Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2011, Article ID 724136, 17 pages
doi:10.1155/2011/724136

Research Article

Distributed and Collaborative Node Mobility Management for Dynamic Coverage Improvement in Hybrid Sensor Networks

Thakshila Wimalajeewa[1] and Sudharman K. Jayaweera[2]

[1] *Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244, USA*
[2] *Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131, USA*

Correspondence should be addressed to Thakshila Wimalajeewa, twwewelw@syr.edu

Received 25 April 2010; Revised 15 January 2011; Accepted 4 February 2011

Academic Editor: Amiya Nayak

With recent advances in deploying sensor nodes mounted on mobile platforms, node mobility is becoming an attractive alternative to improve network coverage dynamically in sensor networks. However, due to energy constraints, it may not be cost effective to deploy a large number of mobile nodes for continuous movements. It might be more desirable to allow only a certain number of nodes to be mobile depending on the affordable cost and desired performance levels. This paper proposes an efficient distributed mobility protocol for mobile node navigation in a hybrid sensor network consisting of both static and mobile nodes to provide efficient time-varying coverage after the initial deployment. In the proposed scheme, mobile nodes collaborate with neighboring static nodes to find their candidate locations to move at each movement step in order to maximize the *coverage time* of the area not covered by static nodes. We also develop an efficient sequential algorithm to find the *exposure* in a hybrid network, which reflects the best path for a target to traverse the sensing region without being detected. By simulations, we show the effectiveness of the proposed mobility protocol in terms of the presence probability matrix and *coverage time* and show its suitability at the worst-case target *exposure*.

1. Introduction

Mobile sensor nodes are deployed in wireless sensor networks in certain applications to enhance the network performance dynamically. Use of node mobility to reposition sensors at the deployment stage to provide a uniform coverage was considered in [1–5], based on different techniques. However, these studies do not consider how to exploit the node mobility in possible performance improvement after the initial deployment stage. Liu et al. in [6] showed that the coverage can be improved dynamically by allowing nodes to be mobile continuously in a mobile sensor network over time unlike in a static network. Distributed detection and tracking tasks by mobile sensor networks consisting only mobile nodes were addressed by some recent work. In [7], the problem of target detection using a mobile sensor network is addressed where the authors analyzed the detection latency. In [8], algorithms to find upper and lower bounds for the target *exposure*, which is defined as the target traversal which results in the worst-case detection performance, in a mobile sensor network deployed for mobile target detection were proposed. In [9], a cat-and-mouse game between targets and mobile nodes was presented based on the sensing capabilities of targets and mobile nodes where mobile nodes try to detect the target as quickly as possible when the target is trying to evade the network before being detected. In [10, 11], distributed tracking by mobile sensor networks is addressed.

However, deploying a large number of mobile sensor nodes is not as cost effective as deploying static nodes in a sensor network due to energy constraints. Thus, it is desirable to allow only a fraction of total nodes to be mobile to improve the network performance depending on application requirements. Use of hybrid sensor networks consisting of both static and mobile nodes is becoming attractive in current sensor network applications. These hybrid networks provide a better tradeoff between the cost of mobile node deployment and the required performance levels. In [12, 13], algorithms for reposition of mobile nodes at the initial deployment stage are developed in hybrid sensor networks. In [12], mobile nodes are directed to move towards the coverage holes detected by static nodes to improve the coverage. In [13], impact of the node density to provide k-coverage at the deployment stage in a hybrid sensor network is discussed. In these approaches, it was assumed that the mobile nodes move only once during the deployment stage and remain stationary while the sensor network performs specific operations. In [14], mobile node navigation towards a specific goal in a hybrid sensor network is addressed where static nodes are used to guide the mobile nodes. Distributed detection by hybrid sensor networks is also addressed in recent works [15, 16] when the sensor node and target positions are known. Target tracking performance of an integrated mobile-static sensor network is addressed in [17] where the mobile nodes are used to aid the data propagation when the communication ranges of static nodes are limited. However, neither of the above works addressed the problem of how to efficiently cover the uncovered area by static nodes in a hybrid sensor network dynamically, by node mobility over time to provide an efficient time-varying coverage.

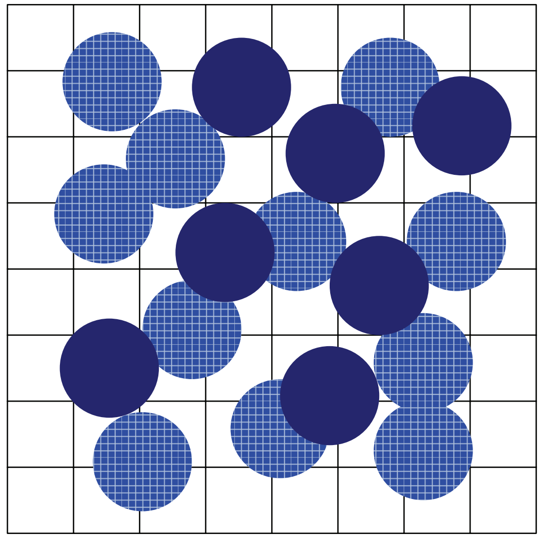


Figure 1: Hybrid sensor network consisting of both static and mobile nodes: solid circles-mobile nodes and checked circles-static nodes.

2. Motivation, Contribution, and Organization

Consider a hybrid sensor network deployed in a square region as shown in Figure 1, where the union of checked circles represents the area covered by static nodes while the union of solid circles represents the area covered by mobile nodes, respectively. When the nodes are first deployed in a region, a random placement is often desirable especially when *a priori* knowledge of the terrain is unavailable. However, such random deployment strategies may not result in effective coverage always, since some nodes might be overly clustered while some of them might be sparsely located. Use of node mobility to reconfigure the node locations to improve the coverage of such networks was addressed by some authors, for example, in [1, 2, 12]. In these approaches, nodes move only during the deployment stage and the maximum coverage area achieved by the network after reconfiguration is limited by the number of total nodes and nodes' sensing ranges. For example, if the total number of nodes is relatively small, even by reconfiguration of mobile nodes to provide a uniform coverage, a large portion of the network may remain not covered. On the other hand, node failures after the initial reconfiguration might cause coverage holes in the network. Thus, the problem addressed in this paper is how to effectively use the node mobility of mobile nodes to provide an efficient dynamic coverage of the region of interest after the initial deployment stage.

Exploiting mobile nodes for continuous coverage in mobile sensor networks is addressed by [6] when the nodes perform random and independent mobility. Although random mobility models are desirable in many applications, and they need minimum coordinations among nodes, they may not always be ideal for hybrid networks consisting of both static and mobile nodes. We need to consider the following factors in designing an algorithm for mobile node navigation in a hybrid sensor network to provide efficient dynamic coverage.

(i) In a hybrid sensor network, a certain portion of the field is covered always (as shown by the union of checked circles in Figure 1) as mentioned before. Mobile nodes are required to assist providing the coverage for the area that is not covered by static nodes. If a random and independent mobility scheme is used, there might be overlappings of the sensing ranges of mobile and static nodes since there is no coordination among nodes. In many real world applications, a mobile node (a sensor node mounted on a mobile platform,) has a fixed power cost for the mobility. Even though sensor nodes mounted on mobile platforms can carry more battery supplies to move a considerable amount of time/distance continuously, it is important to ensure that the available energy is effectively used to perform the required surveillance task, that is, to provide an effective time-varying coverage in the desired field in a given duration of time. Thus, it is required to use mobile nodes to cover only the areas uncovered by static nodes minimizing the overlapping between the mobile and static nodes' sensing ranges.

(ii) When nodes are mobile, previously covered areas by mobile nodes become uncovered while uncovered areas become covered. This requires to manage the mobility of the mobile nodes such that to minimize the duration that a particular location is uncovered. Random mobility schemes do not address these issues.

(iii) If the network does not have any prior knowledge about the sensing field, it is desired that any point not covered by the static nodes is covered almost equally to maintain an approximately uniform coverage over time.

Taking these factors into account, in this paper, we propose a new distributed mobility protocol for mobile node navigation in a hybrid sensor network. In the proposed scheme, collaborating with static nodes, mobile nodes provide an efficient dynamic coverage in the area not covered by the static nodes. More specifically, we assume that the sensor network is partitioned into square cells such that a node can cover such a cell completely when it is located at the center of the cell. We divide these cells into two categories: *static* and *void* cells. *Static* cells correspond to the cells in which there is at least one static node, and the *void* cells are the ones in which there is no any static node. Mobile nodes are directed to move among these *void* cells based on a certain criteria. Each of such *void* cells is given a certain *base price*. This *base price* is updated by static nodes based on the time that the *void* cell remains not covered by at least one mobile node. At each movement step, mobile nodes communicate with their closest static nodes locally to search for *void* cells which are not covered for a long time. Static nodes provide necessary information for mobile nodes in their neighborhoods. At a given time, we assume that a mobile node can visit a certain number of candidate *void* cells from its current position. These candidate *void* cells are determined by the mobile node's maximum speed. Taking base prices (collected from neighboring static nodes) of the candidate *void* cells into account, each mobile node selects the best *void* cell to be visited by the next time step. In the proposed scheme, since the node mobility is performed by mobile nodes by collaborating with static nodes, we call the proposed scheme "*mobile-static collaborative mobility model*." In simulations, we show the effectiveness of the *mobile-static collaborative mobility model* in terms of the presence probability matrix and the average time that an arbitrary point in the network is not covered. (The presence probability matrix contains the probabilities of the presence of at least one node at each cell at any given time instant.)

We further analyze the effectiveness of the proposed mobility scheme in terms of the worst-case detection performance when the network is deployed for detection applications. It is noted that when the application requirement is different, there are other performance measures that can be selected (depending on the type of application) to evaluate the effectiveness of the proposed mobility model. However, in the paper, we restrict ourselves only to a target detection application which is one of the fundamental tasks performed by a sensor network. We analyze the worst-case detection performance in terms of the *exposure* [8, 18, 19], which reflects the quality of the sensor network when the target tries to evade the network with minimum probability of being detected. To find the *exposure*, we develop an efficient sequential methodology based on the presence probability matrix. The proposed methodology to find the *exposure* is valid for hybrid sensor networks with arbitrary mobility models as far as the knowledge of the presence probability matrix is available. We show that the proposed mobility scheme results in a significant performance improvement at the worst-case target *exposure* compared to that with random mobility schemes especially when the fraction of mobile nodes in the hybrid network is small.

The paper is organized as follows: Section 3 presents the network model and the assumptions. In Section 4, the proposed *mobile-static collaborative mobility model* is described in detail. The worst-case performance on target detection by the hybrid sensor network with proposed mobility protocol is addressed in Section 5. Performance results are shown in Section 6, while the concluding remarks are given in Section 7.

3. Network Model and Assumptions

We consider a hybrid sensor network made of N number of sensor nodes deployed in a region $\mathcal{R}$ with network dimension of $b \times b$. Out of N, that there are N_s number of static nodes and N_m number of mobile nodes. Denote $\lambda = N/b^2$ to be the spatial density of the nodes and $\lambda_m = N_m/N$ and $\lambda_s = N_s/N$ to be the fractions of mobile and static nodes, respectively. Let $\mathcal{V}$, $\mathcal{V}_m$, and $\mathcal{V}_s$ be the sets containing all mobile and static node indices, respectively.

Suppose that the sensing region is divided into a virtual square grid with grid length of $l = \sqrt{2}r$ where r is the effective sensing radius of a sensor. We assume that both static and mobile nodes have the same sensing radii. When a sensor node is located at the center of a cell in the grid, the cell is completely covered by the corresponding sensor node. Consider the hybrid network with only static nodes as shown in Figure 2 (dropping the mobile nodes in Figure 1). We denote the cells that are not covered by the static nodes as *void* cells (with void squares as shown in Figure 2). When a static node is located in a particular cell (crossed cell in Figure 2), we consider that the corresponding cell is covered by the relevant static node and call it a *static* cell. However, note that since a static node is not necessarily located at the middle of a cell, corresponding cell may not be completely covered by corresponding the static node. We address this problem later and for the moment assume that the cell is covered by the corresponding static node. Now, the problem is how to use the mobile nodes efficiently to cover the *void* cells as shown in Figure 2 over time, such that the revisiting time of any cell by at least one mobile node is maximized. In the following, we propose a new distributed interactive protocol, called *mobile-static collaborative mobility model* to achieve the required task by collaboration among mobile and static nodes.

In the following, we list the specific assumptions made in the proposed mobility algorithm.

Assumptions. (1) All nodes have the same sensing radius.

(2) There is a fraction λ_m of mobile nodes having enough locomotion energy to provide dynamic coverage in a time duration of $\widetilde{T}$ where $\widetilde{T}$ is determined by several factors, such as the maximum distance that a mobile node can move before the energy is depleted, and application requirements. This assumption is realistic for relatively large $\widetilde{T}$ since sensor nodes mounted on mobile platforms can carry more battery supplies.

(3) λ_m remains constant during the time interval $\widetilde{T}$.

(4) We consider an obstacle-free environment.

(5) Static sensor network is assumed to be connected within the time duration $\widetilde{T}$.

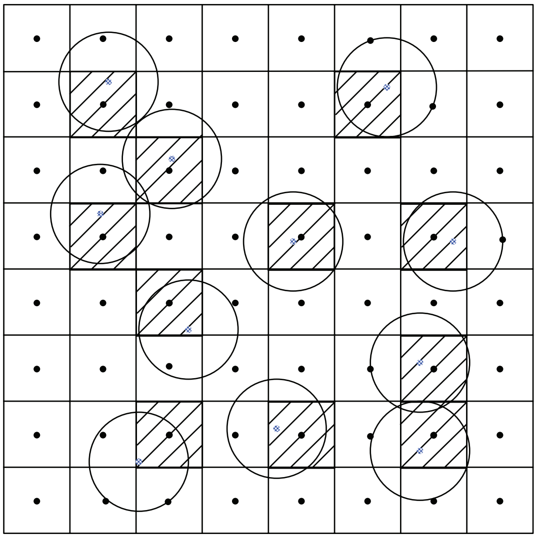


Figure 2: Sensor network with only static nodes.

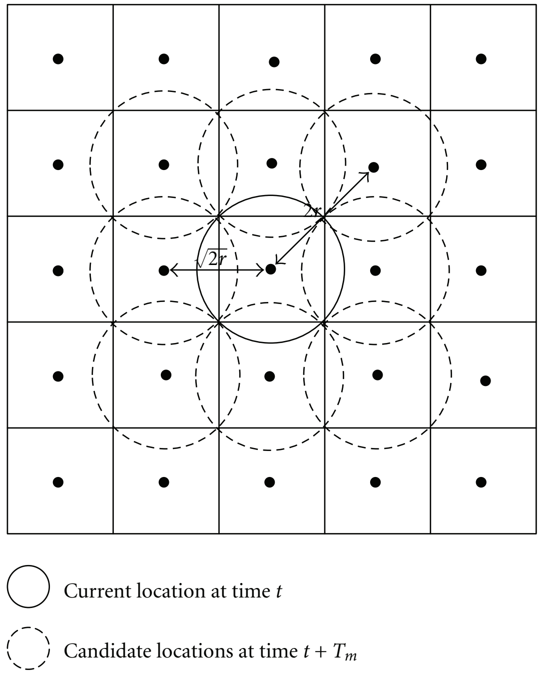


Figure 3: A mobile node's candidate locations at a given time.

For applications where these assumptions are not satisfied, possible modifications to the algorithm are discussed at the end of the Section 4.

4. Distributed Mobility Protocol

In this section, the proposed *mobile-static collaborative mobility model* is discussed in detail.

4.1. Description of the Algorithm. Once identifying the *static* and *void* cells, we assign a base price for each *void* cell according to the following rule. Initially, at time $t = 0$, we assign a base price $\mathcal{P} = 0$ for each *void* cell in which there is at least one mobile node. For all the other *void* cells, we assign $\mathcal{P} = K$ where K is a large value. Let T_m be the time step in which the mobility management is performed, which can be determined as given below.

4.1.1. Determining T_m. We assume that any mobile node can reach $L_c = 8$ number of closest distinct cell centers (and itself) as shown in Figure 3 at any given time step. Then the maximum distant that a node has to move during time T_m is $2r$. Thus, it is desirable to choose the time step T_m as $T_m = \lceil(2r/v_{\max})+\epsilon\rceil s$ where ϵ is a bias factor which accounts for the scenarios when it is needed to heal the lack of coverage at *static* cells which will be explained in Section 4.4 in detail.

At each time step T_m, the base price of each *void* cell is updated considering the time it remains uncovered (or unvisited by at least one mobile node). More specifically, at each step T_m, if a particular cell is visited by a mobile node, its base price $\mathcal{P}$ is set to zero and the base prices of all other *void* cells are increased by 1 unit. Without loss of generality, we assume that at time $t = 0$ each mobile node has moved to the cell center which it belongs to, and at each step T_m, mobile nodes move among cell centers. In the following, we explain how a mobile node selects the best cell to be visited at each time step distributively by collaborating with static nodes.

Let each cell (cell center) in the square grid be given an ID labeled by indices $1, 2, \ldots, L_T$ where $L_T \approx b^2/l^2$ is the total number of cells. Let there be L_s number of cells covered by static nodes (*static* cells) and $L_v = L_T - L_s$ number of cells that are not covered by static nodes (*void* cells). Also denote $\mathcal{U}$, $\mathcal{U}_s$, and $\mathcal{U}_v$ to be the sets containing all cell indices of the network, *static* cell indices and *void* cell indices, respectively.

4.1.2. Assigning Void Cells for Each Static Node. We assign a certain number of *void* cells to each static node in the network. Each static node in the network is responsible for updating the base price of each *void* cell that belongs to it. Corresponding *void* cells for each static node are assigned based on Voronoi partitions (as shown in Figure 4). According to Voronoi partitions, any point inside a Voronoi polygon of a static node is closer to that static node rather than to any other static node in the network. Thus, for a given static node s_k, the cell centers belonging to its Voronoi polygon are closer to the static node s_k than any other static node in the network. We assume that each static node has the knowledge of the positions of the *void* cell centers belonging to itself. At the initial stage, static nodes can communicate with their Voronoi neighbors locally to construct Voronoi polygons. It is noted that each static node needs to know only the existence of its Voronoi neighbors and communicate among them locally to construct the Voronoi polygon. By knowing its own location, and based on the grid length (in terms of the sensing range), each static node can determine the *void* cells in its Voronoi polygon. Since we assume that the static nodes are connected during the time $\widetilde{T}$ in which the node mobility is performed, the *void* cells belong to each static node's Voronoi polygon are always taken care of at each

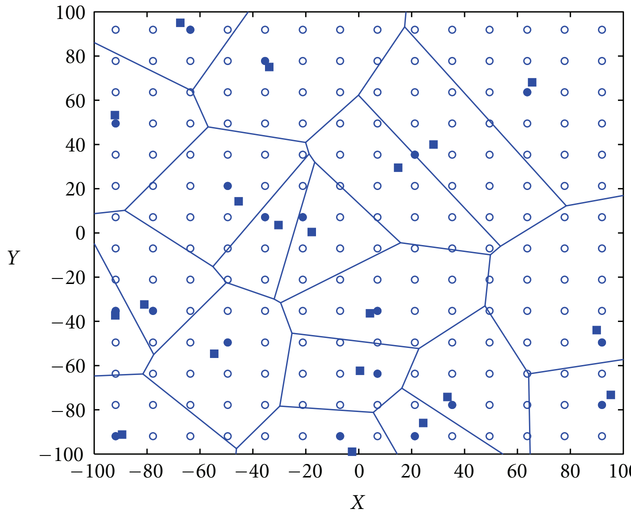


FIGURE 4: Voronoi polygons for each static node: Solid square-static node locations, solid circles-grid points (centers) corresponding to static nodes and void circles-grid points (centers) corresponding to grids not covered by static nodes.

time step. In the proposed algorithm, it is assumed that any *void* cell inside a Voronoi polygon can communicate with at least the corresponding static node of that Voronoi polygon. Since any mobile node is assumed to be located in a *void* cell, and each *void* cell is assumed to belong to a Voronoi polygon of a particular static node, it is assumed that each mobile node can communicate at least with the corresponding static node in that Voronoi polygon.

Denote $\mathcal{U}_{s_k}$ to be the set of *void* cell indices belonging to the Voronoi polygon of the static node s_k for $s_k \in \mathcal{V}_s$ and $L_{s_k} = |\mathcal{U}_{s_k}|$ be the number of *void* cells (cell centers) belongs to static node s_k. Note that we have then $\mathcal{U}_v = \bigcup_{k\in\mathcal{V}_s} \mathcal{U}_{s_k}$. Further denote $\mathbf{g}_{s_k}(nT_m)$ to be an L_{s_k}-length vector containing the base prices for all *void* cells attached to the static node s_k at time nT_m for $s_k \in \mathcal{V}_s$. Each static node s_k is responsible for updating $\mathbf{g}_{s_k}(nT_m)$ at each time step $t = nT_m$ for $n = 1, 2, \ldots$.

4.2. *Updating* $\mathbf{g}_{s_k}(nT_m)$

4.2.1. At Time $t = 0$. At time $t = 0$, each mobile node broadcasts its current location (or equivalently current cell ID) to its neighborhood, such that static nodes located close to the corresponding mobile node receive this information. If the corresponding mobile node's cell ID belongs to $\mathcal{U}_{s_k}$, then the static node s_k sets the base price for the corresponding cell to zero. Base prices for all the other cells in $\mathcal{U}_{s_k}$ are set to a large integer number K. Note that at time $t = 0$, all *void* cells which have no mobile node at time $t = 0$ have the same base price K.

4.2.2. At time $t = nT_m$, $n \geq 1$. At time $t = nT_m$, each mobile node broadcasts its location information (current cell ID) to its nearest static nodes. Let $N_{m,k}(nT_m)$ be the number of mobile nodes that the static node s_k receives location information at time nT_m and $\mathcal{U}_{m,k}(nT_m)$ be the set corresponding to those locations (cell indices). Then for a given static node s_k for all cell indices $c_j \in \mathcal{U}_{s_k}$, it checks whether c_j also belongs to $\mathcal{U}_{m,k}(nT_m)$. If $c_j \in U_{s_k} \cap \mathcal{U}_{m,k}(nT_m)$, the static node s_k sets the base price of the cell c_j to be zero. Otherwise, it increases the base price of the cell c_j by 1 unit.

After updating the base price vector $\mathbf{g}_{s_k}(nT_m)$ at time nT_m at each static node s_k, the problem is to determine the next cell ID to be visited by each mobile node by time $t = (n+1)T_m$, such that the cell-revisiting time is maximized. Denote $\mathcal{C}_{m,j}(nT_m)$ to be the set of candidate locations (cells) of the jth mobile node at time nT_m. Also let $\mathcal{U}_{s_k}^{m_j}(nT_m)$ be the set of cell indices belonging to both $\mathcal{C}_{m,j}(nT_m)$ and $\mathcal{U}_{s_k}$. Note that the maximum size of the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$ is $|\mathcal{U}_{s_k}^{m_j}(nT_m)|_{\max} = L_c + 1 = 9$, since we assume that each mobile node can move to one of the 8 distinct candidate locations and itself during a given time step. For a given mobile node m_j from which the static node s_k receives the location information, the static node s_k checks whether any cell in m_jth candidate set $\mathcal{C}_{m,j}(nT_m)$ belongs to $\mathcal{U}_{s_k}$ at time $t = nT_m$. If not, static node s_k does not need to communicate with mobile node m_j at time nT_m.

If any cell in m_jth candidate set $\mathcal{C}_{m,j}(nT_m)$ belongs to $\mathcal{U}_{s_k}$, or in other words, if the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$ is not empty, the communication between the static node s_k and the mobile node m_j is performed as follows.

(i) Based on the information received by closest mobile nodes, the static node s_k determines whether there are more than two mobile nodes located within a distance d_t. We say the mobile node m_j is *isolated* with respect to another mobile node, if there is no at least one mobile node within a distance d_t from its current location where d_t (equals to $4r$) is a threshold distance which is determined such that no duplicate covering occurs as discussed in Section 4.3. If the mobile node m_j is not *isolated* with respect to another mobile node, there is a possibility for a duplicate covering; that is, two or more mobile nodes try to cover the same cell at the time $(n+1)T_m$. Note that in the rest of the paper a mobile node is *isolated* means that the mobile node is *isolated* with respect to another mobile node. It is noted that (as one reviewer pointed out), if the duplicate covering is going to happen, the same static node is responsible for updating the base price of the corresponding cell (the cell that both mobile nodes are going to cover). Thus, if the static node s_k identifies that there are more mobile nodes within a distance of d_t to each other, it transmits all the base prices corresponding to the candidate locations in the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$ to assist in resolving the duplicate covering problem as discussed in Section 4.3. In this case, the mobile node m_j selects the best cell to be moved by time $(n+1)T_m$ after checking the need for duplicate covering by locally communicating with neighboring mobile nodes. This scenario is further discussed in Section 4.3.

(ii) If m_j is *isolated* (that is there is no any other mobile node within a distance of d_t from the current location of m_j), static node s_k finds the cell from the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$ which has the maximum base price and sends a message corresponding to the cell ID and the maximum corresponding base price. Note that all the candidate cells for mobile node m_j may not belong to a one static node. In particular, they may belong to multiple nearby static nodes. Once the mobile node m_j gets maximum base prices from multiple static nodes which its candidate cells belong to, it selects the best location for time $(n+1)T_m$ by comparing the base prices it gets from different static nodes and selects the one with maximum base price. Note that if there are two or more candidate cells with the same highest base price for a mobile node, it selects the candidate cell randomly from those.

It is worth mentioning that if the mobile node m_j is *isolated*, the static node s_k sends only one base price and cell ID to the mobile node m_j (which is corresponding to the maximum base price in the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$). On the other hand, if m_j is not *isolated*, the static node s_k has to send all base prices and cell IDs in the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$ (which has 9 cells in the worst case).

4.3. Duplicate Covering at a Given Time. As mentioned before, when two mobile nodes are close to each other, there might be situations where both will try to select the same *void* cell as the candidate location based on the values of corresponding base prices. For example, consider the scenario as depicted in Figure 5. Assume that two mobile nodes m_1 and m_2 are located in cells represented by A and B at time $t = nT_m$ as shown in Figure 5. According to the information received from closest static nodes, both mobile nodes can access to the base prices of all of their candidate cells, marked at the north-east corner of each candidate cell for both mobile nodes. According to the base prices, both mobile nodes will try to select the cell C as the next location for time $(n+1)T_m$ which has the highest base price from each mobile nodes' candidate sets. It can be shown that this phenomenon might happen only when two mobile nodes are located within a maximum distance of $d_t = 2\sqrt{2}l = 4r$.

Since this will lead to inefficient coverage, we propose for two mobile nodes to exchange their information locally to avoid duplicate covering. Since this phenomenon occurs when two mobile nodes are located close to each other, we assume that these two mobile nodes can exchange their information to check whether a duplicate covering is going to happen. If so, they exchange the next maximum base prices from their candidate sets and check which mobile node has the second maximum base price (Note that when a mobile node is not *isolated*, they have the access for base prices of all candidate cells as discussed above). Accordingly, the node with the second highest maximum base price selects the corresponding cell as the candidate cell. According to Figure 5, since the mobile node m_1 has the second maximum base price (compared to mobile node m_2), it moves to the corresponding cell (denoted by cell D) while the mobile node m_2 moves to the cell C. If the second maximum base price is the same for both nodes, they can select either one of the nodes to move to the cell with the second maximum base price arbitrarily. When there are more than 1 mobile sensor within the distance d_t from node m_j, the same procedure can be extended by exchanging the relevant information among

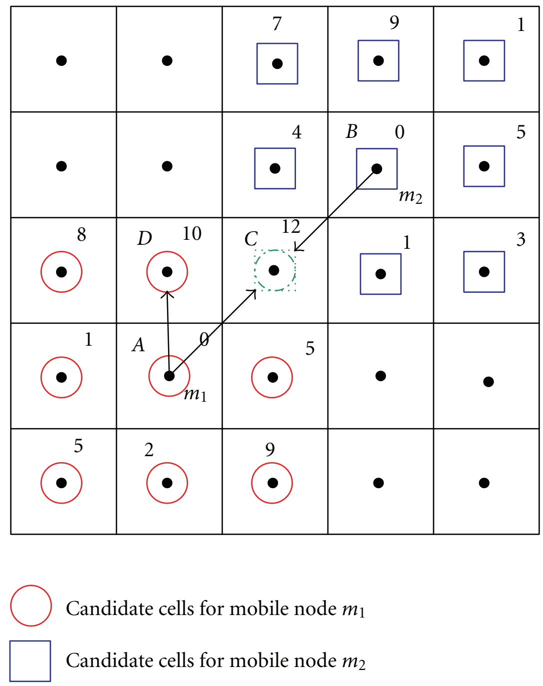


FIGURE 5: Duplicate covering at a given time.

those nodes. In such cases, it might be necessary to exchange 2nd, 3rd,... highest base prices among neighboring mobile nodes.

4.4. Compensating for the Lack of Coverage in a Static Cell. As mentioned earlier in this section, since a static node may not necessarily be located at the center of a *static* cell in the grid, there are certain uncovered portions of the corresponding cell. Note that this uncovered portion is maximized when a static node is located very close to one of the cell corners which it belongs to. Consider the scenario that the static node is located very close to the north-east corner of the cell it belongs to (denoted by c_1), as shown in Figure 6 with a circle with solid line. To compensate for the lack of coverage in the corresponding cell, we propose the following procedure. It can be shown that with the relationship between the side length of a cell in the grid and the sensing range, when a mobile node comes to a cell located either to the left or to the bottom of the *static* cell, and if they are moved a distance of $r - (r/\sqrt{2})$ (at the worst case) beyond the cell center towards the *static* cell, the uncovered portion of the corresponding *static* cell can be completely covered. This is illustrated in Figure 6 where a mobile node comes to either cell center A or C, and if it is allowed to move a distance of $r - (r/\sqrt{2})$ (i.e., either to B or D, resp.), the uncovered portion of the *static* cell can be completely covered. To address this problem, at time nT_m, when a mobile node selects its candidate cell for time $(n+1)T_m$, it also checks whether there is a static node located to the right, left, up, or down to the selected cell. Based on the static node location, it approximates the required distance it should move (maximum of $r - (r/\sqrt{2})$)

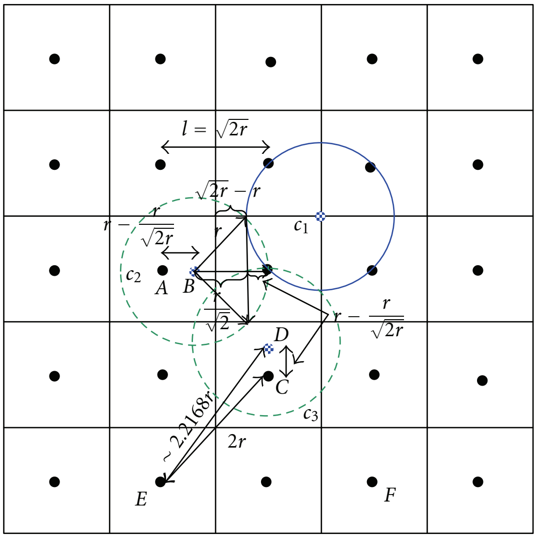


Figure 6: Compensating for the lack of coverage in *static* cells.

beyond the selected cell center to compensate for the lack of coverage of the *static* cell.

Note that according to the proposed mobility algorithm we allow mobile nodes to move between cell centers at consecutive time steps T_m. However, when we need to address this *static* cell compensating problem, mobile nodes have to move little far away from a cell center. When this happens (i.e., a mobile node may move to location B (or D) instead of A (or C) in Figure 6), the mobile node may need to move a maximum distance of $\approx 2.2168r$ to reach its next candidate cell at next time step. As shown in Figure 6, when the mobile node is at the point D in the cell c_3, it can reach all candidate cells by next time step, except E and F, by moving a maximum distance of $2r$. To reach the candidate cells E and F it has to move a maximum distance of $\approx 2.2168r$. Thus, when determining the time step T_m as pointed out in Section 4.1.1, we need to take this scenario into account. Thus, T_m is selected as $T_m = \lceil (2r/v_{\max}) + \epsilon \rceil s$ where $\epsilon = 0.2168r/v_{\max}$.

The proposed *mobile-static collaborative mobility model* for node mobility management of hybrid sensor network is summarized in Algorithm 1.

It is worth mentioning that the Algorithm 1 requires proper time synchronization for its operation. It is assumed that each static node enters the initialization phase by locally communicating among them. This initial synchronization among sensors can be achieved with a similar scheme as presented in [20]. During the initialization period,

(i) all static nodes broadcast their location information locally to construct Voronoi polygon at each static node and to assign the corresponding *void* cells to each static node;

(ii) all static nodes initialize their *base price* vectors;

(iii) static nodes broadcast a message to mobile nodes in their neighborhoods to set the *timers* of mobile nodes to the initialization phase and ask to broadcast their location information locally.

After the initialization phase, it is assumed that static and mobile nodes manage to have time synchronization at each time step T_m via local communication among static and mobile nodes. During each time step T_m, each static and mobile node can enter the different phases on their task schedules as described in Algorithm 1.

4.5. Modifications to the Algorithm When Certain Assumptions Are Relaxed. It should be noted that the algorithm is based on certain assumptions stated in Section 3. In the following, we discuss how the algorithm can be modified when some of these assumptions are relaxed.

In the algorithm, it was assumed homogeneous sensors; that is, each node has identical effective sensing radius. According to the proposed algorithm, the nature of the sensing radius of nodes matters when the grid length of the virtual grid is selected. With homogeneous sensing radius, the grid length is selected as $\sqrt{2}r$, since then when a sensor node lies at the center of a cell, that cell is completely covered by the corresponding node. If nodes have different sensing radii, the algorithm can be modified in following ways. Let $r_{\max}$ and $r_{\min}$ be the maximum and minimum values of sensing radii of nodes.

(i) If $r_{\max} - r_{\min}$ is small: in this case, a simple modification can be employed to the current algorithm. The virtual grid can be constructed such that the grid length equals to $\sqrt{2}r_{\min}$. This ensures that if any node is located at the middle of a cell, the corresponding cell is completely covered. If the grid length is selected as $\sqrt{2}r_{\min}$, it is noted that when $r > r_{\min}$, a certain portions of neighboring cells will also be covered by the corresponding node. However, if the difference $r_{\max} - r_{\min}$ is small, selecting grid length as $\sqrt{2}r_{\min}$ does not cause a large performance degradation with the proposed algorithm.

(ii) If $r_{\max} \gg r_{\min}$: if $r_{\max} \gg r_{\min}$, letting grid length $\sqrt{2}r_{\min}$ and continuing moving among candidate locations at each time step as discussed in the current algorithm would not give effective coverage, since then many overlapping among sensing ranges at consecutive time steps will occur for nodes having $r > r_{\min}$. Thus, depending on the sensing radius and allowable maximum speed, the candidate locations and thus the time step for a movement for a given mobile node should be carefully decided.

In the proposed algorithm, it was assumed that the mobile nodes have enough energy to perform mobility in the required time duration $\widetilde{T}$. As one of the reviewers pointed out, in many real-world settings, mobile nodes have limited energy and may deplete the power supplies before the required task is done. In the following, we discuss how to modify the algorithm in order to address this problem.

Approach 1. Assume that the energy of some mobile nodes may be depleted before completing the required mobility during the time interval $\widetilde{T}$. Let $\rho_{m_j,\max}$ be the maximum

A. NOTATIONS:
$\mathbf{g}_{s_k}(nT_m)$: base price vector at static node s_k at time $t = nT_m$
$\mathcal{U}_{s_k}$: set of all *void* cell indices belongs to static node s_k
$N_{m,k}(nT_m)$: number of mobile nodes from which the static node s_k receives locations information at time nT_m
$\mathcal{C}_{m,j}(nT_m)$: set of cell indices corresponding to candidate cells of mobile node m_j at time nT_m
$\mathcal{U}_{s_k}^{m_j}(nT_m)$: set of cell indices belongs to both $\mathcal{C}_{m,j}(nT_m)$ and $\mathcal{U}_{s_k}$
$\mathbf{g}_{s_k}^{m_j}(nT_m)$: base price vector corresponding to cell indices in $\mathcal{U}_{s_k}^{m_j}$
$P_{j,k}^*$: element with maximum value (maximum base price) in $\mathbf{g}_{s_k}^{m_j}(nT_m)$
$c_{j,k}^*$: cell index corresponding to $P_{j,k}^*$

B. INITIALIZATION AT TIME $t = 0$:
    Determine $\mathcal{U}_{s_k}$ for all $s_k \in \mathcal{V}_s$ based on Voronoi partitions
    Initialize $\mathbf{g}_{s_k}(0)$ as in Section 4.2.1

C. AT STATIC NODE s_k AT TIME $t = nT_m$:
After receiving location (cell) information from neighboring mobile nodes:
    Update the base price vector $\mathbf{g}_{s_k}(nT_m)$ as in Section 4.2.2
    for $j = 1 : N_{m,k}(nT_m)$ **do**
        Check → $\mathcal{U}_{s_k}^{m_j}(nT_m)$ is non-empty
        if $\mathcal{U}_{s_k}^{m_j}(nT_m)$ is non-empty **then**
            check → m_j is isolated
            if m_j is isolated **then**
                Find $P_{j,k}^*$ and $c_{j,k}^*$ and transmit to mobile node m_j
            else $\{m_j$ is not isolated$\}$
                Send cell IDs and their base prices in the set $\mathcal{U}_{s_k}^{m_j}(nT_m)$ to mobile node m_j
            end if
        else $\{\mathcal{U}_{s_k}^{m_j}(nT_m)$ is empty$\}$
            Send nothing to mobile node m_j
        end if
    end for

D. AT MOBILE NODE m_j AT TIME $t = nT_m$:
    Broadcast location information to neighboring static nodes
After receiving base prices for relevant candidate locations from neighboring static nodes:
    check → m_j is isolated
    if m_j is isolated **then**
        select candidate cell with maximum base price
    else $\{m_j$ is not isolated$\}$
        call $duplicate\_covering(m_j)$
    end if
After selecting candidate cell corresponding to time $(n+1)T_m$:
    Check → need for *static* cell compensation
    if *static* cell compensation is required **then**
        Adjust the location to be moved in the selected candidate cell according to Section 4.4
    else $\{$*static* cell compensation is not required$\}$
        Move to the center of the selected candidate cell by time $(n+1)T_m$
    end if

$duplicate\_covering(m_j)$
    Exchange local information with neighboring mobile nodes to check for duplicate covering
    if yes:(duplicate covering) **then**
        Exchange next highest base prices to determine the best candidate cell as in Section 4.3
    else $\{$no:(no duplicate covering)$\}$
        select candidate cell with maximum base price
    end if

Algorithm 1: Mobile-static collaborative mobility protocol.

distance that the mobile node m_j can travel before recharging/replacing its battery. Let $\mathcal{E}_{(n+1)T_m}(c_{m_j}(nT_m), c_{m_j}((n+1)T_m))$ be the energy consumption of the mobile node m_j when moving from the cell $c_{m_j}(nT_m)$ to the cell $c_{m_j}((n+1)T_m)$ during the time step from nT_m to $(n+1)T_m$ where $c_{m_j}(nT_m)$ is the index of the cell in which the mobile node m_j is located at time nT_m. If we assume that a simple energy model, where the energy is linearly related to the distance traveled by the mobile node, we have $\mathcal{E}_{(n+1)T_m}(c_{m_j}(nT_m), c_{m_j}((n+1)T_m)) = \alpha_0 \| d(c_{m_j}(nT_m), c_{m_j}((n+1)T_m)) \|$ where $\| d(c_{m_j}(nT_m), c_{m_j}((n+1)T_m)) \|$ is the Euclidian distance from the location of the cell $c_{m_j}(nT_m)$ to the cell $c_{m_j}((n+1)T_m)$ and α_0 is a constant (in units Joules per meter). Further let $\rho_{m_j}((n+1)T_m) = \rho_{m_j}(nT_m) + \| d(c_{m_j}(nT_m), c_{m_j}((n+1)T_m)) \|$ be the total distance that the mobile node m_j has moved by time $(n+1)T_m$. We assume that each mobile node m_j can update $\rho_{m_j}((n)T_m)$ at time nT_m by itself.

Now, as described in Section 4.1, when the mobile m_j broadcasts its current cell ID at time nT_m, it also sends a message to its nearby static nodes to inform that its energy is about to be depleted if $\rho_{m_j,\max} - \rho_{m_j}(nT_m) < \rho_0$ where ρ_0 is a threshold value. This value can be determined by the average time it takes for the network to insert another mobile node before the energy of m_j is completely depleted. This information lets the nearby static nodes know that the energy of mobile node m_j is about to be depleted, so the network can take necessary actions to replace it. Once a new mobile node is added to the network (this can be initially located in a different cell), the cell in which the mobile node m_j is located is considered as a general *void* cell (in which there is no mobile node) and its base price is updated as described in Section 4.2.

Note that in this approach, it is able to maintain the same fraction of mobile nodes until the required task is completed (time $\widetilde{T}$ is elapsed). Also the mobile nodes in which the energy is depleted can be made available for reuse once the batteries are replaced/recharged. Further, the network has to have immediate access to some extra mobile nodes.

Approach 2. Another approach to resolve the problem is to allow time-varying number of mobile nodes in the network, that is, to add and remove certain number of mobile nodes in a timely manner. Since still the number of static nodes is assumed to be a constant, the *void* cell assignment for each static node is the same. Thus, when a mobile node is removed from the network at any given time, the cell in which the corresponding node was located is assumed to be a *regular void* cell (in which there is no mobile node). The base price of the corresponding *void* cell is incremented by 1 unit at each time step since the time in which the corresponding mobile node is removed until the time that the cell is visited by another mobile node. When a mobile node is added to the network at a given time, the cell in which the mobile node initially present is assumed to be a *void* cell with a mobile node in it. The base prices of corresponding *void* cells are updated as given in Section 4.2 at successive time steps.

In the *mobile-static collaborative mobility model*, it was assumed that static nodes are in operation during the time $\widetilde{T}$ without any failure. However, if a static node fails before the time $\widetilde{T}$ is elapsed, there are certain number of *void* cells (which belong to the corresponding static node's Voronoi polygon) which are not going to be covered by mobile nodes over time. Thus, in that case, the remaining static nodes require to construct new Voronoi polygons and update the IDs of *void* cells that they are responsible to update at each time step.

5. Worst-Case Detection Performance

In this section, we explore an important measure named as *Exposure* [18, 21] which will reflect the effectiveness and the validity of the proposed mobility protocol when the hybrid sensor network is used for target detection applications. *Exposure* is defined in different contexts in the literature, and the general idea behind that is how can a target traverse through the desired field with the minimum probability of being detected (or minimum detection time) by the network. To find the exposure path, different algorithms were proposed in [18, 19, 21] considering different performance measures. For example, in [19], the exposure path was formulated in terms of the sensor field intensity where sensor field intensity is defined as a measure of distance-dependent effective sensing function at a given point from all the sensors in the filed. In [18], algorithms are presented to find exposure in terms of the worst-case coverage. In the worst-case coverage, the exposure path is found by maximizing the closest distance to any sensor node in the target traversal, based on Voronoi partitions and the graph theoretic techniques. In [21], a different definition is given for the exposure. The exposure path is defined as the one with the least probability of being detected, and the authors have taken the measurement uncertainties at sensor nodes into account in finding the *exposure* path. The *exposure* in a mobile sensor network is addressed in [8]. The authors consider minimizing the probability of being detected, based on a given sensing architecture in which mobile nodes make noisy measurements on the emitted signals by the target at a given set of location of the route of the mobile nodes. However, the authors in [8] did not consider specific mobility models for the mobile nodes.

In this work, we find the exposure as the target traversal which minimizes the probability of being detected where the probability of detection is associated with a given presence probability matrix of the hybrid sensor network, in contrast to the work in [8]. Thus, the procedure given in this paper to find the exposure can be generalized to any mobility model in a hybrid/mobile sensor network with a given presence probability matrix.

5.1. Target Model. Without loss of generality, we assume that the target traversal also is a sequence of cells in the grid formed in Section 4. We denote by S, a set of cell sequences which forms a path for the target. We assume that a target can enter and leave the desired region from any boundary (boundary cell). Further we assume that the target should spend at least T_1 time after it enters the region to accomplish the required task and has to leave the region before a maximum of $T_2 \geq T_1$ time. The goal is to find the best path for the target to minimize the probability of being detected by the sensor network.

5.2. Probability of Detection. Let us assume that a target can visit 8 numbers of distinct candidate cells at a given time from its current cell as assumed for the mobile nodes. Let T_r be the time that the target needs to visit its candidate cells from its current position and $v_{r,\max}$ be the maximum speed of the target. Note that if the target has the same speed as with mobile nodes, then we have $T_r \approx T_m$. When the target visits the cell c_k at time $t = nT_r$, the probability of target being detected at time $t = nT_r$, $P(c_k, nT_r) = p_{c_k}$. By p_{c_k}, we denote the presence probability of cell c_k, which is defined as the probability that at least one node is present at the cell c_k at any given time instant. Note that $p_{c_k} = 1$ if c_k is a *static* cell. When a target traverses along the path S for n_0 time steps, where $T_1 \leq n_0 T_r \leq T_2$, the probability that the target is detected by the sensor network is given by

$$P(S, n_0) = 1 - \prod_{j=0}^{n_0} \left(1 - P\left(c_j, jT_r\right)\right), \tag{1}$$

where c_j is the cell index where the target is located at time jT_r.

5.3. Analyzing the Worst-Case Exposure. Let $\mathcal{S}$ be the set of all cell sequences that the target can traverse by time $T_1 \leq n_0 T_r \leq T_2$, then the exposure is defined as [8]

$$\kappa = \min_{S \in \mathcal{S}} P(S, n_0). \tag{2}$$

Note that minimizing $P(S, n_0)$ is equivalent to maximizing $\prod_{j=0}^{n_0}(1 - P(c_j, jT_r))$ and thus maximizing $\sum_{j=0}^{n_0} \log(1 - P(c_j, jT_r))$. Since $\log(1 - P(c_j, jT_r)) \leq 0$, we take maximizing $\sum_{j=0}^{n_0} \log(1 - P(c_j, jT_r))$ as equivalent to minimizing $-\sum_{j=0}^{n_0} \log(1 - P(c_j, jT_r))$. As given in [8], to find the path with minimum exposure, we may convert the problem into a shortest path problem in a time expansion-directed graph by assigning vertices and weights.

For a given time $t = nT_r$, the vertices of the graph represent all the cell indices. We consider the same grid structure as given in Section 4 which has a total of L_T number of cells. We represent vertices at time $t = nT_r$ as (c_k, nT_r) consisting of all cells where $c_k \in \mathcal{U}$. The weight assignment of the graph from time $t = nT_r$ to time $(n+1)T_r$ is performed as follows. If the cell c_k at time $t = nT_r$ (i.e., vertex (c_k, nT_r) in the expansion graph) is a nonboundary cell, it has 9 (including itself) outgoing edges to the corresponding neighboring cells. In particular, let $(c_{k1}, (n+1)T_r)$, $(c_{k2}, (n+1)T_r)$, $(c_{k3}, (n+1)T_r)$, $(c_{k4}, (n+1)T_r)$, $(c_{k5}, (n+1)T_r)$, $(c_{k6}, (n+1)T_r)$, $(c_{k7}, (n+1)T_r)$, $(c_{k8}, (n+1)T_r)$, and $(c_k, (n+1)T_r)$ be the vertices at time $(n+1)T_r$ corresponding to neighboring (candidate) cells of the cell c_k including itself when the current time is $t = nT_r$. Then the vertex (c_k, nT_r) has outgoing edges to all vertices listed above at time $(n+1)T_r$, and the corresponding edge weighs are given by $-\log(1 - P(c_{n+1}, (n+1)T_r))$, where c_{n+1} is the corresponding cell index at time $(n+1)T_r$. For a boundary cell, the number of candidate cells is less than that with a nonboundary cell, and the vertices are connected only with the valid candidate cells. An illustration of vertex and edge assignments for a

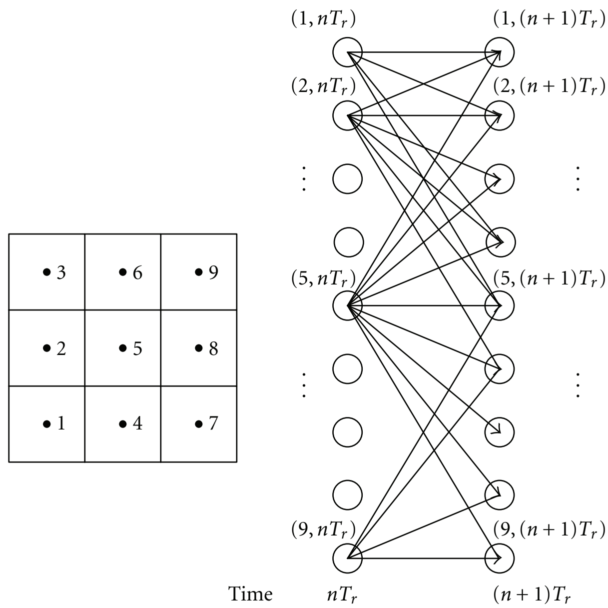


FIGURE 7: Vertex and edge assignment of the expansion graph from time nT_r to time $(n+1)T_r$ for 3×3 square grid; edge weights are not marked. Note that the vertex $(5, nT_r)$ at time nT_r corresponds to a nonboundary cell of the considered grid, and it has 9 outgoing edges from time nT_r to $(n+1)T_r$. All the other vertices at time nT_r correspond to boundary cells. For vertices $(1, nT_r)$, $(3, nT_r)$, $(7, nT_r)$, and $(9, nT_r)$ at time nT_r, they have 4 outgoing edges while for vertices $(2, nT_r)$, $(4, nT_r)$, $(6, nT_r)$, and $(8, nT_r)$, they have 6 outgoing edges from time nT_r to $(n+1)T_r$.

3×3 grid is shown in Figure 7 where edge weights are not marked. Since the target needs to exit the region after time T_2 in the worst case, the graph is expanded at most T_2/T_r steps. Now the problem is to find the target traversal which will result in the minimum weight $w = -\sum_{j=1}^{n_0} \log(1 - P(c_j, jT_r))$ for any $T_1 \leq n_0 T_r \leq T_2$.

Note that in [8], an upper bound and a lower bound for the exposure were given instead of the exact exposure. In contrast, with the constraints that the target may have to exit the region within $[T_1, T_2]$, we present a sequential procedure to find the exact exposure with reduced complexity using graph theoretic techniques.

Denote $\mathcal{U}_b$ and $\mathcal{U}_{nb}$ to be the sets containing indices of boundary and nonboundary cells, respectively. Recall that we assume that the target may enter and exit from any boundary cell after spending T_1 time. Based on the above graph theoretic view, the shortest path (cell sequence) that any cell can be reached (from starting cell) by time $t = T_1$ can be found based on a single-source shortest path algorithm. For simplicity, we assume that $T_1/T_r = q$ is an integer. Denote $s_k(qT_r)$ to be the shortest path (or cell sequence) for the target traversal with the destination being the cell c_k at time qT_r, and $w_k(qT_r)$ be the corresponding weight where $w_k(qT_r) = -\sum_{j=1}^{q} \log(1 - P(c_j^*, jT_r))$ where c_j^*s are in the cell sequence of the corresponding path. Now, we propose the following procedure to find the best traversal for the target.

Let $w_{k,b}^{\min}(qT_r) = \min_{k\in\mathcal{U}_b} w_k(qT_r)$ be the minimum weight of all the shortest paths with a boundary cell being the destination cell at time $t = qT_r = T_1$ and $w_{k,nb}^{\min}(qT_r) = \min_{k\in\mathcal{U}_{nb}} w_k(qT_r)$ be the minimum weight of all the shortest paths with a nonboundary cell being the destination cell at time $t = qT_r = T_1$. It can be shown that if $w_{k,nb}^{\min}(qT_r) \geq w_{k,b}^{\min}(qT_r)$, by expanding the graph beyond the time $t = qT_r = T_1$ will not result in any shorter path with corresponding weight less than $w_{k,b}^{\min}(qT_r)$. Thus, if $w_{k,nb}^{\min}(qT_r) \geq w_{k,b}^{\min}(qT_r)$ at time qT_r (or (T_1)), the path with minimum weight is the path corresponding to $w_{k,b}^{\min}(qT_r)$ for a target enters from a particular boundary cell. If $w_{k,nb}^{\min}(qT_r) < w_{k,b}^{\min}(qT_r)$ is a possibility to have a shorter path for the target to exit the region with a less weight (or less probability of being detected) than the path corresponding to the weight $w_{k,b}^{\min}(qT_r)$ which is terminated by time $t = qT_r$, then if $w_{k,nb}^{\min}(qT_r) < w_{k,b}^{\min}(qT_r)$, the graph is expanded to time $t = (q+1)T_r$ while keeping $w_{k,b}^{\min}(qT_r)$ in the memory. The weight assignment for edges connecting vertices from time $t = qT_r$ to $t = (q+1)T_r$ is performed as follows.

From all the shortest paths with the destination cell as a nonboundary cell at time qT_r, we find the set of nonboundary cells which have the corresponding weights at time qT_r less than $w_{k,b}^{\min}(qT_r)$. We connect only these nonboundary cells to their candidate cells at time $(q+1)T_r$. The reason is for the other nonboundary cells at time qT_r where the corresponding weights of their shortest paths are greater than $w_{k,b}^{\min}(qT_r)$, by expanding the vertices corresponding to them beyond qT_r, will not give any shorter path which will result in a less value compared to $w_{k,b}^{\min}(qT_r)$. That is because any path with such a cell being the cell at time qT_r will always result in a weight greater than $w_{k,b}^{\min}(qT_r)$.

At time $(q+1)T_r$, we follow two steps. (i) As in time qT_r, $w_{k,b}^{\min}((q+1)T_r)$ and $w_{k,nb}^{\min}((q+1)T_r)$ are computed. If $w_{k,b}^{\min}((q+1)T_r) \leq w_{k,b}^{\min}(qT_r)$, $w_{k,b}^{\min}(T_r)$ is deleted from the memory, since then it makes sure that there is a shorter path on or beyond time $(q+1)T_r$ having a smaller weight than $w_{k,b}^{\min}(qT_r)$. Then again as in time qT_r, the condition $w_{k,nb}^{\min}((q+1)T_r) \geq w_{k,b}^{\min}((q+1)T_r)$ is checked, and if it is true, the expansion is stopped by time $(q+1)T_r$. If not, that is, if $w_{k,nb}^{\min}((q+1)T_r) < w_{k,b}^{\min}((q+1)T_r)$, the same procedure is continued as in time qT_r, to find the required set of nonboundary cells from which the edges are connected to time $(q+2)T_r$ while keeping $w_{k,b}^{\min}((q+1)T_r)$ in the memory. (ii) If $w_{k,b}^{\min}((q+1)T_r) > w_{k,b}^{\min}(qT_r)$, it checks whether the condition $w_{k,nb}^{\min}((q+1)T_r) \geq w_{k,b}^{\min}(qT_r)$ is satisfied. If it is satisfied, the expansion is stopped by time $(q+1)T_r$ resulting in $w_{k,b}^{\min}(qT_r)$ the minimum weight corresponding to shortest path for the target. If the condition is not satisfied (i.e., if $w_{k,nb}^{\min}((q+1)T_r) < w_{k,b}^{\min}(qT_r)$), the graph is expanded to time $(q+1)T_r$ after finding the required set of nonboundary cells from which the edges are connected to time $(q+2)T_r$ (as in time qT_r) while keeping $w_{k,b}^{\min}(qT_r)$ in the memory. The expansion is stopped at time q_0T_r if either one of the following criteria is met. (i) If $w_{k,nb}^{\min}(q_0T_r) \geq \min\{w_{k,b}^{\min}(qT_r), w_{k,b}^{\min}((q+1)T_r), \ldots, w_{k,b}^{\min}(q_0T_r)\}$ for $q \leq q_0 < T_2/T_r$ and (ii) If $q_0 = T_2/T_r$, where the maximum time for expansion is reached.

Note that with the proposed scheme, the complexity is greatly reduced since after time T_1 a certain number of vertices corresponding to nonboundary cells at each time step do not need to be expanded. On the other hand, with the proposed *mobile-static collaborative mobility model*, as can be observed from the simulation results, the graph does not need to be expanded a large number of time steps after time T_1 due to the approximately uniform nature of the presence probability matrix for the *void* cells. This essentially implies that after the required time is spent in the region (i.e., time T_1), by circulating inside the region to minimize the detection probability is not desirable for the target. That is because, due to the nearly uniform nature of the presence probabilities of *void* cells, target will not find a safer area to avoid being detected inside the region as time goes. Note that the above procedure is for the target traversal starting at a given boundary cell. Thus, to find the worst case scenario over all starting boundary cells, the procedure can be repeated.

Since there is a total of L_T number of cells and the graph is expanded up to time T_2 at the worst case, there is a total of $L_T\tilde{q}$ number of vertices (in the worst case) in the time expansion graph where $\tilde{q} = T_2/T_r$. Each vertex is connected to at most L_c+1 number of vertices where L_c is the number of candidate cells that a node/target can reach from the current position, (in this paper, we have $L_c = 8$). Thus the worst-case complexity in finding the shortest path from a boundary cell is $O(L_T L_c \tilde{q})$. Since there is $|\mathcal{U}_b|$ number of boundary cells, the time complexity of the algorithm is upper bounded by $O(|\mathcal{U}_b| L_T L_c \tilde{q})$. As mentioned before, since the graph does not need to be expanded a large number of time steps after time T_1, a lower bound on the complexity of the algorithm is given by $O(|\mathcal{U}_b| L_T L_c q)$ where $q = T_1/T_r$ as defined before. The proposed procedure is summarized in Algorithm 2.

6. Performance Evaluation

To evaluate the effectiveness and efficiency of the proposed *mobile-static collaborative mobility protocol*, we perform numerical experiments to investigate how well the desired area is covered over time to minimize the time that a *void* cell is unvisited by a mobile node. We depict the results in different perspectives taking the factors, the probability that at least one mobile node visits a particular cell at any given time instant, the average time that any arbitrary point in the network is unvisited, effect of the node speed, and the fraction of mobile nodes, into account.

6.1. Presence Probability Matrix. Denote p_{c_k} to be the probability that at least one node is present at the cell c_k at any given time. Let $\mathbf{\Lambda}$ be the presence probability matrix containing the probabilities of the presence of at least one node at each cell at a given time instant. It is noted that the presence probability of a *static* cell is always 1. For simulations, we consider a sensor network deployed in a $\approx 200 \times 200\,\text{m}^2$ square region with 14×14 grid. Thus, we have $L_T = 196$ total cells in

A. NOTATIONS:

$q = T_1/T_r$: minimum number of time steps a target needs to spend in the region
$s_k^*(qT_r)$: the shortest path (cell sequence) for the target traversal with the destination cell being the cell c_k at time qT_r
$w_{k,b}^{\min}(qT_r)$: (minimum) weight of the shortest path with a boundary cell being the destination cell at time qT_r
$s_{k,b}^*(qT_r)$: corresponding shortest path (cell sequence) which results the weight $w_{k,b}^{\min}(qT_r)$
$w_{k,nb}^{\min}(qT_r)$: (minimum) weight of the shortest path with a non-boundary cell being the destination cell at time qT_r
$s_{k,nb}^*(qT_r)$: corresponding shortest path (cell sequence) which results the weight $w_{k,nb}^{\min}(qT_r)$
$\mathcal{U}'_{nb}(qT_r)$: set of non-boundary cells with the corresponding weights at time qT_r are less than $w_{k,b}^{\min}(qT_r)$
$\overline{w}_{k,b}^{\min}(nT_r)$: $\min\{w_{k,b}^{\min}(qT_r), w_{k,b}^{\min}((q+1)T_r), w_{k,b}^{\min}(nT_r)\}$ is the minimum weight of a boundary cell over time qT_r to nT_r with $n \ge q$

B. AT TIME $t = qT_r$:

Construct the expansion graph over q time steps
Find $w_{k,b}^{\min}(qT_r)$ and $w_{k,nb}^{\min}(qT_r)$
if $w_{k,nb}^{\min}(qT_r) \ge w_{k,b}^{\min}(qT_r)$ **then**
  end procedure: result $\rightarrow$ shortest path $s_{k,b}^*(qT_r)$
else $\{w_{k,nb}^{\min}(qT_r) < w_{k,b}^{\min}(qT_r)\}$
  Find $\mathcal{U}'_{nb}(qT_r)$
  Expand the graph to time $(q+1)T_r$ by connecting edges from vertices corresponding to the cells in $\mathcal{U}'_{nb}(qT_r)$
  Keep $\overline{w}_{k,b}^{\min}(qT_r) = w_{k,b}^{\min}(qT_r)$ in memory
end if

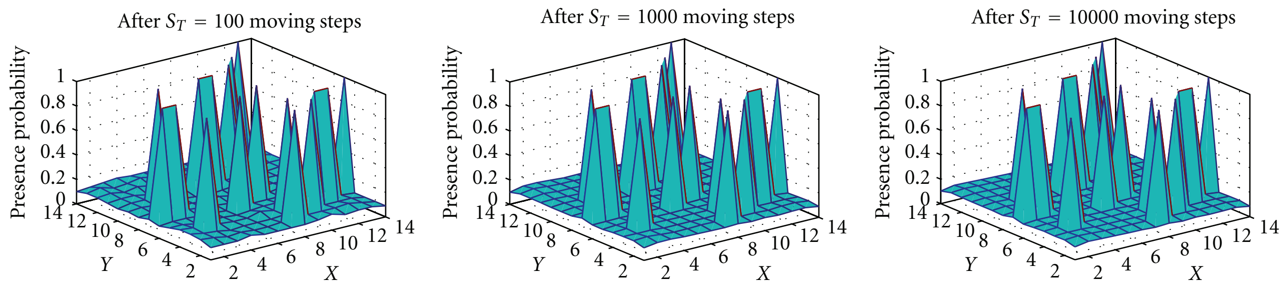
C. AT TIME $t = nT_r$ WITH $q < n < q_0$

Compute $w_{k,b}^{\min}(nT_r)$ and $w_{k,nb}^{\min}(nT_r)$
Check $\rightarrow$ $w_{k,b}^{\min}(nT_r) \le \overline{w}_{k,b}^{\min}((n-1)T_r)$
if $w_{k,b}^{\min}(nT_r) \le \overline{w}_{k,b}^{\min}((n-1)T_r)$ **then**
  $\overline{w}_{k,b}^{\min}(nT_r) = w_{k,b}^{\min}(nT_r)$
else $\{w_{k,b}^{\min}(nT_r) > \overline{w}_{k,b}^{\min}((n-1)T_r)\}$
  $\overline{w}_{k,b}^{\min}(nT_r) = \overline{w}_{k,b}^{\min}((n-1)T_r)$
end if
Check $\rightarrow$ $w_{k,nb}^{\min}(nT_r) \ge \overline{w}_{k,b}^{\min}(nT_r)$
if $w_{k,nb}^{\min}(nT_r) \ge \overline{w}_{k,b}^{\min}(nT_r)$ **then**
  end procedure: result$\rightarrow$ the shortest path corresponding to $\overline{w}_{k,b}^{\min}(nT_r)$
else $\{w_{k,nb}^{\min}(nT_r) < \overline{w}_{k,b}^{\min}(nT_r)\}$
  Find $\mathcal{U}'_{nb}(nT_r)$
  Expand the graph to time $(n+1)T_r$ by connecting edges from vertices corresponding to the cells in $\mathcal{U}'_{nb}(nT_r)$
  Keep $\overline{w}_{k,b}^{\min}(nT_r)$ in memory
end if

Algorithm 2: (Procedure to find best target traversal).

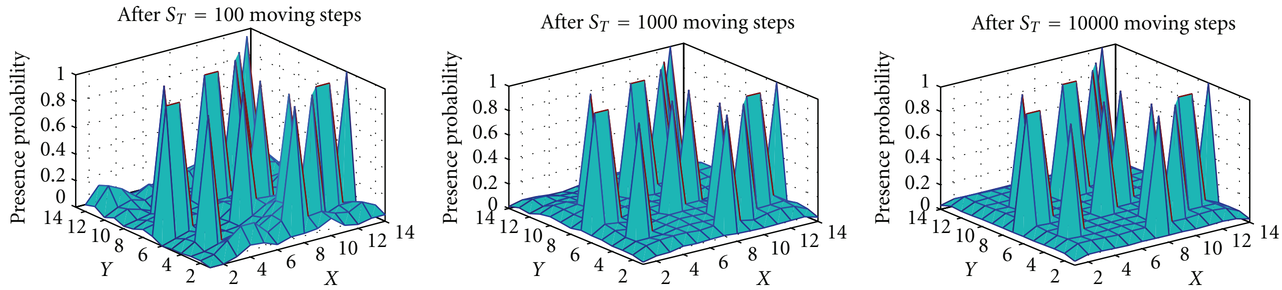
the network. We let $r = 10$ m such that the grid length becomes $l = \sqrt{2}r \approx 14.14$ m. Denote S_T to be the number of moving steps where it is assumed that $S_T T_m \le \widetilde{T}$ where $\widetilde{T}$ is the maximum duration of time in which the node mobility should be performed, as discussed before. We compare the performance of the proposed mobility protocol with widely used bounced random walk mobility model with a step size of l. We mean by bounced random walk that when the mobile nodes hit the boundary under random walk, they bounce back with probability 1. It is noted that with the bounced random walk model, mobile nodes move independently, and there is no collaboration among nodes. At a expense of certain collaboration with static nodes, our goal is to investigate how efficient the scheme presented in the paper in providing dynamic coverage compared to that with a mobility scheme which does not have any collaboration. In other words, this comparison quantitatively illustrates the gain that can be achieved by collaboration among nodes compared to that with no collaboration among nodes.

Figures 8 and 9 show the presence probability matrices with proposed mobility scheme and with bounced random walk scheme, respectively. The presence probability matrices are shown after completing $S_T = 100$, $S_T = 1000$, and $S_T = 10,000$ moving steps, respectively, for $N = 40$ and $\lambda_m = 0.5$. Note that in Figures 8 and 9, the high peaks with presence probability 1 reflect the presence probability of *static* cells. Looking at the presence probabilities of *void* cells under two mobility schemes, from Figure 8 it can be seen that the presence probabilities of *void* cells become uniform after completing relatively a small number of steps compared to that with random walk model (Figure 9). When the number of movements steps is large, it can be seen from Figure 9 that the presence probabilities of *void* cells under random walk mobility models also become uniform, as expected.



(a) $S_T = 100$ (b) $S_T = 1000$ (c) $S_T = 10,000$

FIGURE 8: Presence probability matrix with proposed mobility protocol, $N = 40$, $\lambda_m = 0.5$, and $v_{\max} = 10$ m/s (a) after moving steps $S_T = 100$, (b) after moving steps $S_T = 1000$, and (c) after moving steps $S_T = 10,000$.



(a) $S_T = 100$ (b) $S_T = 1000$ (c) $S_T = 10,000$

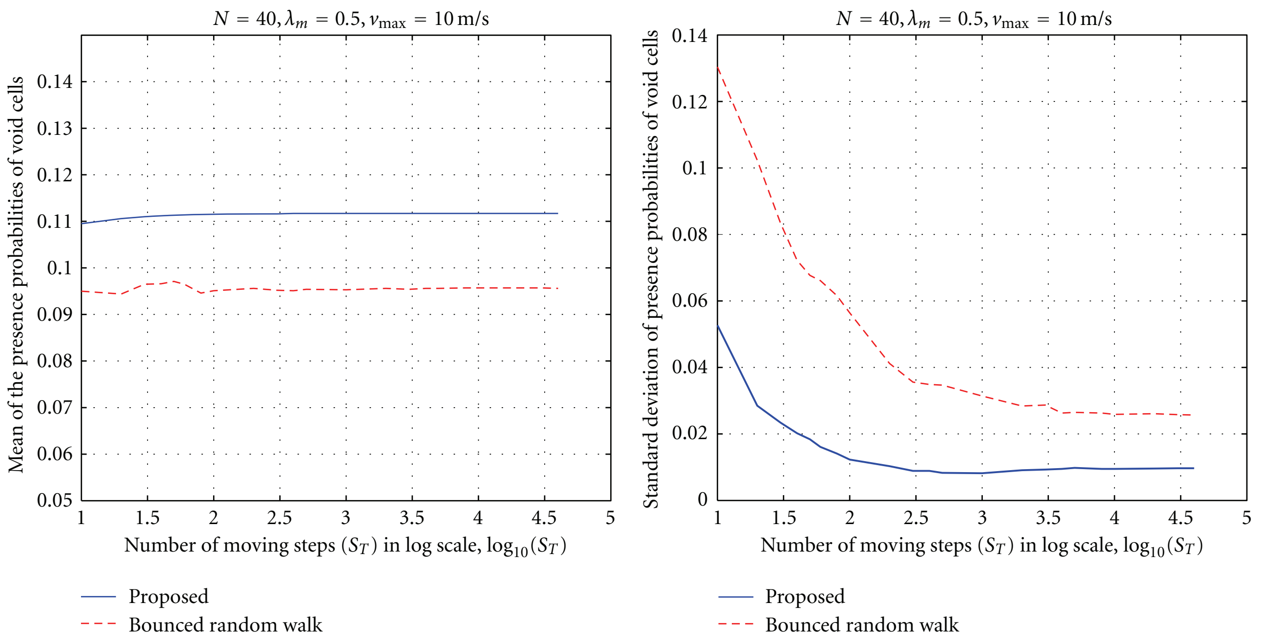
FIGURE 9: Presence probability matrix with proposed mobility protocol, $N = 40$, $\lambda_m = 0.5$, and $v_{\max} = 10$ m/s (a) after moving steps $S_T = 100$, (b) after moving steps $S_T = 1000$, and (c) after moving steps $S_T = 10,000$.

This is because, with any independent and random mobility scheme, each point in the region of interest is visited equally likely as the number of steps increases. However, as can be seen from Figures 8 and 9, in terms of the number of movement steps needed to achieve this uniformity, the *mobile-static collaborative mobility protocol* for hybrid sensor network outperforms the random mobility schemes.

To further investigate the relationship between the number of movement steps and the uniformity of presence probabilities of *void* cells, in Figure 10 we plot the mean and the standard deviation of the presence probabilities of *void* cells as the number of movement steps (S_T) increases for the *mobile-static collaborative mobility protocol* and random walk mobility scheme. In Figure 10, we use S_T in $\log_{10}$ scale. From Figure 10(a), it can be seen that the mean of the presence probabilities of *void* cells converges to a constant with a relatively small number of movement steps for both schemes, and the corresponding mean value is relatively large with the *mobile-static collaborative mobility protocol* compared to that with the random mobility scheme. This essentially implies, with the proposed protocol, *void* cells are covered much efficiently over time compared to that with random mobility scheme. In Figure 10(b), we plot the standard deviation of the presence probabilities of *void* cells with $\log_{10}S_T$. Note that the standard deviation of presence probabilities of *void* cells acts as a measure of the quality of uniformness of the presence probabilities. From Figure 10(b), it can be seen that the standard deviation of presence probabilities of *void* cells converges to a constant value for both mobility schemes and the threshold number of movement steps required for this to happen is much less with the *mobile-static collaborative mobility protocol* compared to that with the random mobility scheme. Moreover, the constant value of this convergence is small for proposed protocol compared to that with the bounced random walk scheme. These observations imply that the presence probabilities of *void* cells approach a constant value after a relatively small number of moving steps with the *mobile-static collaborative mobility protocol* compared to that with random walk model. In other words, the collaboration among the nodes in mobility management results in a more uniform coverage of the area not covered by the static nodes with a small number of moving steps compared to that with the independent random walk mobility model.

In Figure 11, the presence probabilities of *void* cells are shown when the fraction of mobile nodes varies. In Figure 11, we let $N = 40$, $v_{\max} = 10$ m/s, and the number of moving steps $S_T = 1000$. It can be seen that when the fraction of mobile nodes increases, the presence probability of *void* cells also increases, since then the frequency that any mobile node can visit a *void* cell becomes high.

In Figure 12, we illustrate how effective the collaborative mobility management algorithm is when the number of static nodes varies for a given number of mobile nodes. In Figure 12, we let $v_{\max} = 10$ m/s, the number of moving steps $S_T = 1000$, and the number of static nodes varies from 10



(a) mean

(b) standard deviation

Figure 10: Mean and the standard deviation of presence probabilities at *void* cells versus the number of movement steps S_T (in log scale) for proposed protocol and the bounced random walk mobility model, $N = 40$, $\lambda_m = 0.5$, and $v_{max} = 10$ m/s. (a) Mean. (b) Standard deviation.

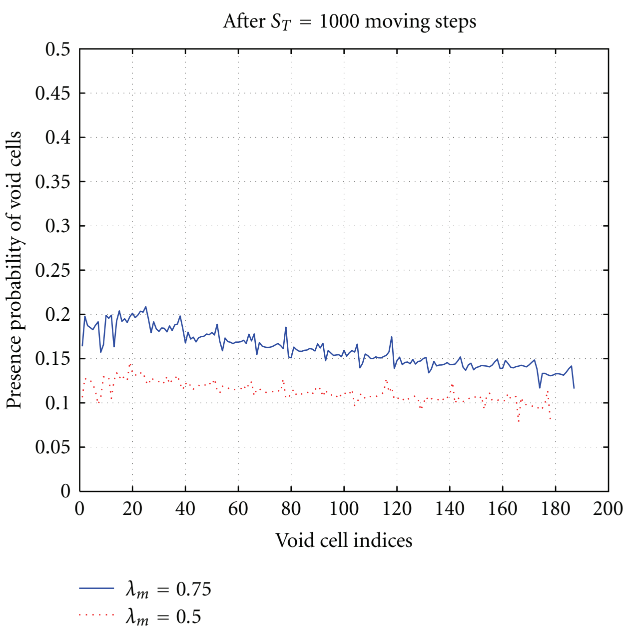


Figure 11: Presence probabilities of void cells for $N = 40$, $\lambda_m = 0.5$, and $\lambda_m = 0.75$.

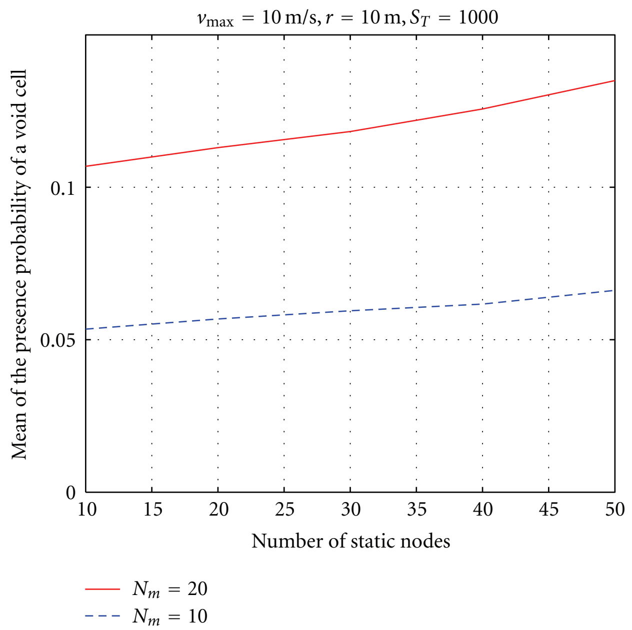
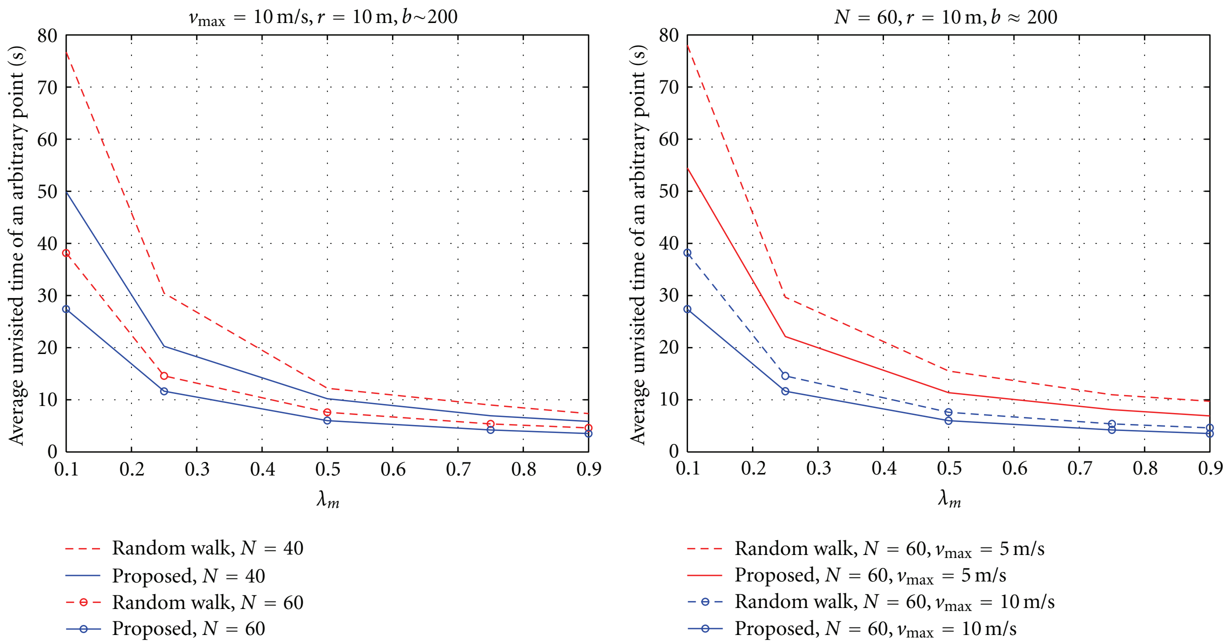


Figure 12: Mean of the presence probabilities of *void* cells when the number of static nodes varies; $v_{max} = 10$ m/s, $S_T = 1000$.

to 50. Since we assume 14×14 grid (196 total cells) in the simulations, there is a maximum of 186 and 146 *void* cells when there are 10 and 50 static nodes, respectively. The mean of the presence probabilities at *void* cells is averaged over 20 iterations. For a fixed number of mobile nodes, it can be seen from Figure 12 that the mean of the presence probability at a *void* cell increases with the proposed algorithm as the number of static nodes increases. When the number of static nodes increases, the number of *void* cells decreases, since then there are more *static* cells in the network. Then based on the collaboration among static and mobile nodes, the given number of mobile nodes only needs to move among those *void* cells. Furthermore, as the number of mobile



(a) (b)

Figure 13: (a) Average time taken for an arbitrary point to be revisited for different network sizes N: $v_{max} = 10$ m/s, $r = 10$ m, $b \approx 200$ m. (b) Average time taken for an arbitrary point to be revisited for different node speeds: $N = 60$, $r = 10$ m, and $b \approx 200$ m.

nodes increases, it can be seen from Figure 12 that the rate of increase of the mean of the presence probabilities of the *void* cells also increases. These observations validate the effectiveness of the proposed collaborative mobility protocol in dynamic coverage improvement in hybrid sensor networks.

6.2. Average Unvisited Time of an Arbitrary Point. In the next experiment, we evaluate the performance of the *mobile-static collaborative mobility protocol* in terms of the average time that any arbitrary point remains uncovered by the hybrid sensor network. We compare the results of the proposed scheme with a random mobility model as before. Figure 13(a) shows the average unvisited time of an arbitrary point in the network with the proposed mobility protocol and random walk mobility model (with step size of l) when the number of total nodes in the network varies ($N = 40$ and $N = 60$). In Figure 13(a), we let $v_{max} = 10$ m/s, $r = 10$ m.

It can be seen that when the fraction of mobile nodes is small, a significant performance improvement can be obtained by the *mobile-static collaborative mobility protocol*, over that with the random walk mobility model. Note that due to extra cost needed for deploying mobile nodes compared to static nodes, this is the most interesting scenario. As mentioned earlier in the paper, with the random mobility models, efficient coverage is not achievable in hybrid sensor networks specially for small values of λ_m. With the random walk mobility model, there is an overlapping among sensing ranges of static and mobile nodes in a hybrid sensor network since there is no coordination among static and mobile nodes. Thus, any point that is not covered by a static node will be covered with a less frequency with random walk model compared to that with *mobile-static collaborative mobility protocol*, especially when the fraction of mobile nodes is small. However, from Figure 13(a), it can be seen that when λ_m increases, the unvisited time with the proposed scheme is not much different from the random walk scheme. That is because when there is a large number of mobile nodes compared to static nodes, the frequency that a mobile node can cover any point not covered by static nodes is high. Also when the total number of nodes increases, it can be seen that even with very small fraction of mobile nodes, very efficient coverage is achieved in terms of the revisiting time by the proposed scheme. The performance gain of the proposed scheme over the random walk mobility model is more significant when N is smaller, that is, when the network is to be covered by a small number of total nodes.

Figure 13(b) shows the average unvisited time of an arbitrary point when the speed of a mobile changes. In Figure 13(b), we let $N = 60$, $r = 10$ m, and the plots correspond to $v_{max} = 5$ m/s and 10 m/s. It can be seen that especially with a lower fraction of mobile nodes, the speed of mobile nodes affects the network performance significantly compared to that with a large fraction of mobile nodes. However, irrespective of the node speed, it can be seen that with relatively small fraction of mobile nodes, the *mobile-static collaborative mobility protocol* outperforms the random mobility schemes. For results in Figure 13, we ran simulations for 10000 s and averaged over 50,000 arbitrary points.

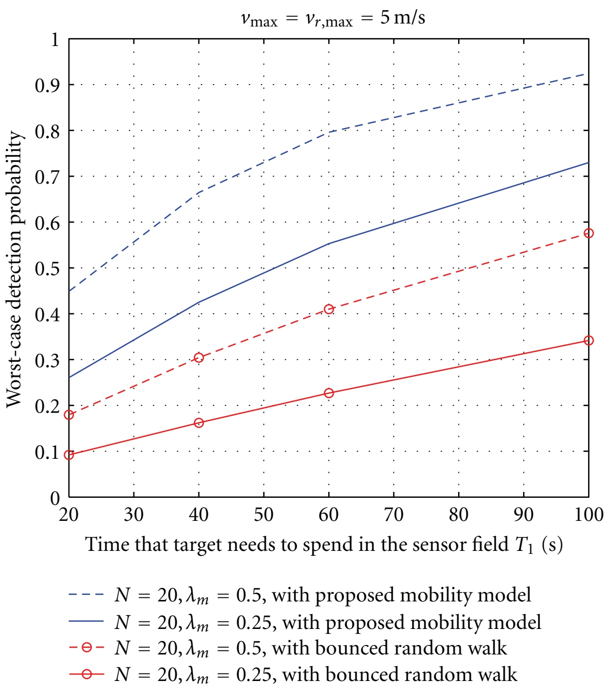


FIGURE 14: Worst-case detection performance, $v_{\max} = v_{r,\max} = 5$ m/s, $b \approx 200$ m.

6.3. Worst-Case Detection Performance. In this subsection, we evaluate the worst-case detection performance based on the algorithm presented in Section 5. The worst-case detection performance with the *mobile-static collaborative mobility protocol* is compared again with that with the bounced random walk model. To find the worst-case detection performance as given by Section 5, we find the presence probability matrix with random walk with a step size of l.

Figure 14 shows the worst-case detection probability versus the minimum time that the target needs to spend in the desired field, T_1. In Figure 14, we assume that the maximum speed of a mobile node and the target is the same, where $v_{\max} = v_{r,\max} = 5$ m/s. Note that the higher the worst-case detection probability, the less safe for the target to enter the sensing region. It can be seen from Figure 14 that, with the proposed node mobility scheme, it is more dangerous for the target to enter the sensing region and very less likely that it can find a safe path to exit. Also, it can be seen that the more time the target has to be in the desired region (i.e., T_1 is increases) to perform the required task, the more vulnerable for the target, and the vulnerability is more severe as T_1 increases with the *mobile-static collaborative mobility protocol* when compared to that with random walk model.

7. Conclusions

In this paper, we proposed a distributed and collaborative mobility management algorithm, called *mobile-static collaborative mobility protocol* for mobile node navigation in a hybrid sensor network consisting of both static and mobile nodes. The *mobile-static collaborative mobility protocol* provides efficient dynamic coverage for the area not covered by static nodes by maximizing the revisiting time of an arbitrary point by any mobile node in the network. Moreover, the proposed scheme can be implemented distributively by collaborating among static and mobile nodes locally, having only communication in the local neighborhood. It was shown that the proposed scheme provides an approximate uniform coverage for the area not covered by the static nodes after completing relatively a small number of movement steps compared to that with random walk model. Thus, the proposed model is more effective when the network is designed for detecting targets in which the existence is unknown. The proposed scheme also outperforms the random mobility schemes in terms of the average revisiting time of an arbitrary point by any mobile node in the network, especially when the fraction of mobile nodes is small. Moreover, we developed a sequential methodology to find the worst-case target traversal when the target tries to evade the region with the minimum probability of being detected by the sensor network. It was shown that with the *mobile-static collaborative mobility protocol*, it is very less likely that a target can find a safe path to traverse in the sensing region without being detected.

In the future, we hope to extend the work in different directions. One direction is to modify the proposed node mobility algorithm when certain prior information (probabilistically) is available about the phenomenon of interest. In such cases, the node mobility should be managed to optimize different cost/reward functions by taking these prior information into account in contrast to the current work. Another direction of interest is to investigate how to effectively schedule mobile nodes for different tasks when the hybrid sensor network is designed for different concurrent applications.

Disclosure

A part of this work was presented at IEEE Global Communications Conference at Miami, FL in Dec 2010.

Acknowledgment

This research was supported by the National Science Foundation (NSF) under the grant CCF-0830545.

References

[1] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, pp. 1293–1303, April 2003.

[2] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[3] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem," in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS '02)*, Fukouka, Japan, June 2002.

[4] S. Chellappan, X. Bai, B. Ma, and D. Xuan, "Sensor networks deployment using flip-based sensors," in *Proceedings of the 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '05)*, pp. 291–298, November 2005.

[5] J. Wu and S. Wang, "Smart: a scan based movement-assisted deployment method in wireless sensor networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '05)*, Miami, Fla, USA, March 2005.

[6] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05)*, pp. 300–308, May 2005.

[7] T.-L. Chin, P. Ramanathan, and K. K. Saluja, "Analytical modeling of detection latency in mobile sensor networks," in *Proceedings of the Conference on Information Processing in Wireless Sensor Ntworks (IPSN '06)*, April 2006.

[8] T.-L. Chin, P. Ramanathan, K. K. Saluja, and K.-C. Wang, "Exposure for collaborative detection using mobile sensor networks," in *Proceedings of the 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '05)*, pp. 743–750, Washington, DC, USA, November 2005.

[9] J.-C. Chin, Y. Dong, W.-K. Hon, and D. K. Y. Yau, "On intelligent mobile target detection in a mobile sensor noetwork," in *Proceedings of the IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '07)*, Pisa, Italy, October 2007.

[10] Y. Zou and K. Chakrabarty, "Distributed mobility management for target tracking in mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 872–887, 2007.

[11] R. Olfati-Saber, "Distributed tracking for mobile sensor networks with information-driven mobility," in *Proceedings of the American Control Conference (ACC '07)*, pp. 4606–4612, New York, NY, USA, July 2007.

[12] G. Wang, G. Cao, and T. LaPorta, "A bidding protocol for deploying mobile sensors," in *Proceedings of the 11th IEEE International Conference on Network Protocols*, pp. 315–324, November 2003.

[13] W. Wang, V. Sirinivasan, and K.-C. Chua, "Trade-offs between mobility and density for coverage in wireless sensor networks," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, pp. 39–50, Montral, Canada, 2007.

[14] A. Verma, H. Sawant, and J. Tan, "Selection and navigation of mobile sensor nodes using a sensor network," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pp. 41–50, Pisa, Italy, March 2005.

[15] R. Tan, G. Xing, J. Wang, and H. C. So, "Collaborative target detection in wireless sensor networks with reactive mobility," in *Proceedings of the 16th International Workshop on Quality of Service (IWQoS '08)*, Enschede, The Netherlands, June 2008.

[16] G. Xing, J. Wang, KE. Shen, Q. Huang, X. Jia, and H. C. So, "Mobility-assisted spatiotemporal detection in wireless sensor networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS '08)*, pp. 103–110, Beijing, China, June 2008.

[17] O. Kosut, A. Turovsky, J. Sun, M. Ezovski, L. Tong, and G. Whipps, "Integrated mobile and static sensing for target tracking," in *Proceedings of the Military Communications Conference (MILCOM '07)*, pp. 1–7, October 2007.

[18] S. Megerian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Worst and best-case coverage in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 84–92, 2005.

[19] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 139–150, July 2001.

[20] G. Barriac, R. Mudumbai, and U. Madhow, "Distributed beamforming for information transfer in sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 81–88, Berkeley, Calif, USA, April 2004.

[21] V. Phipatanasuphorn and P. Ramanathan, "Vulnerability of sensor networks to unauthorized traversal and monitoring," *IEEE Transactions on Computers*, vol. 53, no. 4, pp. 364–369, 2004.

Preliminary call for papers

The 2011 European Signal Processing Conference (EUSIPCO-2011) is the nineteenth in a series of conferences promoted by the European Association for Signal Processing (EURASIP, www.eurasip.org). This year edition will take place in Barcelona, capital city of Catalonia (Spain), and will be jointly organized by the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) and the Universitat Politècnica de Catalunya (UPC).

EUSIPCO-2011 will focus on key aspects of signal processing theory and applications as listed below. Acceptance of submissions will be based on quality, relevance and originality. Accepted papers will be published in the EUSIPCO proceedings and presented during the conference. Paper submissions, proposals for tutorials and proposals for special sessions are invited in, but not limited to, the following areas of interest.

Areas of Interest

- Audio and electro-acoustics.
- Design, implementation, and applications of signal processing systems.
- Multimedia signal processing and coding.
- Image and multidimensional signal processing.
- Signal detection and estimation.
- Sensor array and multi-channel signal processing.
- Sensor fusion in networked systems.
- Signal processing for communications.
- Medical imaging and image analysis.
- Non-stationary, non-linear and non-Gaussian signal processing.

Submissions

Procedures to submit a paper and proposals for special sessions and tutorials will be detailed at www.eusipco2011.org. Submitted papers must be camera-ready, no more than 5 pages long, and conforming to the standard specified on the EUSIPCO 2011 web site. First authors who are registered students can participate in the best student paper competition.

Important Deadlines:

Proposals for special sessions	15 Dec 2010
Proposals for tutorials	18 Feb 2011
Electronic submission of full papers	**21 Feb 2011**
Notification of acceptance	23 May 2011
Submission of camera-ready papers	6 Jun 2011

Webpage: www.eusipco2011.org

Organizing Committee

Honorary Chair
Miguel A. Lagunas (CTTC)

General Chair
Ana I. Pérez-Neira (UPC)

General Vice-Chair
Carles Antón-Haro (CTTC)

Technical Program Chair
Xavier Mestre (CTTC)

Technical Program Co-Chairs
Javier Hernando (UPC)
Montserrat Pardàs (UPC)

Plenary Talks
Ferran Marqués (UPC)
Yonina Eldar (Technion)

Special Sessions
Ignacio Santamaría (Unversidad de Cantabria)
Mats Bengtsson (KTH)

Finances
Montserrat Nájar (UPC)

Tutorials
Daniel P. Palomar (Hong Kong UST)
Beatrice Pesquet-Popescu (ENST)

Publicity
Stephan Pfletschinger (CTTC)
Mònica Navarro (CTTC)

Publications
Antonio Pascual (UPC)
Carles Fernández (CTTC)

Industrial Liaison & Exhibits
Angeliki Alexiou (University of Piraeus)
Albert Sitjà (CTTC)

International Liaison
Ju Liu (Shandong University-China)
Jinhong Yuan (UNSW-Australia)
Tamas Sziranyi (SZTAKI -Hungary)
Rich Stern (CMU-USA)
Ricardo L. de Queiroz (UNB-Brazil)