



## Self Organizing Sensor Network to Enhance Event Coverage

<sup>+</sup>S. Indu, <sup>^</sup>Venu Kesam, <sup>^</sup>Santanu Chaudhury, <sup>+</sup>Asok Bhattacharyya

<sup>+</sup>Delhi Technological University, Delhi

<sup>^</sup>Indian Institute of Technology, Delhi

Emails: [s.indu@rediffmail.com](mailto:s.indu@rediffmail.com)

---

*Submitted: October 16, 2010    Accepted: January 10, 2010    Published: March 1, 2011*

---

*Abstract- We are proposing a self deploying mobile sensor network which is empowered with event based relocation of redundant sensors for enhancing the quality of event sensing. Energy efficient Cell quorum based protocol is used for communication between event location and redundant sensors. Computationally light cascaded reorganization of sensor is suggested for relocation of sensors. The proposed method provides good coverage as it dynamically relocates the sensors either for avoiding coverage hole or for improving the quality of event sensing.*

**Index terms:** Cell Quorum based protocol, redundant sensors, event detection, and relocation

### I. INTRODUCTION

Sensor deployment has received considerable attention recently. When the environment is unknown or hostile such as remote harsh fields, disaster areas and toxic urban regions, sensor



cascaded movement of sensors with minimum moves make the proposed method efficient and reliable.

## II. RELATED WORK

Many research works are being directed towards overcoming the common issues of sensor networks like self organizing, energy efficiency, event detection and coverage etc. and a considerable amount of research is being carried out on sensor deployment issue itself. A centralized virtual force based mobile sensor deployment algorithm (VFA) [4] [5] proposed by Y. Zou later modified by Jiming, combines the idea of potential field and disk packing [6]. This algorithm uses a powerful cluster head, which will communicate with all the other sensors, collect sensor position information, and calculate forces and utilizes these forces to position each sensor. Also this method assumes that the sensor node knows its initial location.

Zak Butler and et. al. in their work "Event Based Motion Control" [7], discussed enhancement of coverage of mobile sensor network by replacing damaged sensors. They discussed different methods which were computationally complex and may not even converge for some probable cases. Their main assumption was complete initial coverage which may not be possible for all practical cases such as herd monitoring, environmental monitoring etc.. Further in their work "controlling mobile sensors for monitoring events with coverage constraints" [8], they used veroni diagram to find out coverage holes, which is computationally intense. The algorithm proposed by Bin Zhang [9], randomly partitions space into sufficiently small neighborhoods at each iteration. Within each neighborhood are distribution process directed by a cluster head is enacted. As this is cluster based method it has high message complexity. Yee Ming Chen et.al [10] has proposed a similar work using fire tracker as an agent for event identification. It is assumed that sensor knows it's relative location with respect to neighboring nodes. They also suggest movement of sensors in coalation with other sensors to enhance the effectiveness of extinguishing the fire, but the method used to move along with other sensors is not clear. The challenging issues of self organization is selective relocation of sensors, energy efficiency, event detection etc. Zihui Ge and et.al. proposed match making in their work "Matchmaker" [11] they developed a pub/sub strategy so that nodes can publish their facilities / their needs on a common platform and can be shared effectively. Guiling Wang and et. al. used rectangular grid quorum



### III. PROBLEM STATEMENT

The basic issues of mobile sensor networks are localization, coverage and energy efficiency. The global locations of each sensor node can be easily obtained by using GPS enabled nodes, but no literature is available to calculate its relative position in the field. Proper deployment algorithm can provide good coverage, but mobile sensor networks are more susceptible to coverage holes as well as sparsely covered event location. The time taken for healing a coverage hole using iterative algorithm is very high and hence is not suited for event based relocation. We developed a spreading algorithm which computes relative positions of sensor nodes dynamically, identify event location and redundant sensors, fast and automatically heals coverage holes by doing minimum moves. Cell Quorum based communication protocol is used for communication between redundant sensors and sensor node at event location. Optimal relocation of redundant sensors is accomplished by doing minimum moves as the sensor nodes are aware of their relative location in the field. The energy efficient protocol and minimum movement of sensor nodes helps to improve the life of the network

#### A. Spreading Algorithm

The spreading algorithm provides complete coverage [20]. A brief description is given in this paper for proper understanding. This algorithm needs the area of the sensing field as reference for localization of sensor nodes. This can be given as the range of Latitude and Longitude in case of an open field or as coordinates in case of a closed field. The coverage space is logically partitioned into several disjoint and equal-sized hexagonal cellular regions. The size of the hexagon depends on the coverage capacity of sensor used. Each cell is then assigned a unique Cell-id ( $C_x, C_y$ ) relative to the field as shown in fig. 1. For the given ( $x, y$ ) coordinates ( $(x, y)$  coordinates of the sensor node), the corresponding cell-id ( $C_x, C_y$ ) of the node is calculated by comparing the node coordinates with the coordinates of the central point of each cell. The coordinates of each node can be calculated either from satellite (GPS) or from 3 Beacons placed locally as shown in fig. 2.

The nodes are made 'ON' randomly. Once a node is 'ON' it will broadcast its cell ID and coordinates, then it will listen from other nodes. If it is not hearing from any other node of the same cell, it will elect itself as the Cell Head otherwise it will register itself as a slave node. Once a head node is elected then all the other nodes in that cell will become slaves and will



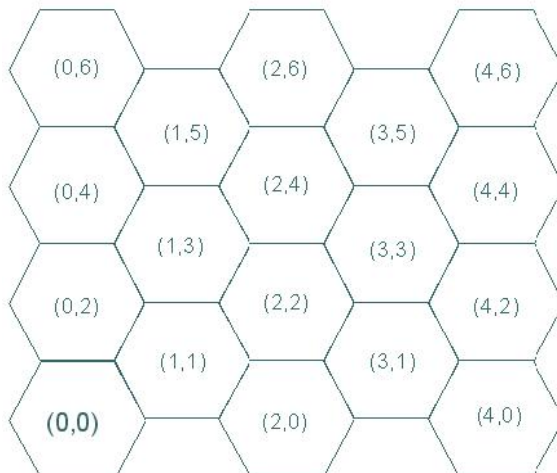


Fig-1 Hexagonal Space Tessellations

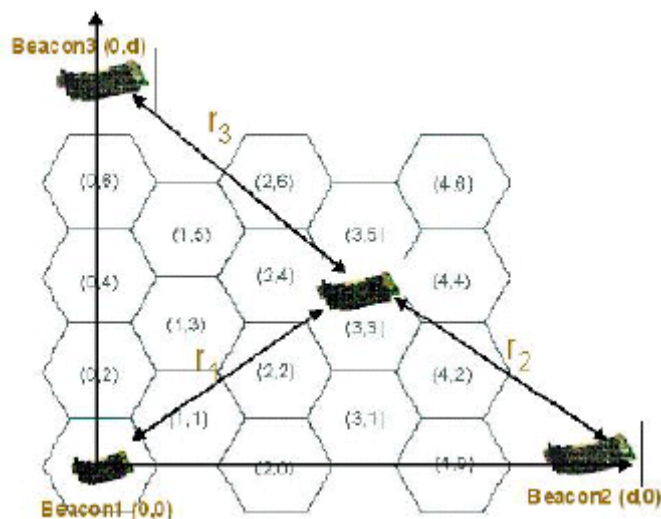


Fig-2 Localization

#### IV SELF ORGANIZING SENSOR NETWORK

##### A. Spreading Algorithm

Let  $N$  be the total number of cells in the given area whose side is given by "S"  $O_i$  be the occupancy at a iteration  $i$  with a given probability of electing head be  $P_{head}$

$$O_i = \sum_{k=1}^N C_i^k \tag{1}$$





Hence for large value of  $k$  the  $P_{\text{head}}$  will become 1 and hence at least one node will become head, then that head node will go through the steps 5-9 which will fill the empty cell hence increasing the occupancy.

If the number of nodes is less than number of cells, the occupancy  $O_i$  will remain constant and it will not satisfy (2) the algorithm will not converge. The nodes will always be in motion, providing dynamic coverage.

*a. Algorithm:*

$H_i^k$  represents the head node of a cell  $k$  at  $i^{\text{th}}$  iteration and  $P$  is the set of current node locations.  $A_i^k$  is the adjacent cell information of the head cell  $H_i^k$  and  $V_i^k$  are the victim nodes that are targeted to the empty neighboring cells of the cell  $C_k$  by the head node  $H_i^k$

1. Partition the sensing field into small sub cells  $C_k$  having a regular pattern (hexagonal);
2.  $i=0$ ;
3. While termination conditions are not satisfied do;
4. Select a cell head  $H_i^k$  from the nodes which are moved to a cell  $C_k$ ;
5. Each node  $n$  within a cell sends position  $P(n)$  to the cell head;
6. Each cell head  $H_i^k$  learn the neighboring cells information and constructs the adjacency list  $A_i^k$
7. Each cell head  $H_i^k$  selects the victim nodes  $V_i^k$  that are to be sent to the neighboring cells  $C_j = C_{\text{nbrs}}$
8. Assign each victim node a new position;
9. Notify the adjacent cells with positions of the victim nodes;
10. All the Victim nodes will move to the new cells;
11.  $i = i + 1$



Cell Quorum based communication protocol is used to advertise the location of redundant sensors and request for more sensors at the event location by the respective cell heads. By organizing cells as quorums, each advertisement and each request can be sent to a quorum of cells. Due to the intersection property of quorums, there must be a cell which is the intersection of the advertisement and the request. The cell head will be able to match the request to the advertisement. A simple publisher quorum and subscriber quorum can be constructed by choosing all the nodes along the same row and column. For example, as shown in Fig.3., suppose cell (1,5) has redundant sensors, it sends the advertisement to cells in a row ((1,5), (3,5), (5,5), (7,5), (9,5)) and a column ((1,3), (1,1), (1,-1)). When cell (4,0) is looking for redundant sensors, it only needs to send a request cells ((4,0), (3,1), (2,2), (1,3)) and ((3,-1), (4,0), (5,1)). The intersection node (1, 3) will be able to match the request to the advertisement. Suppose  $N$  is the number of cells in the network. By using this quorum based system, the message overhead can be reduced from  $N$  to  $\sqrt{N}$ . The message overhead is very low compared to flooding. We can further reduce the message overhead by observing the specialty of our problem. This can be even extended to multiple event detection.

### C. Sensor movement;

Having obtained the location of the redundant sensors, we need to determine how to move the redundant sensors to the target location (destination). It is very important that the movement should be completed within the time frame of the event i.e. before the event dies out. We should look for cascaded movement instead of direct movement as it will reduce power consumption as well as time frame.



The set of cascading nodes for relocation and their departure time together is defined by a cascading schedule. For example, in fig.8,  $S_3(t_3)$ ,  $S_2(t_2)$ ,  $S_1(t_1)$ ,  $S_0$  is a cascading schedule, which can be used to increase the coverage of the event. The cascading schedule should ensure that the number of movements should be minimum and it should happen well within the required time of event. Each movement should also consider the restriction of recovery delay of individual node. i.e. movement should satisfy both conditions. As each sensor knows its cell ID and its coordinates relative to the field, the number of movements can be minimized by moving along the intermediate diagonal cells (by incrementing or decrementing both coordinates by 1 depending on the event location and redundant sensor location) till the node reaches the corresponding column/row of the event location and there after moving along the cells of the column/row of the event location. If the event and redundant sensors lie on same row or column, the schedule should be made along the corresponding row/column. This method not only minimizes the movement but also avoids collision of sensors. If there are redundant sensors in more than one cell, the nearest one is identified from its location. The sensor at the event location initiates the formation of cascading schedule by broadcasting the allowable time delay, its cell ID and cell ID of redundant sensor. The neighboring cell head will respond if it is on the diagonal path or in the same row/ column of the redundant sensor.

*d. Algorithm for cascaded movement.*

$E_i$  represents the head node of an Event cell and  $R_m j$  is the head cell of one of the 'm' cells having redundant sensors. The event cell head will execute cascaded movement algorithm.

1. Calculate the Euclidean distance between the redundant cell head and event cell head, and select the redundant cell head with minimum distance;
2. Compare the cell-ID of cell having redundant sensors and event cell;
3. Decide movement path;
4. Execute the movement;



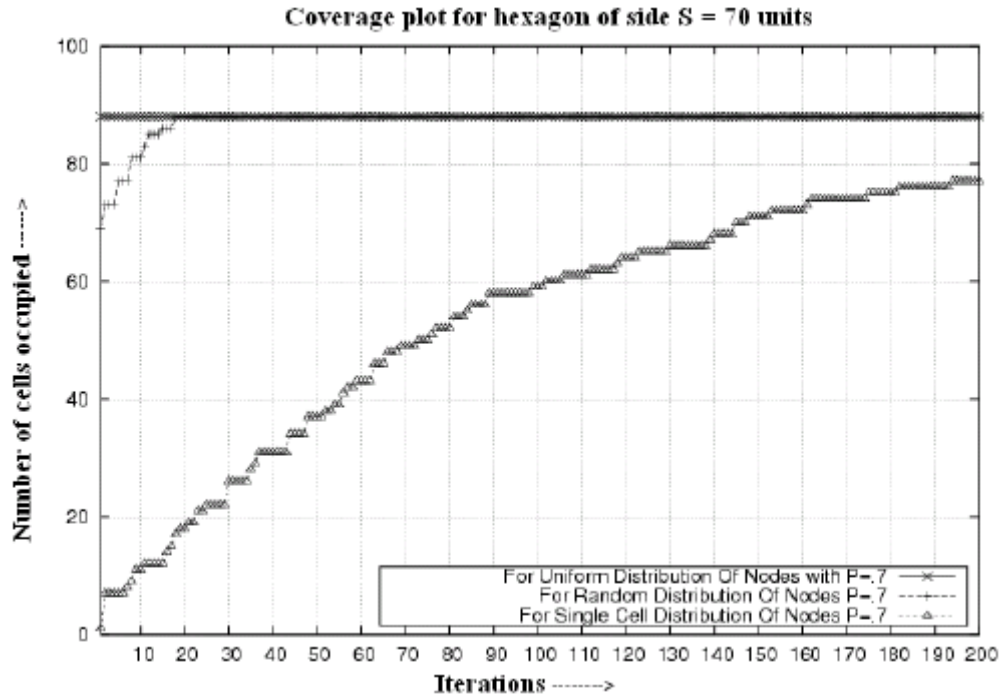


Fig. 6. Coverage plot

B. Event detection and Sensor relocation

Simulations were carried out to find out average battery remaining with variable number of nodes for comparing with Naive protocol. We compared the power consumed by the sensor nodes and

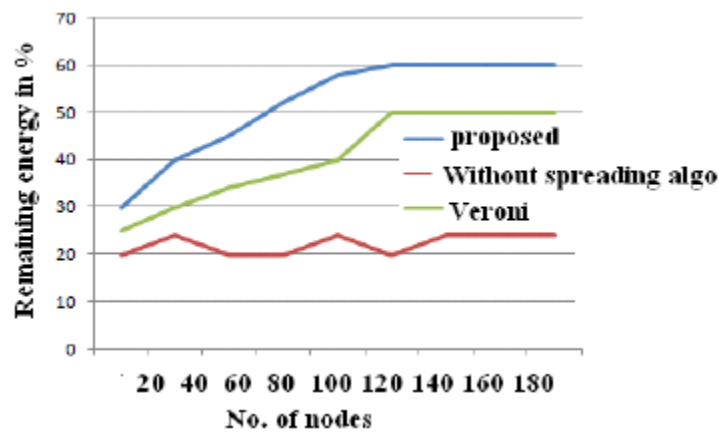


Fig.7. Plot of remaining battery vs number of nodes





response time for relocation with self organizing algorithm (Veroni diagram based) and also with sensor relocation without spreading algorithm for deployment and found that with increase in no. of nodes the average battery remaining is increasing as shown in fig. 7 and fig. 8 . We assumed energy consumption based on distance travelled by the node. Single event detection and sensor relocation is simulated and shown in fig. 9(a) and fig. 9(b). The deployment algorithm gave complete coverage and the redundant sensors moved to the event location.

## VI. EXPERIMENTL EVALUATION

For experimental evaluation a sensor node, a range of sensors for monitoring various phenomena and a mobility platform are used. The first two have been purchased from vendors where as the mobility platform was developed in the lab. The cricket motes from Crossbrow are used, as it can be programmed for both sensing computational task as well as location awareness. MPR410 Sensor Nodes with MIB510 programming board as shown in fig: 10 are used for validation. Three separate software were used to implement the system.

The firmware is uploaded on three motes which served as Beacons. This firmware was slightly modified and used as a Listener. Finally a single node was programmed as a packet forwarder for the coverage panel application. The BeeBots are used (as shown in fig. 11) to provide mobility or the nodes. BeeBots have been inspired from CotsBot [21] and have been similarly built by modifying toy car as shown in fig. The BeeBots are configured using TinyOs [22] for interfacing with PIC microcontroller for providing mobility.

Three Beacons and four mobile nodes were used. We developed a coverage panel in Java for displaying, in real time, the actual location of sensor nodes. 3 cricket beacons were used for localization of nodes which are represented as black dots and mobile nodes are represented as red nodes. The distribution of nodes after running deployment algorithm is shown in fig.13. We used nesC to embody the structuring concepts of TinyOS operating system. Three separate software were used to implement the system. Firstly the firmware supplied was uploaded on 3 motes which served as Beacons. This firmware was slightly modified to use as a node which ran our model. Finally a single node was programmed as a packet forwarder for coverage panel. This packet forwarder acts as an interface between sensor nodes and serial port of computer so that we can see the respective positions of all sensors in the field. We programmed the nodes to handle

self deployment, localization, event detection and relocation themselves. Fig. 14 and fig.15 gives the intermediate location and final positions of mobile nodes after implementation of DHM based selforganizing algorithm.

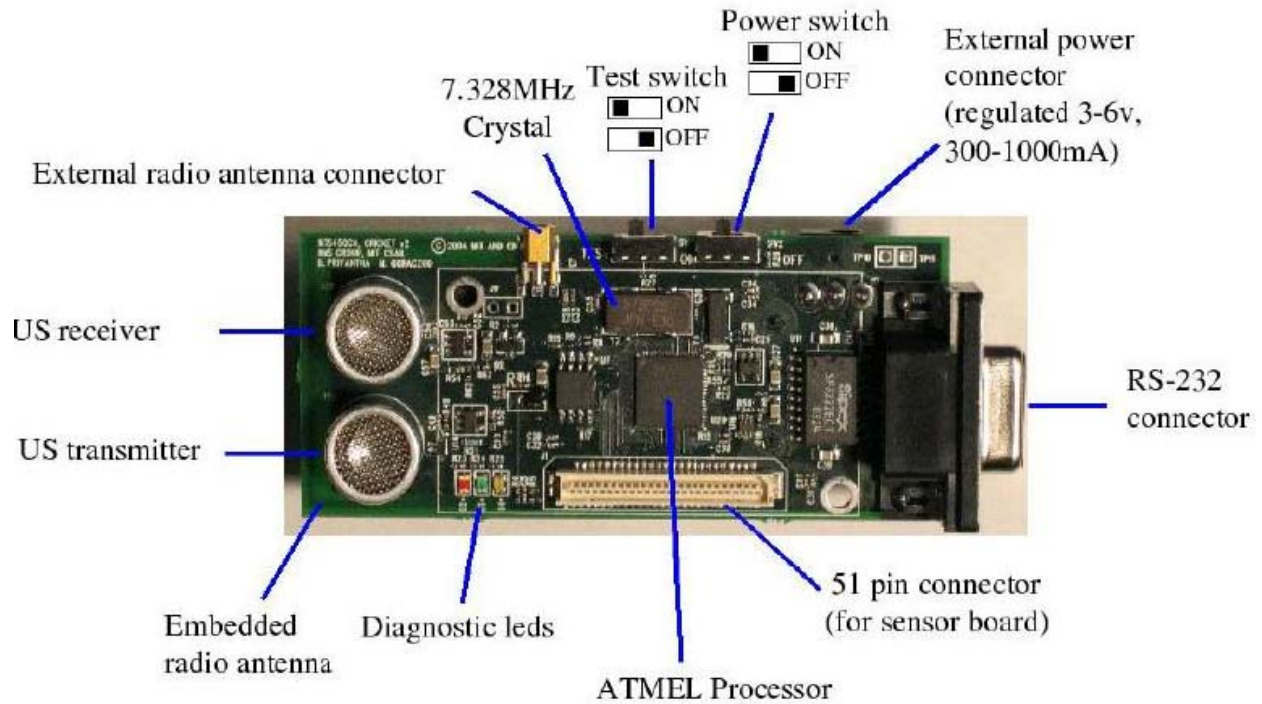


Fig. 10 MPR410 Sensor Node

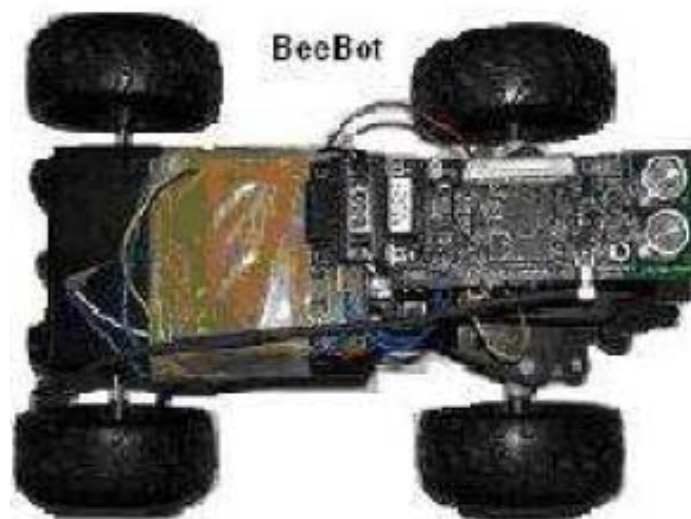


Fig.11 A sample Beebot

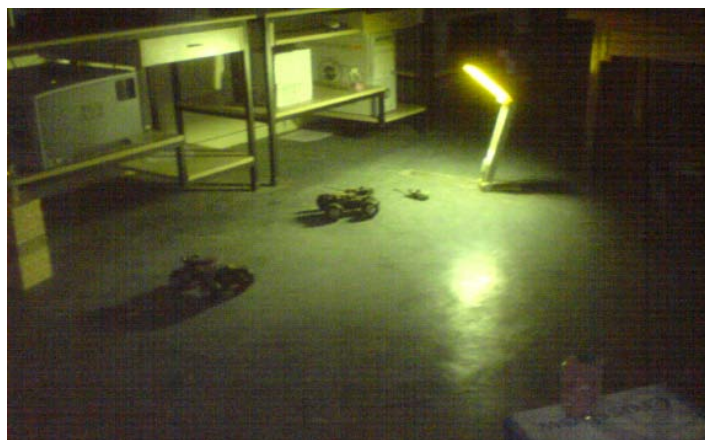


Fig.12. Experimental Setup

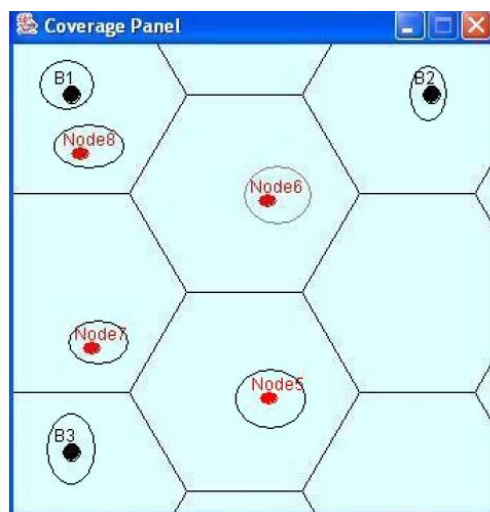


Fig. 13 The locations of the sensors after running deployment algorithm

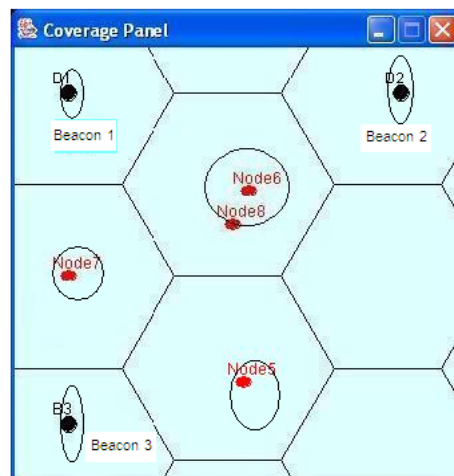


Fig. 14 Intermediate location of sensor nodes

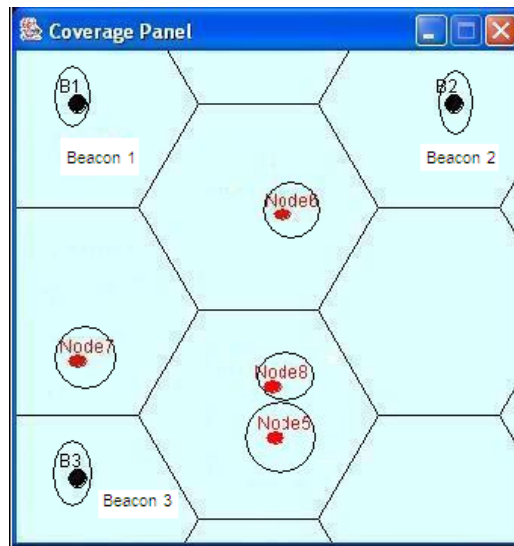


Fig.15. Final location of nodes.

## VII. CONCLUSION

We have presented a novel distributed Self organizing sensor network which heals coverage holes and provides accurate information by increasing the coverage at event location using energy efficient relocation of redundant sensors. The simulation results show the correctness of the algorithm. The algorithm is computationally light and scalable. For simulation purpose we have developed a java simulator to display the position of the individual sensor nodes.

We have also verified the simulated results using experimental evaluation. We programmed the motor of remote car and interfaced it with the sensor node for mobility. Then we run the self organizing algorithm for deployment and sensor relocation. The results are self explanatory. This work will be instrumental for sensing disaster fields which are inaccessible for human beings.

## REFERENCES

- [1]. A. Howard, M. J. Mataric, and G. S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks," *Autonomous Robots Special Issue on Intelligent Embedded Systems* 13(2), Sept 2002, pp. 113--126
- [2] J. A. Paradiso and T. Starner, "Energy Scavenging for Mobile and Wireless Electronics," *IEEE Pervasive Computing* January, pp. 1568 – 1573, 2005.

- [3] K. M. Krishna, “A Surveillance System based on Multiple Mobile Sensors,” Proceedings of FLAIRS, Special Track in Sensor Fusion, pp 1-4, 2004.
- [4] Y. Zou and K. Chakrabarty, “Sensor Deployment and Target Localization based on Virtual Forces,” ACM Transactions on Embedded Computing Systems, Vol. , No. , 20, Pages 1–30.
- [5] J. Chen, S. Li, and Y. Sun, “Novel Deployment Schemes for Mobile Sensor Networks,” In International Journal SENSORS Volume 7, pp. 2907 – 2919, 2007.
- [6] J. H. Reif and H. Wang, “Social potential fields: A distributed behavioral control for autonomous robots,” Robotics and Autonomous Systems, vol. 27, pp. 171 – 194, 1999.
- [7] Z. Bulter and D. Rus, “Event based motion control for mobile sensor network,” IEEE Pervasive Computing, Volume 2 Issue 4, October 2003, pp 34-42.
- [8] Z. Bulter and D. Rus, “Controlling Mobile Sensors for Monitoring Events with Coverage Constraints,” In Proc. of the IEEE International Conference on Robotics, pp. 1568 – 1573, 2003.
- [9] B. Zhang and G. Sukatame, “Controlling ensor density using mobility, ” Proceedings of 2nd IEEE workshop on Embedded Networked Sensors, pp. 141 – 149, 2005.
- [10] Y. M. Chen and S.-H. Chang, “Purposeful Deployment via Self-Organizing Flocking Coalition in Sensor Networks, ” International Journal of Computer Science and Applications volume 4 issue 3, pp. 84 – 94, 2007.
- [11] J. K. Z. Ge, P. Ji and D. Towsley, “Matchmaker: Signaling for dynamic publish/subscribe applications,” Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP’03) 1092-1648/03 \$17.00 © 2003 IEEE
- [12] G. Wang, G. Cao, and T. L. Porta, “Sensor relocation in mobile sensor network,” In Proceedings of 23rd International Conference of IEEE Communications Society, pp. 1– 11, March 2004.
- [13] J. Teng, T. Bolbrock, G. Cao, and T. L. Porta, “Sensor Relocation with Mobile Sensors Design Implementation and Evaluation,” 1-4244-1455-5/07/2007 IEEE, 2007.
- [14] W. L. Yeow, C. K. Tham, and W. C. Wong, “A novel target movement model and energy efficient target tracking in sensor network,” 0-7803-8887-9/05/IEEE, 2005.
- [15] N.A.Vasanthi, S.Annadurai, and S. Krishna, “Energy saving schedule for target tracking sensor networks to maximise the network life time,” 0-7803-9575-1/06/IEEE, 2006.

[16] H. Rababaah and A. Shirkhodaie, "Gaurd duty alarming technique: A novel scheduling approach for target tracking in large scale distributed sensor network," 1-4244-1160-2/07/IEEE, 2007.

[17] R. Mathur, M. K. Sharma, A. Misra, and D. Baveja, "Energy-Efficient Deployment of Distributed Mobile Sensor Networks Using Fuzzy Logic Systems," In proceedings of 2009 International Conference on Advances in Computing Control and Telecommunication Technologies, pp. 121 –125, 2009.

[18] Antti Ropponen, Matti Linnavuo, Raimo Sepponen "LF Indoor Location and Identification System" International Journal on Smart Sensing and Intelligent Systems, vol. 2, no. 1, march 2009, pp 94-117.

[19] Jiann-Liang Chen, Yu-Ming Hsu and I-Cheng Chang "Adaptive Routing Protocol for Reliable Sensor Network Applications" International Journal on Smart Sensing and Intelligent Systems, vol. 2, no. 4, December 2009, pp 515-539

[20] S.Indu, Venu Kesam and Asok Bhattacharyya "SELF Deployment OF Mobile SensorNetwork for optimal coverage" International Journal of Engineering Science and Technology Vol. 2(7), 2010, 2968-2975

[21]"The cotsbot project homepage," Website,  
<http://wwwwsac.eecs.berkeley.edu/projects/cotsbots/index.html>.

[22] P. Levis, "Tinyos programming," Website, February 2006, <http://www.tinyos.net/>.