of mobile sensors in a sensor network composed of both mobile and static sensors. The simulation results show that our motion planning scheme can increase the detection probability while keep a low moving distance for mobile sensors. This scheme is good at fault tolerance. The desired detection probability is maintained at a cost of more movement of mobile sensors when sensors fail.

Acknowledgment

This work is supported by the National Science Foundation Grants NSF-1128369.

References


Jie Tian received his BS degree in Computer Science, Department of Computer Science from Tianjin University, Tianjin, China, in 2005, and got his MS degree in Computer Science from Department of Computer Science at Nankai University, Tianjin, China, in 2008. He is currently a PhD candidate in Department of Computer Science at New Jersey Institute of Technology. His research includes wireless networks, ad hoc/sensor network and mobile computing.

Guiling (Grace) Wang received the BS degree from Nankai University, China. She received the PhD degree in computer science and engineering and a minor in statistics from Pennsylvania State University in May 2006. She joined the New Jersey Institute of Technology as an assistant professor after that. Her research interests include distributed systems, wireless networks, and mobile computing with a focus on wireless sensor networks. She is a member of the IEEE.

Tan Yan received his BE in Electrical Engineering, School of Information Science and Technology from Southeast University, Nanjing, China, in 2007, and got his ME in Electrical Engineering from ECE department at New Jersey Institute of Technology, New Jersey, USA in 2008. He is currently a PhD candidate in Department of Computer Science at New Jersey Institute of Technology. His primary research interests are Wireless ad hoc/sensor networks and Mobile computing.

Wensheng Zhang received the BS degree from Tongji University, Shanghai, China, and the MS degree from Chinese Academy of Sciences. He received the PhD degree in computer science from the Pennsylvania State University in 2005. Since then, he has been with the Department of Computer Science at Iowa State University as an assistant professor. His research interests are wireless networks and network security. His research is sponsored by US National Science Foundation (NSF). He is a member of the IEEE and the ACM.