Topology Control for Secured Coverage in Wireless Sensor Networks *

Zhen Jiang
Computer Sci. Dept.
West Chester University
West Chester, PA 19383, USA
zjiang@wcupa.edu

Jie Wu
Computer Sci. & Eng. Dept.
Florida Atlantic University
Boca Raton, FL 33431, USA
jie@wcupa.edu

Afrand Agah, Bin Lu
Computer Sci. Dept.
West Chester University
West Chester, PA 19383, USA
{aagah, blu}@wcupa.edu

Abstract

In this paper, we present a new control method that makes some node adjustments in local areas in an effort to cover the "holes" in wireless sensor networks. Many security applications often face the problem of holes when some sensor nodes are disabled from the collaboration due to their failures and misbehavior. Affected by malicious attacks, these holes may occur dynamically and such a problem cannot be solved completely by simply deploying more redundant sensors. We propose a snake-like cascading replacement process in a local area in order to fill in the vacant area with trusted nodes. Only 1-hop neighborhood is used in our approach. Its implementations under both a passive model and an active model are discussed. The simulation results of our new control method show substantial improvements in total moving distance, total number of moves, and process converging speeds, compared with the best result known to date.

1 Introduction

Recent advances in micro-electromechanical systems, digital electronics, and wireless communications have enabled the development of low-cost, low-power, multifunction sensor devices [7]. These devices can operate autonomously to gather, process, and communicate information about their environments. When a large number of sensor devices collaborate using wireless communications, they constitute a *wireless sensor network* (WSN) [2]. Applications of WSNs range from environmental monitoring to surveillance to target detection. Due to the fact that sensors can very easily fail or misbehave, many nodes should be isolated from the network collaboration [9]. Thus, a "hole" in the surveillance area may occur in the deployed area, and such an occurrence may be dynamic. Many security applications often face the problem of such holes in surveillance areas, causing incomplete coverage. For instance, as indicated in [15], the attacker can cause the nodes to move and deplete their battery power, which might reduce node density in certain areas. The holes of surveillance area can occur even when many redundant sensors are deployed. To secure the whole network and ensure that it works correctly, a complete coverage of its surveillance area must be provided.

Recently, movement-assisted sensor deployment has received considerable attention. Some extended virtual force methods that simulate the attractive and repulsive forces between sensor nodes have been proposed [8, 13, 17]. In these methods, sensors in a relatively dense region will explode slowly according to each other's repulsive force and head towards a hole in the network. However, as indicated in [14], without global information, these methods may take a long time to converge and are not practical for real applications due to the cost in total moving distance, total number of moves, and communication/computation. Then, in [14], a more practical balancing method under the virtual grid model [16] is discussed. This method allows for quick convergence. However, node adjustments in the entire grid network are needed, and many unnecessary node movements are incurred just for providing the coverage for a single hole.

In this paper, rather than preventing the occurrence of the holes, we present our localized control method that makes some node adjustments in a local area to cover the holes so that the connectivity and the coverage of networks can be recovered. According to how the vacant area is detected, we provide two different implementations, one under a passive model and the other under an active model. Only 1-hop neighborhood is used and the control schemes are implemented in a fully distributed manner so that the entire network system becomes more scalable under dynamic changes. Compared with the best results known to date in this field [14], our new method relaxes the assumption that the entire network is connected. It is more practical due to

*The work was supported in part by NSF grants CNS 0422762, CNS 0434533, CNS 0531410, and CNS 0626240.

1-4244-1455-5/07/$25.00©2007 IEEE

its improvement in total moving distance, total number of moves, and process converging speed.

A short summary of our approach follows. First, we partition the surveillance area into many small squares (of the size $r \times r$) in a virtual grid model [16]. After many faulty sensors and misbehaving sensors (affected by malicious attacks) are disabled, the rest of the nodes (also called trusted nodes) form trust networks. In each grid, one trusted node will be elected as the grid head to monitor the neighborhood. As indicated in [16], the connectivity and coverage of trust networks can be guaranteed if each grid has its own head. When a grid does not have a head, a *replacement process* will be initiated to move trusted node(s) into this vacant area. In this way, each grid will be filled by at least one trusted node which will become the head, allowing the whole network to keep its coverage and connectivity, even when many nodes are disabled and the network is disconnected.

According to how the vacant gird is detected, two implementations are introduced: the passive model and the active model. In the passive model, the vacant grid is detected only when a communication flow needs to pass it. In the active model, each grid head will monitor the whole area of its neighboring grids. Whenever a vacant grid occurs, a replacement process will be initiated immediately at its neighboring grids. It is noted that the movement of a node during the replacement process may trigger another replacement for that particular node. In our experimental results, we show that such a cascading movement can converge quickly under both the passive model and the active model.

The identification of trusted nodes depends on the analysis of node behavior and the power management. The corresponding procedure can be vary by system. This paper only focuses on the sufficient coverage of WSN by solving the coverage hole problem. It uses a more general security system in which we assume that the trusted nodes have been identified via public keys [3], the "monitoring and rating system" [4, 5, 11], or the game theory [1]. It is noted that a trusted node may not really be trustworthy. For instance, the wormhole attacks will distort the picture of trust networks. The node can be removed without the neighbors knowing; that is, a hole may occur without any neighboring nodes noticing. However, after the change in topology is detected by applying the methods in [6, 10], a more trustworthy network can be achieved and furthermore a more secured complete coverage can be achieved.

2 Preliminary

We assume that all the nodes have the same communication range R. The nodes inside the communication range are called neighbors and two neighboring nodes are directly

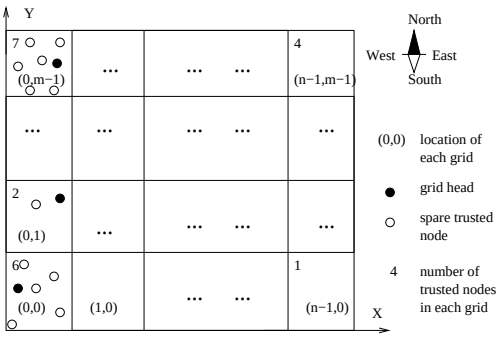


Figure 1. Virtual grid system and grid heads.

connected. Each node u has its location, which is simply denoted by $L(u)$. The location information can be discovered by having Global Positioning System (GPS) receivers at some fixed nodes or a mobile beacon node, or just by relying on the relative coordinate system. We partition the whole network into an $n \times m$ 2-D grid system (see Figure 1). Each grid is of a square size $r \times r$ and is denoted by its relative location in the entire system, say (x, y) ($0 \leq x \leq n-1$, $0 \leq y \leq m-1$). Two grids (x_1, y_1) and (x_2, y_2) are called neighboring grids if their location addresses differ in one and only one dimension, say X. Moreover, $\mid x_1 - x_2 \mid + \mid y_1 - y_2 \mid = 1$. Each grid (x, y), except the one at the edge of grid system, has four neighbors $(x, y+1)$, $(x-1, y)$, $(x, y-1)$, and $(x+1, y)$, with one in each of four directions: north, west, south, and east. $[x_1 : x_2, y_1 : y_2]$ represents a rectangle with four corner grids (x_1, y_1), (x_1, y_2), (x_2, y_2), and (x_2, y_1).

After many faulty nodes and misbehaving nodes are disabled from the collaboration, the rest of the nodes, also called trusted nodes, will form trust networks. According to the results presented in [16], when $R = \sqrt{5}r$, each trusted node can communicate with nodes in the neighboring grids. In each grid, one of the trusted nodes will be elected as the grid head. The rest of the trusted nodes in the same grid are called spare trusted nodes, or simply spare nodes. In this way, when each grid has its own head, the connectivity of all the heads and the coverage of the entire network can be guaranteed. Each head can monitor the status of the heads in neighboring grids. To minimize the coverage overlaps between the heads, we do not pursue the surveillance of diagonal neighboring grids for each head, which requires a larger communication range $R = 2\sqrt{2}r$ ($> \sqrt{5}r$). As a result, each move monitored by a head will be limited within two neighboring grids. The role of each head can be rotated within the grid. Each head, in charge of communication with heads of neighboring grids, knows the following information: (1) its grid location, and (2) the number of trusted

nodes in the grid and their locations.

It is noted that the grid partition with global information can ensure one and only one head existing in each grid territory. By only using the 1-hop neighborhood information, we can guarantee the existence of heads in any $r \times r$ square territory with a localized coverage scheduling algorithm, such as the one presented in [12]. After that, all the schemes presented in this paper can be extended easily under such a local view model. To make our movement control schemes clear, we only use the global partition model. Moreover, we describe the schemes in a synchronous, round-based system. All the schemes presented in this paper can be extended easily to an asynchronous round-based system. However, to simplify the discussion, we do not pursue the relaxation.

3 Movement Control Scheme

This section introduces our movement control scheme to fill in any vacant grid with trusted nodes. As a result, each grid will have its own head and the coverage problem will be solved. According to the method in which the vacant grid is detected, we have two implementations. Under the passive model, a vacant grid is detected only when a certain communication flow across it is blocked. After the detection, the replacement process is initiated to find a spare trusted node to move into the vacant area as well as the transaction of a data packet. The coverage will be complete whenever it is needed. However, when the move of a node is much slower than the transaction of packet between neighboring grids, the above scheme will cause a communication delay. To maintain a decent quality of communication service, in this paper, we also provide the active model. In this model, each grid head, say node u, will monitor its neighborhood. Whenever the neighboring grid becomes vacant, i.e., no head exists, the replacement process will be initiated immediately at u. In this way, the coverage is guaranteed. It is noted that only the 1-hop neighborhood is used in our approaches and the control scheme is implemented in a fully distributed manner to support any dynamic change, which makes the entire system more scalable.

3.1 Passive model

In this approach, we assume the path of communication flow crossing grids has been constructed (see Figure 2 (a)). The message packet will be forwarded hop by hop along such a path. Whenever a relay node cannot find its successor node to pass the next grid, that grid is identified as a vacant area. The replacement process will be initiated at that blocked node in which the forwarding of data packet holds.

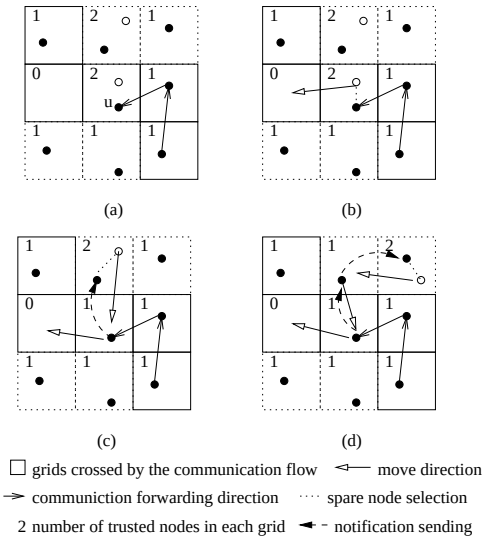


Figure 2. Passive control scheme. (a) Vacant grid detected, (b) moving control when a spare trusted node is found, (c) moving control when a spare trusted node is found in neighboring grid, and (d) cascading move.

First, that blocked node, say node u, will select one spare trusted node in its grid to move to the vacant grid (see Figure 2 (b)). If such a spare node cannot be found, u itself will move to the vacant grid. Before the movement, u will send a notification to the head of one neighboring grid, say node v. When v receives such a notification (in the next round), the above selection process will be repeated (see Figure 2 (c)). When the selection process of a spare neighboring trusted node fails, it triggers another notification process and then causes a so-called cascading move (see Figure 2 (d)). To ensure that the cascading process can converge quickly, the head in neighboring grid is selected randomly for sending the notification. Moreover, the notification is always sent to those grids with spare trusted nodes first. The whole cascading movement process of nodes is snake-like (see Figure 2 (d)). The details are shown in Algorithm 1.

Algorithm 1: Control scheme under passive model.

1. At a relay node u, the following replacement process will be initiated when u cannot find the successor node to cross the next grid along the path; i.e., a vacant gird in the forwarding direction is detected.
2. Find a spare trusted node in the grid of u, node v, to move into that vacant area before the next round starts.
3. If the above step fails, repeat the follows until the notified node u can find a spare trusted node v in the above

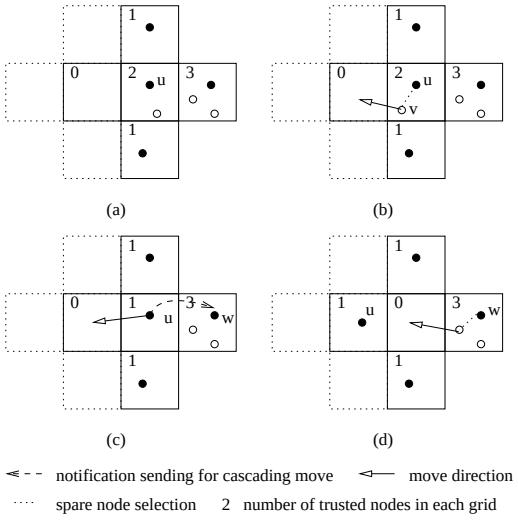


Figure 3. Active control scheme. (a) Vacant grid detected, (b) moving control when a spare neighboring trusted node is found, (c) notification sending for node replacement, and (d) cascading move.

step: (a) Select a neighboring grid other than the vacant one. A grid with spare trusted nodes are always preferred. (b) Send out the notification attaching such a selection to ask for a replacement of u. (c) Move u to the vacant grid before the next round starts; i.e., leaving the current grid vacant for cascading replacement.

3.2 Active model

In this approach, each grid area is monitored by not only the head of itself but also the heads of neighboring grids. A replacement process, unlike the one under the passive model which is initiated at a relay node of communication, will be initiated at a grid head u when u cannot find the head of one neighboring grid, i.e., a vacant neighboring grid is detected (see Figure 3 (a)). After that, the replacement process will continue and this head u will play the role of a grid head that receives the notification under the passive model. Figure 3 shows some samples. In Figure 3 (b), after the detection, node u finds its neighboring spare trusted node v in the grid. After the node selection, v will move into the vacant grid. When u cannot find such a node v, seen in Figure 3 (c), a notification will be sent to the head of one neighboring grid, say node w. At the same time, node u will move into that vacant grid and leave its own grid vacant. As mentioned before, the grid with spare trusted nodes is always selected first. Then, when the grid head w receives

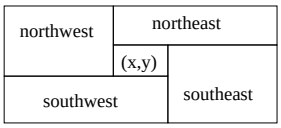


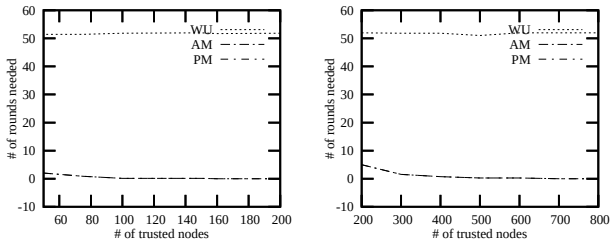
Figure 4. Region partition.

the notification in the next round, a spare trusted node will be selected to replace node u (see Figure 3 (d)).

When a grid head detects a vacant neighboring grid, it initiates a replacement process. Because only the 1-hop neighborhood is used, it may not know if this hole is also detected by other heads. In our active model, to ensure that this hole will be covered in any situation, each grid head will start the replacement process individually. In other words, the vacant grid can be detected by more than one neighboring grid. In many cases, the hole will be filled by more than one trusted node. Its advantage is to have more backup nodes in the event that this hole area is very critical and node failures or attacks will occur frequently.

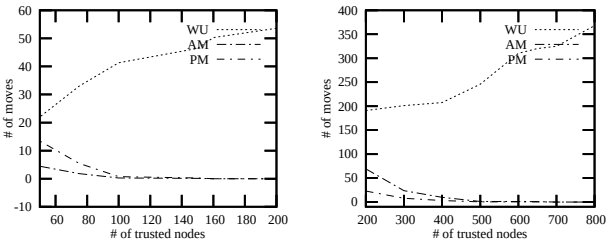
However, the existence of multiple replacement processes may cause conflicts when the paths of two cascading moves intersect. We partition the whole grid system into four parts for a given vacant grid (x, y): northeast region $[x : n-1, y+1 : m-1]$ which contains the northern neighboring grid $(x, y+1)$, northwest region $[0 : x-1, y : m-1]$ which contains the western neighboring grid $(x-1, y)$, southwest region $[0 : x, 0 : y-1]$ which contains the southern neighboring grid $(x, y-1)$, and southeast region $[x+1 : n-1, 0 : y]$ which contains the eastern neighboring grid $(x+1, y)$, as shown in Figure 4. To avoid a conflict, the cascading move initiated at each neighboring grid is limited in its corresponding region. When the grid head detects a neighboring vacant grid, with the location information of these two neighboring grids, the corresponding region area can easily be determined. To guide the selection of the successor head node in a cascading move, the area information of such a region will be attached into each notification message in this replacement process.

Another issue that we face is overreaction. In the cascading movement process, after the grid head moves out, a grid is vacant until the new trusted node moves in. During this period, the vacancy can be detected by grid heads in neighboring grids. In such a case, the replacement processes initiated by those grid heads are unnecessary and the corresponding initialization action is overreaction. In our approach, the cascading movement is triggered by sending the replacement notification to a selected grid head. When the heads in neighboring grids also receive this notification, they know that this vacancy is just a temporary status. In this way, the unnecessary initialization can be avoided. The details are shown in Algorithm 2.



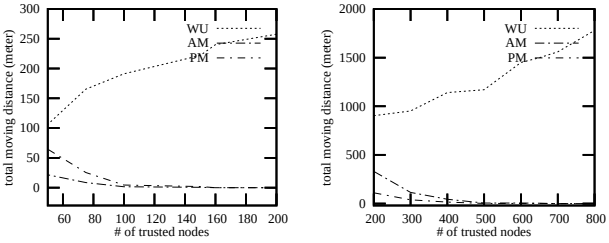
(a) 5×5 grid system. (b) 10×10 grid system.

Figure 5. Converging rounds needed.



(a) 5×5 grid system. (b) 10×10 grid system.

Figure 6. Total moves needed.



(a) 5×5 grid system. (b) 10×10 grid system.

Figure 7. Total node moving distance (meter).

Algorithm 2: Control scheme under active model.

1. For any grid head u that detects a vacant neighboring grid, if no notification is received in the previous round from that area (to avoid overreaction), a replacement process is initiated after the corresponding region area Δ is determined.

2. For any grid head u that has started the replacement process or that is notified in the cascading replacement process, find one of neighboring spare trusted nodes in its grid, say node v, to move into that neighboring vacant grid before the next round starts.

3. If such a node v cannot be found, select any neighboring grid that is other than the vacant one but still in region Δ. Then, send out the notification for the replacement of u, attaching the information of Δ and such a selection. It will always select the one with spare trusted node(s) first. After that, move u to the vacant grid before the next round starts.

---

4 Simulation

In this section, we verify the improvement of our control scheme, under both the passive model (PM) and the active model (AM), and compare it with the best result known to date, as seen in [14] (WU). The results show that our snake-like cascading movement will successfully cover any hole while keeping the cost substantially lower. For the number of randomly deployed sensors with communication range $R = 10m$, we determine the grid size $4.4721m \times 4.4721m$ and then form the virtual grid system [16] in the target area. After the deployment, we randomly disable some nodes from the trust collaboration in order to simulate the malicious attacks. Then, the rest of the nodes are trusted nodes and form the trust networks. One of trusted nodes in each grid (if any) will be elected as the head. The connectivity and coverage of such networks can be guaranteed when each grid has its own head. However, by the simulated attacks, the hole will occur in the trust networks. Thus, we apply schemes WU, PM, and AM to fix the coverage problem. At last, we test the number of rounds needed (i.e., the converging speed) in these schemes and compare their performances. We also test the cost of these schemes in terms of the total moving distance and the number of total node moves. It is noted that each move of node u from one grid to its neighbor will randomly select the destination location $L'(u)$.

The turnable parameters in our simulation are as follows. **(1) Number of grids** $n \times n$. Once the size of each grid has been decided, the surveillance area of security applications will determine the size of grid system needed. We use 5×5 and 10×10 in the simulation. **(2) Number of sensors** N **in the trust networks**. In [14], it has been mentioned that the control scheme can be applied to the network with at least $4n^2$ nodes. Therefore, we deploy $k \times n^2 (k \geq 4)$ sensors and only select those cases when N's value is in the range from $4n^2$ to $4n^2 \times 2$. We also include cases of under $4n^2$ sensors to check the robustness of our approach. It is noted that when the number of nodes is larger than $8n^2$, there is no vacant grid occurring in our random attack simulation and no need for any node movement.

Figure 5 shows the number of rounds needed in schemes WU, AM, and PM in the cases when 5%∼20% trusted nodes survive from the attacks. We also show the corresponding costs in these cases: the number of node moves in Figure 6, and the total node moving meters/distance in Figure 7. Results can be summarized as follows: **(1)** It is claimed in [14] that among all the existing movement-assisted balancing methods, scheme WU has the best performance insofar as process converging is concerned. However, this method requires a scan prior to node movement. Considering the communication costs in its scan process, the number of rounds needed, i.e., the converging speed, is $O(n)$. Both PM and AM schemes use the 1-hop neighborhood and will converge much faster than scheme WU. Furthermore, our results show that there is no big difference between schemes PM and AM in the converging speed. **(2)** Due to the localized adjustment in replacement process, our snake-like replacement processes, schemes PM and AM, are more scalable than scheme WU, as shown in Figure 5. **(3)** It is also claimed in [14] that scheme WU has the lowest cost as far as the total moving distance and the total number of moves are concerned. However, it requires an adjustment in the entire network. By limiting the adjustment within a local area, the number of total moves and the corresponding total moving distance can be reduced greatly in our snake-like replacement process (in both PM and AM schemes). The replacement process is initiated in the PA scheme only when it is needed. This being the case, the PA scheme incurs the lowest cost. However, it requires the replacement process in order to move a spare node as quickly as the communication propagation. Scheme AM finds a trade-off between this hard constraint and the cost, with its performance in converging speed still acceptable. Therefore, it is more practical.

5 Conclusion

In this paper, we have presented a snake-like replacement process to cover the surveillance holes of WSNs where all sensors deployed in certain sensing areas are disabled from the collaboration. As a result, the connectivity and coverage of WSNs can be guaranteed and such networks become applicable for security applications, even when the working status of nodes changes dynamically. Its implementation under two different models are discussed: one under the passive model and the other under the active model. In our methods, only the 1-hop neighborhood is used and the adjustment of nodes can be controlled within the local area. The experimental results show the proposed method to be scalable and that it converges quickly with a minimized cost. In future work, the energy consumption will be considered in the node adjustment so that the lifetime of the complete coverage can be extended.

References

[1] A. Agah, S. Das, K. Basu, and M. Asadi. Intrusion detection in sensor networks: A non-cooperative game approach. *Proc. of ICWN'06.* 2006, pp. 29-36.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*. Vol. 38, 2002, pp. 393-422.

[3] S. Basagni, K. Herrine, D. Bruschi, and E. Rosti. Secure pebblenets. *Proc. of MobiHoc'01*. 2001, pp. 156-163.

[4] S. Buchegger and J. Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. *Proc. of PDP'02.* 2002, pp. 403-410.

[5] S. Buchegger and J. Boudec. Performance analysis of the CONFIDANT protocol: Cooperation of nodes fairness in dynamic ad-hoc networks. *Proc. of MobiHoc'02.* 2002, pp. 226-236.

[6] S. Capkun, L. Buttyan, and J. Hubaux. SECTOR: Secure tracking of node encounters in mult-hop wireless networks. *Proc. of the ACM Workshop on Security of Ad Hoc and Sensor Networks*. 2003, pp. 21-32.

[7] J. Carle and D. Simplot-Ryl. Energy-efficient area monitoring for sensor networks. *Computer*. Vol. 37, Issue 2, Feb., 2004, pp. 40-46.

[8] A. Howard, M. Mataric, and G. Sukhatme. An incremental self-deployment algorithm for mobile sensor networks. *Autonomous Robots, Special Issue on Intelligent Embedded Systems*. Vol. 13, No. 2, Sept. 2002, pp. 113-126.

[9] F. Hu and N. Sharma. Security considerations in ad hoc sensor networks. *Ad Hoc Networks*. Vol. 3, No. 1, 2005, pp. 69-89.

[10] Y. Hu, A. Perrig, and D. Johnson. Wormhole detection in wireless ad hoc networks. *Technical report, Rice University Department of Computer Science*. June 2002, available at http://citeseer.ist.psu.edu/hu02wormhole.html.

[11] C. Karloff and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad Hoc Networks*. Vol. 1, Nos. 2-3, 2003, pp. 293-315.

[12] D. Tian and N. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. *Proc. of the 1st ACM Workshop on Wireless Sensor Networks and Applications*. 2002.

[13] G. Wang, G. Cao, and T. Porta. Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing.* Vol. 5, No. 6, June. 2006, pp. 640-652.

[14] J. Wu and S. Yang. SMART: A scan-based movement-assisted sensor deployment method in wireless sensor networks. *Proc. of INFOCOM 2005.* Vol. 4, March 2005, pp. 2313-2324.

[15] W. Xu, K. Ma, W. Trappe, and Y. Zhang. Jamming sensor networks: Attack and defense strategies. *IEEE Network.* Vol. 20, No. 3, 2006, pp. 41-47.

[16] Y. Xu and J. Heidemann. Geography-informed energy conservation for ad hoc routing. *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'01).* 2001, pp. 70-84.

[17] Y. Zou and K. Chakrabarty. Energy-aware target localization in wireless sensor networks. *Proc. of the First IEEE International Conference on Pervasive Computing and Communications (PerCom'03).* 2003, pp. 60.