Energy Conserving Movement-Assisted Deployment of Ad hoc Sensor Networks

Hamid Mousavi, Amir Nayyeri, Nasser Yazdani, and Caro Lucas

***Abstract*— Sensor network deployment is very challenging due to hostile and unpredictable nature of usage environments. In this letter, we propose two methods for the self-deployment of mobile sensors. The first one is a randomized solution that provides both simplicity and applicability to different environments. Inspired by simulated annealing, it improves both speed and energy conservation of the deployment process. The other method is suggested for environments where sensors form a connected graph, initially. At the cost of this extra limitation, we gain considerable improvements.**

***Index Terms*— Ad hoc sensor network, deployment, simulated annealing, heuristic algorithms.**

I. Introduction

SENSOR networks, that are composed of tiny and resource constrained computing devices, have been widely deployed for monitoring and controlling applications in physical environments [1]. Due to the unfamiliar nature of such environments, deployment of sensors has become a challenging problem [2]. To resolve the problem, mobility has been suggested for self-deployment of sensors.

Diverse ideas have been proposed for self-deployment of sensors. In [3], the burden of determining each node's destination is put on head clusters which results in a centralized mechanism and emergence of failure points. Considering each sensor as a magnetic particle, [4][5] have reduced the problem to finding the equilibrium state of a magnetic field. Nevertheless, the convergence speed is truly low because of the sequential manner of these methods. Similarly, some articles such as [4][6][7] have introduced parallel virtual forces. However, as indicated in [6], the rate of convergence could not reach an acceptable point. In addition, [6] has introduced methods based on Voronoi diagram that could statistically overcome the others, but they suffer from either low convergence rate or high energy consumption.

In this letter, we propose Stochastic Deployment Routine (SDR), a randomized distributed algorithm for sensor networks self-deployment, motivated by simulated annealing as a Monte Carlo scheme [8]. A second method also presented in this paper, One Step Deployment (OSD), could register optimum convergence rate for those environments in which the sensors form a connected graph initially. In addition, Exchanging Goals of Sensors (EGS) is proposed to reduce the average movement of devices. All proposed methods run completely distributed to enhance the fault tolerance and scalability of the system. The experimental results confirm superiority of both SDR and OSD in total movement and convergence time comparing with the existing techniques.

SDR, OSD, and the heuristic extensions are thoroughly explained in Section 2. Section 3 is devoted to experimental and analytical analyses of the proposed methods. Finally, Section 4 concludes the article.

Manuscript received October 5, 2005. The associate editor coordinating the review of this letter and approving it for publication was Ezio Biglieri.

H. Mousavi is with the CE Department, Sharif University of Technology, Tehran, Iran (e-mail: h_mousavi@ce.sharif.edu).

A. Nayyeri, N.Yazdani, and C. Lucas are with the ECE Department, University of Tehran, Tehran, Iran (e-mail: a.nayyeri@ece.ut.ac.ir, yazdani@ut.ac.ir, lucas@ipm.ir).

Digital Object Identifier 10.1109/LCOMM.2006.04023.

II. Self-Deployment Schemes

We propose two different methods for movement assisted self-deployment of sensors. The formal goal of the suggested schemes is to cover a predefined area with both high convergence speed and low energy consumption.

A. Stochastic Deployment Routine (SDR)

Motivated from Simulated Annealing, SDR is the first proposal of this paper. For running SDR, the time vector is divided into equal-sized partitions, called epochs. At each epoch, the sensors determine their new destinations stochastically, and start moving toward them. The ever-decreasing expected value of the moving distance, which simulates the annealing process, guarantees the termination of SDR.

Assume that the field of sensor deployment is an $X \times Y$ rectangle. The movement of a sensor at epoch t takes place in the *Moving Rectangle* (MR_t) probabilistically with uniform distribution. The size of MR at a given time t, is determined by two constants, p and p_0, associated with t. Starting with MR_0 as a $(X.p_0) \times (Y.p_0)$ rectangle, MR shrinks exponentially by the factor p. It means that at epoch t the size of MR_t is $(X.p_0.p^t) \times (Y.p_0.p^t)$; as shown in Fig. 1. It is worth mentioning here that considering Gaussian distribution with decreasing variance can be an extending idea.

In addition to the passed time from the deployment beginning, the nodes density around a sensor also affects its rectangle size. This important factor is applied in SDR using some heuristic rules. For example, having no adjacent node, the sensor will cancel its next epoch movement. Similarly, when a node has less than a specified number of neighbors, it will reduce its range of motion.

After computing the size of MR, its exact place will be determined considering the condition of the node's neighborhood. To calculate the location of the rectangle according to the sensor's coordinates, we introduce x_l, and y_u, as the distances of the sensor with the left and up edges of MR, respectively. Also, x_r, and y_d have similar definitions for the right and down edges of MR. Moreover, we define n_l, and n_r as the number of neighbors that are located on the sensor left and right sides; see Fig. 1. At last, n_u, and n_d are defined similarly for up and down directions. Regarding that moving to crowded areas is not preferable, we conclude:

$$\frac{x_l}{x_r} = \frac{n_r}{n_l} \Rightarrow x_l = \frac{n_l}{\mid N \mid} \times MR.x \tag{1}$$

where $\mid N \mid= n_r + n_l$ is the number of sensor's neighbors, and $MR.x$ is the length of the MR. y_u can be computed similarly.

Having the values of x_l and y_u the exact place of MR is practically achieved.

B. Exchanging Goals of Sensors (EGS)

Early energy depletion is the main shortcoming of SDR, due to long distances that sensors should pass. Thus, EGS is proposed to alleviate this problem, similar to [9]. In EGS, sensors communicate with each other to find pairs whose destinations swapping results in reduced total movement. As the sensors are similar, these two nodes can exchange their destinations to shorten the total distance that should be passed. To avoid much energy consumption incurred by message passing, each sensor only communicates with its k-neighbors (nodes that can be accessed with path lengths smaller than k hops) to find another node to exchange destinations. Of course, the matter of deciding about k is a trade-off between communication and locomotion energy utilization.

C. One Step Deployment (OSD)

Although SDR convergence time is rather low, it is not still optimum. In fact, reaching full coverage in only one step can be of interest in all deployment methods. Here, we proposed OSD as a self-deployment method that can accomplish full coverage in only one step. However, an extra condition is required to make OSD applicable that is initial connectedness of the network. Since in many deployment scenarios the devices form a dense cluster initially, this assumption is not so restricting.

The algorithm runs on an $X \times Y$ field. To simplify the description, sensing radius of the devices is assumed to be $\sqrt{2}$. The sensors try to fill a grid map of integer points in the OSD algorithm. More formally, each point (i, j) should be occupied by at least one sensor, where $0 \leq i < X$, $0 \leq j < Y$, for integers i and j. This occupation assignment happens in three steps:

Step 1: A spanning tree of the network connectivity graph is constructed using a distributed implementation of Breadth First Search. Considering itself as the root of the tree, each node sends $(rID, dist = 0)$ message to its neighbors where rID is its unique identifier and dist stands for its assumed distance to the currently selected root. When the node v receives the message $(rID, dist)$ from u, it updates its assumed root and parent as follows. If rID is greater than the current root it will be selected as the new root. Then, v will reset its parent to the message sender, u, and will forward $(rID, dist+1)$ to its neighbors, otherwise an ignore message will be sent to u. As BFS prefers the shortest path, messages with smaller dist are selected to break the tie. Clearly, receiving ignore messages from all of the adjacent nodes a sensor identifies itself as a leaf of the tree. This phase of the algorithm terminates when nodes do not receive any message for a predetermined period of time.

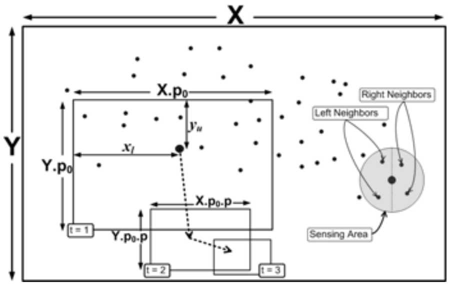


Fig. 1. Moving randomly with respect to Movement Rectangle.

Step 2: Starting from the leaves of the constructed tree, upon receiving the size of all its branches, each node sends its own sub-tree size, including itself, to its parent. By the completion of this recursive calls, all of the sensors know the number of nodes in each of their branches.

Step 3: Finally, the root selects the first point, $(0, 0)$, as its destinations and assigns the remaining points to its children according to their sub-tree size. Each sub-tree is responsible to cover a sub-area whose number of points is equal to the sub-tree size. The starting point (i, j) that should be covered by each sub-tree will be sent to its root. Knowing the size of its branches, the corresponding root does the same, in turn. Hence, after completion of this algorithm each node knows its final destination. Once EGS is applied to optimize the total movement of the nodes, the sensors can start their travels to the computed destinations.

Note that if the problem is to maximize the coverage area the same approach can be applied with few refinements. Briefly, after constructing the tree the centroid of the sensors can be easily calculated, similar to counting branch nodes. Then, the $\sqrt{N} \times \sqrt{N}$ area centered by the gravity center will be chosen to be covered where N is the number of nodes.

III. Experimental Results

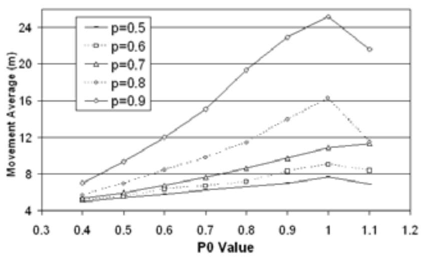
The experiments were done within a simulated environment of 100 sensors deployed in $10m \times 10m$ area, where sensing area of a sensor is a circle with radius $\sqrt{2}$ meter.

In the previous section, we explained the shrinking process for MR parameterized with p and p_0. However, sufficient explanation about the effect of these values on the system is not provided till now.

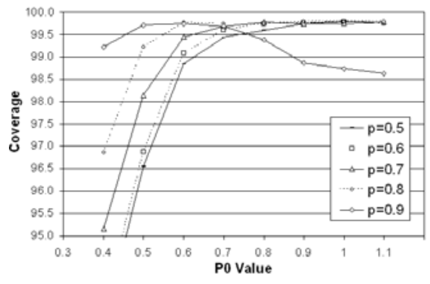
The maximum movement of a sensor after T epochs parallel to the x axis can be calculated from the following formula:

$$\frac{1}{2}\sum_{t=0}^{T} X.p_0.p^t = \frac{1}{2}X.p_0.\sum_{t=0}^{T} p^t = \frac{1}{2}X.p_0.\frac{1-p^{T+1}}{1-p} \tag{2}$$

Consider the special case that all of the sensors reside in the right edge of the field, initially. To gain full coverage, some sensor should pass all X parallel to the horizontal axis, to



(a) Average Movement vs. the p_0 value.



(b) Coverage vs. the p_0 value.

Fig. 2. The effect of different p and p_0 on the output.

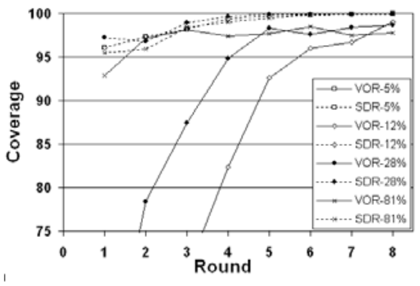


Fig. 3. SDR, and VOR, Coverage vs. Round.

reach the left edge. Thus, the following inequality is obtained:

$$\frac{1}{2}X.p_0.\frac{1-p^{T+1}}{1-p} \geq X \tag{3}$$

Form formula (3), and with the premise that we want to gain convergence to full coverage when $T \to \infty$, we gain:

$$p \geq 1 - \frac{p_0}{2} \tag{4}$$

This inequality extremely helped us to reduce the amount of testing values. Fig. 2 depicts the variance of the average movement and coverage metrics with respect to p, and p_0. Fig. 2 (a) demonstrates an increasing nature which means high values of p, and p_0 causes higher total movement. The justification behind this fact is that larger MR results in larger expected value of total movement. On the other hand, it can be deduced from (b) that the coverage for p_0 in the interval $[0.6, 0.8]$ are near optimal. Except the curve with $p = 0.9$ others present an increasing nature later on. Nevertheless, their average movement also increases; (a). Consequently, $0.5 \leq p \leq 0.7$, and $0.6 \leq p_0 \leq 0.8$ are selected as the best intervals.

[6] presents effective methods for movement-assisted self-deployment of the sensors. Its proposals could overcome the previous alternatives considerably. VOR is the best method suggested their with respect to average movement. In addition, its performance in coverage is very close to the Minimax that is declared to be the most effective scheme of the paper. Therefore, SDR method has been compared with VOR as a benchmark. Fig. 3 presents the convergence progression, in SDR, and VOR, for some sample networks with different initial coverage. SDR outperforms VOR, for low initial coverage extremely. This superiority roots in the randomized nature of the algorithm that is with a stochastic movement in the first step a near uniform distribution is gained. Moreover, for semi-covered initial conditions, SDR demonstrates its capability over the other method. [6] also proposed an optimization to overwhelm this problem by random movement in the first step, when sensors are not distributed evenly. Nevertheless, lacking some methods like EGS causes too much movement and energy consumption in the first step.

Totally, SDR improves the movement results with respect to VOR about 22%. Moreover, SDR reaches the acceptable coverage in about 4.3 rounds in average, while this value is 9.5 for VOR. The most effective approach of this paper, OSD, can reach full coverage in only one step with similar movement results as SDR. Nonetheless, sensors should form a connected graph initially to make this method applicable.

IV. Conclusion

In this letter, we proposed SDR for movement-assisted self-deployment of sensor networks. The stochastic nature of this algorithm helps it to be simpler, more effective and more practicable than current techniques. Some extensions also are introduced to reduce the amount of energy usage, and to increase the convergence rate. Our declarations are also supported with analytical and experimental results. In addition, the comparison between SDR, and the Voronoi based method as one of the best existing techniques, demonstrates superiority of SDR, particularly when the initial clustering rate is high. Furthermore, for those networks that are connected initially, OSD is proposed in which the deployment process converges in one step which is certainly the optimum value.

References

[1] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors,” *Commun. of the ACM*, May 2000.
[2] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, *Sensor Deployment Strategy for Target Detection*. WSNA, 2002.
[3] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization based on virtual forces,” in *Proc. INFOCOM 2003*.
[4] A. Howard, M. J. Mataric, and G. S. Sukhatme, “Mobile sensor networks deployment using potential fields: a distributed, scalable solution to the area coverage problem,” in *Proc. 6th International Symposium on Distributed Autonomous Robotics Systems 2002*.
[5] 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, Sept. 2002.
[6] G. Wang, G. Cao, and T. L. Porta, “Movement-assisted sensor deployment,” in *Proc. IEEE INFOCOM 2004*.
[7] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization based on virtual forces,” in *Proc. IEEE INFOCOM 2003*.
[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, 13 May 1983.
[9] G. Wang, G. Cao, and T. L. Porta, “Proxy-based sensor deployment for mobile sensor networks,” in *Proc. IEEE MASS 2004*.