Efficient Boundary Estimation for Practical Deployment of Mobile Sensors in Hybrid Sensor Networks

Nadeem Ahmed [1] [2]
nahmed@cse.unsw.edu.au

Salil S. Kanhere [1]
salilk@cse.unsw.edu.au

Sanjay Jha [1]
sjha@cse.unsw.edu.au

[1] Computer Science and Engineering,
University of New South Wales, Sydney,
Australia.

[2]National ICT Australia (NICTA)
Australian Technology Park, Sydney,
Australia.

***Abstract*— We address the deployment issues in a hybrid sensor network consisting of both static and mobile sensor nodes. Existing deployment schemes often assume either known regular boundaries of the region, or that mobile sensors are able to *detect* the region boundary. This is overly idealistic especially for unknown, outdoor environments. In our proposed two-phase deployment scheme, following their initial random deployment, the static sensors estimate the boundary of the unknown region by using the *right-hand rule*. This phase results in the identification of static boundary nodes, *B-nodes*. The mobile sensors are assumed concentrated at one or more points within the target area. In phase II, mobile sensor nodes spread in the target area in a distributed manner using one of the proposed variations of the *Virtual Force Algorithm*. Neighboring B-nodes form a *Virtual Boundary* and exerts repulsive forces on mobile nodes to keep them in the target area. Using simulations, we demonstrate the effectiveness of our proposed scheme in uniformly deploying mobile sensor nodes in a hybrid sensor network.**

I. INTRODUCTION

Wireless sensor networks (WSN) are networks of tiny autonomous devices that combine sensing, computing and wireless communication capabilities [1]. Potential applications include environmental monitoring, disaster recovery operations and target tracking etc.

In this paper, we seek to address the problem of deploying mobile sensors nodes in an unknown outdoor environment. Our system assumes a hybrid network consisting of a large number of static sensors, which we assume are deployed in a non-deterministic random manner, and a few mobile sensors nodes. We assume that sensors cannot detect the physical boundary of the region in outdoor environments that is consistent with the real world capabilities of existing sensor hardware. As we aim to uniformly spread the mobile sensors in the unknown area, it is imperative to provide a notion of boundary that define the extent of the area. For this purpose, we employ a *virtual boundary* formed by the static nodes lying on the perimeter of the deployed topology.

Our deployment scheme works in two distinct phases. Static nodes estimate the boundary of the unknown region in Phase I. Boundary estimation is done in a distributed manner resulting in the identification of B-nodes, nodes lying on the perimeter of the deployed topology. Neighboring B-nodes form the *virtual boundary* of the region and help in deployment of mobile sensor nodes in phase II. Phase II aims at uniformly spreading out the mobile sensors in the target area with minimal expenditure of energy. The mobile nodes use one of the variants of the Virtual Force Algorithm (VFA) for distributed decision making to spread in the region. The B-nodes keep the mobile nodes in the area bounded by the virtual boundary by exerting repulsive virtual forces.

Note that we are not utilizing mobility at deployment time to *extend* the coverage by going beyond the region currently covered by static nodes or *increase* the coverage by filling any *coverage holes*, areas not covered by any sensor, that may exist in the topology. We focus on uniformly distributing the available mobile sensors in the topology. This uniform distribution can be utilized for clustering, wherein the mobile sensors can act as cluster head, overlay network consisting of mobile nodes with long-range and high bandwidth communication capabilities, hierarchical routing, and, fault repair using mobile sensors [2], [3] etc. Our proposed deployment scheme can also help in coverage *maintenance* as uniform distribution minimizes the reaction time in relocating a mobile sensor for replacing a low energy/faulty/dead sensor during the operation of the WSN.

We have carried out extensive simulations, which demonstrate that our scheme can uniformly deploy the mobile sensor nodes with minimal expenditure of energy. The remainder of this paper is organized as follows. We discuss related research work in Section II. Sections III, and IV cover various phases of our deployment scheme. Simulation setup and results are discussed in Section V. Section VI concludes the paper.

II. RELATED WORK

Deployment of mobility capable sensors has been proposed in various research efforts [4], [5], [6], [7] and, [8] etc. [4] proposed three different deployment protocols that spread out the mobile sensors once coverage holes are detected using Voronoi diagrams. [5] and [8] are potential field based approaches both assuming compact initial concentration of the mobile nodes. [6] is a centralized, incremental deployment scheme deploying sensors one by one and requiring line of sight among the nodes. Similarly a centralized virtual force algorithm for the movement of mobile sensor nodes is proposed in [9]. In [7] the

1-4244-0507-6/06/$20.00 ©2006 IEEE

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 00:42 from IEEE Xplore. Restrictions apply.

coverage problem is solved by a moving robot that is guided by the already deployed nodes for exploring poorly covered areas.

Boundary estimation in WSN is usually referred to as estimation of delineation between homogeneous sensing regions using sensor data [10]. Boundary estimation in our work refer to the detection of nodes lying on the boundary of the target region. Algorithm for boundary estimation are presented in [11], [12], and [13]. [11] proposed the BoundHole Algorithm using the right-hand rule to identify nodes on the boundary of geometric holes. [13] proposes a boundary estimation algorithm without assuming that location information is available.

Our work is different to these proposed approaches in several ways. Almost all of the existing mobile sensor deployment schemes either assume regular, known deployment regions without taking into account practical boundary estimation/detection or assume indoor deployment where mobile nodes are able to detect the walls as obstructions. We assume an outdoor, obstacle free environment and employ a realistic boundary estimation mechanism using the static sensors nodes. Our proposed algorithm is inspired from the Jarvis walk proposed in [14]. The Jarvis walk is a centralized approach to find the convex hull of a set of points in plane. We propose a simple and distributed boundary estimation mechanism that does not require flooding the deployed network for gathering the topology information. In addition, we propose a localized and distributed variant of the VFA for spreading out the mobile sensors as opposed to the centralized VFA employed in [9]. We also propose the simulated movement approach for energy efficient movement of mobile sensor nodes. This simulated approach is different from the proxy based logical movement approach proposed in [15] where static sensor nodes are utilized as proxies for message passing.

III. PHASE I-BOUNDARY ESTIMATION BY STATIC NODES

This phase starts with the deployment of static nodes in the target area. We assume a random non-deterministic deployment of static nodes and that location information is available using any existing sensor network localization scheme. This phase aims to identify the B-nodes, i.e. nodes lying on the boundary/perimeter of the topology formed by the deployed static nodes. A naive approach is to collect the deployed topology information at the sink/base station by flooding probing packets in the whole network. The topology information once gathered can help in identification of boundary nodes by geometric computation for convex hull. This approach is centralized and involves high communication and computation cost and does not scale well with the size of the network. A localized and distributed boundary node selection algorithm is thus desired.

We propose a B-node selection algorithm, based on the simple geometric right-hand rule, for identification of boundary nodes. This algorithm is executed at B-nodes only and it consists of two parts; application of right-hand rule to compute the next B-node and sending of a B-message to the next selected B-node. After neighbor information has been exchanged, a node that receives a B-message marks itself as a B-node. It then sweeps clock-wise from the direction of the sending node and selects the first node that intersects the sweeping line from its neighbor list (by simple geometric calculations as location information of neighbors is available). Note that our algorithm is different than the Jarvis walk in that we compute the angles for neighbors only (nodes in communication range) as compared to all points in the plane. The node then unicasts a B-message to the next selected node and stores the information about previous B-node (node from which the B-message was received) and the next B-node (node to which B-message is sent). The node that initially originated the B-message is referred as OB-node (originating B-node). The B-message continues through the perimeter of the deployed topology till the B-message reaches back the OB-node.

The B-node selection algorithm is simple, distributed and only involves nodes lying on the perimeter of the topology. Some of the related issues are discussed below.

A. Selection of the Originating B-node

The data from a deployed WSN is often collected at a base station or sink for analysis. We assume that a sink node is available (outside the WSN convex region) and that it can communicate with a set of deployed nodes. After learning the location information of the neighboring static nodes that are within its communication range, the sink selects the neighbor static node with the least relative x-coordinate as the OB-node. The sink unicasts a B-message to the selected node to trigger the B-nodes selection algorithm. The OB-node applies the right-hand rule by sweeping clock-wise from the direction of the sink(see Figure 1 (a)) to select the next B-node. Note that the sink is only used for starting the B-node selection process and it does not participate any further in the selection process. Also the OB-node does not mark the sink as its previous B-node, the correct previous B-node will connect with the OB-node when the B-node selection terminates.

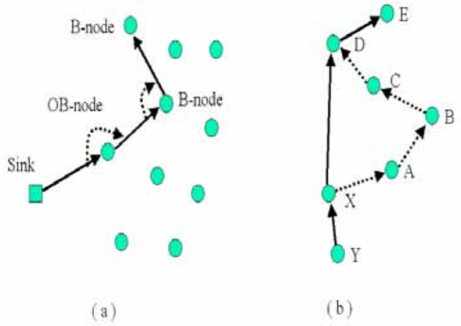

Fig. 1. Right Hand Rule, (a) Selection of OB-node (b) Selection of B-nodes

B. B-nodes are a subset of Perimeter Nodes

The set of nodes that get selected as B-nodes depends on the node communication range and are a subset of all boundary/perimeter nodes. As an example see Figure 1 (b). Node X has nodes $Y, A, B, C,$ and D in its neighbor list. Node X applies the right-hand rule and selects node D from its neighbor list. The selected perimeter is $YXDE$ and is possible because node D is within the communication range of node X. Note that nodes $A, B,$ and C have not been selected, although the perimeter can also be traced as $YXABCDE$ if we use a shorter communication range.

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 00:42 from IEEE Xplore. Restrictions apply.

C. Local Minima

Consider Figure 2. Node X receives the B-message from node Y. X applies the right-hand rule and selects node A as the next B-node. As node A has only one neighbor, node X, it cannot select any new B-node except node X. We call this special case as the local minima for the algorithm. Recall that each node maintains the previous and next B-node information. Node A sends the B-message, with a local minima flag, back to previous B-node X without marking itself as a B-node. Node X removes node A as the next B-node, continues with the right hand rule and selects node Z as next B-node now. In case a node receiving the B-message with local minima flag has no other neighbor to select than its previous B-node, it again sends a local minima B-message to the previous B-node (see Figure 2 (b)). The algorithm can thus recover from a local minima.

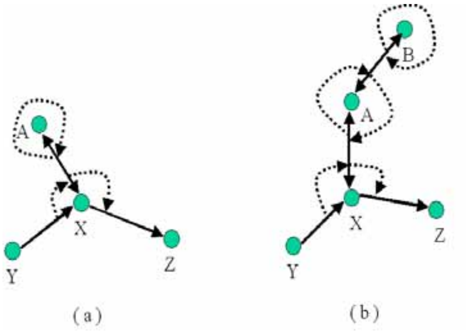

Fig. 2. Local Minima

D. Algorithm Termination

The algorithm terminates when the B-message (without the local minima flag) is received at the OB-node. Recall that an OB-node is the one with least x-coordinate. This ensures the protocol termination (see [14] for details).

IV. PHASE II-SPREADING OUT OF MOBILE SENSORS

Phase II aims at uniformly distributing the mobile sensor nodes in the target area. For mobile sensors, we consider two different initial deployment methodologies namely *Island* and *Normal* distribution. In normal distribution, mobile sensors form a single cluster at the boundary while in island distribution they form different disconnected clusters at different locations on the boundary. For spreading out of the mobile sensor nodes, we propose to use the concept of Virtual Forces from the robotics [5]. In this context we propose two movement strategies namely Virtual Force Algorithm (VFA) and VFA with Simulated Movements (VFA-SM). We make the following assumptions.

- Location information is available.
- The target area is an unknown obstacle-free environment.
- Mobile nodes have more initial energy than the static nodes. For example, Robomote [16] has 4528J (3.7V Lithium battery) while initial energy for Mica2 motes is about 3000J (2 AA batteries). The difference in the initial energies permits the relocation of mobile nodes such that after movement, a mobile node has remaining energy comparable with that of static nodes.

This phase starts with mobile nodes and B-nodes sharing the location information with their neighbor mobile nodes using HELLO messages. HELLO messages contain the sender node ID, current location coordinates, and a status flag. A mobile node on reception of a HELLO message marks the node as current neighbor and stores the information contained in the HELLO message. The mobile node now uses one of the following movement calculation algorithm;

A. Virtual Force Algorithm (VFA)

We design a variant of VFA proposed in [5], [9] for deploying the mobile sensors. VFA attempts to iteratively spread the given mobile sensors in the target area by using a combination of attractive and repulsive forces. Two mobile sensors will exert virtual forces on each other if the Euclidean distance, $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, between them is not between a given range of thresholds, Th_{push} and Th_{pull} (discussed in detail in a later Section). This virtual force, F_{ij} is a pull or attractive force if the distance between the two mobile nodes is greater than the pull threshold, Th_{pull}, while if the distance is less than the push threshold, Th_{push}, a push or repulsive force is exerted. Equation 1 shows the model used for decision making.

$$\vec{F_{ij}} = \begin{cases} F_{push}, & \text{if } d_{ij} < Th_{push} \\ 0, & \text{if } Th_{push} \le d_{ij} \le Th_{pull} \\ F_{pull}, & \text{if } d_{ij} > Th_{pull} \end{cases} \qquad (1)$$

where $\vec{F_{ij}}$ is the force exerted on mobile node Si by neighbor Sj. Taking the midpoint of range between Th_{push} and Th_{pull} as the desired distance between the mobile nodes, Equations 2 and 3 gives the push/pull virtual forces.

$$F_{push} = \frac{\left(\frac{Th_{pull} + Th_{push}}{2}\right) - d_{ij}}{2} \qquad (2)$$

$$F_{pull} = \frac{d_{ij} - \left(\frac{Th_{pull} + Th_{push}}{2}\right)}{2} \qquad (3)$$

Note that half of the exerted force is absorbed by each neighbor i.e. each mobile node will only move half of the total distance due to the full virtual force. We can express the total force applied on a mobile sensor S_i by its k mobile neighbors, denoted by $\vec{F_i}$, as,

$$\vec{F_i} = \sum_{j=1, j \neq i}^{k} \vec{F_{ij}} \qquad (4)$$

where $|F_{ij}|$ represent the magnitude of the force absorbed by node i. Note that $\vec{F_i}$ is the vector sum of all the forces acting on mobile sensor node S_i, the magnitude and orientation of which can be easily calculated, e.g. Robomote [17] has an on board compass for this purpose, which combined with localization information is used for navigation.

We introduce and discuss here certain constraints and improvements in the proposed VFA while the pseudo code of the modified VFA appears as Algorithm 1.

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 00:42 from IEEE Xplore. Restrictions apply.

1) Choice of Thresholds, Th_{push} and Th_{pull}: The movement triggering thresholds, Th_{pull} and Th_{push}, depends on the link quality. These two thresholds when characterized on the basis of radio link ensure continued connectivity during movements and result in uniform distribution of the nodes. Using a simple radio propagation model based on log-normal shadowing, the path loss PL(in dB) at a distance d is given by Equation 5.

$$\overline{PL(d)} = \overline{PL(d0)} + 10 \cdot n \cdot log(\frac{d}{d0}) + X_\sigma \tag{5}$$

where

$d0$ = Reference distance

n = Path loss component, indicating the rate at which the path loss increases with distance

X_σ = Zero-mean Gaussian distributed random variable (in dB) with σ-variance (shadowing, also in dB)

$\overline{PL(d0)}$ = Mean path loss at reference distance d0.

X_σ in Equation 5 captures various environmental factors resulting in different received signal values at different locations although the distance between the two sensors is the same. $PL(d0)$ can be measured experimentally for a given event and sensor characteristics or can be calculated using free space path loss model [18].

Each sensor has a *receive threshold* value γ that describes the minimum signal strength that can be correctly decoded at the sensor. The probability Pr that the received signal level, P_{rec} at a sensor will be above this receive threshold, γ, is given by Equation 8, with Q-function to compute probability involving the Gaussian process. The Q-function is defined as

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^\infty exp(-\frac{x^2}{2})dx \tag{6}$$

where

$$Q(z) = 1 - Q(-z) \tag{7}$$

$$Pr[P_{rec}(d) > \gamma] = Q[\frac{\gamma - P_{rec}(d)}{\sigma}] \tag{8}$$

For a given transmit power and receive threshold value, we can calculate the probability of receiving a signal above the receive threshold value, γ, at a given distance using Equations 6 and 8 as shown in Figure 3.

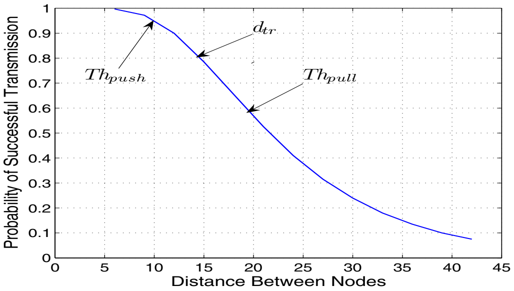

Fig. 3. Change in communication probability with distance

Following the terminology in [19], there are three distinct reception regions in a wireless link: connected, transitional, and disconnected. The transitional region has highly unreliable links and its region bounds can be found either by analytical or empirical methods [19]. Let d_{tr} represent the point where the transitional region begins. Regions between Th_{pull} and d_{tr}, and d_{tr} and Th_{push} reflects the tolerance to the errors in localization and odometry during navigation of the mobile nodes. As long as the final position after movement is within this range, the deviation from the ideal trajectory during movement can be tolerated by our movement algorithm. d_{tr}, Th_{push}, and Th_{pull} are represented in terms of probability of correct packet reception in Figure 3.

2) Boundary considerations: Notion of boundary is modelled by B-nodes exerting virtual forces ($\vec{F_{ib}}$) on mobile sensor nodes. This is to ensure that mobile sensors do not "fall off" the boundary of the target region while moving in the unknown environment.

A virtual force $\vec{F_{ib}}$, based on the mobile node distance from a B-node, is included in resultant vector sum of all virtual forces. The new total force is then expressed as,

$$\vec{F_i} = \sum_{j=1, j \neq i}^{k} \vec{F_{ij}} + \sum_{b=1}^{n} \vec{F_{ib}} \tag{9}$$

where $\vec{F_{ib}}$ is the repulsive force exerted on the mobile node Si by n neighbor B-nodes and is modelled by Equation 10.

$$\vec{F_{ib}} = \begin{cases} \vec{F_b}, & \text{if } d_{ib} < \frac{(Th_{push} + Th_{pull})}{2} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

The repulsive force $\vec{F_b}$ is given by Equation 11 where d_{ib} is the distance between the node and the B-node.

$$\vec{F_b} = \frac{(Th_{push} + Th_{pull})}{2} - d_{ib} \tag{11}$$

As B-nodes are all static nodes, mobiles nodes absorb all of the virtual force resulting from the B-nodes. As a final check, mobile nodes should not cross the virtual boundary formed by the known B-nodes.

3) Energy considerations: To safeguard against a node exhausting a major portion of its energy in the spreading out phase, we define two roles for the mobile sensors namely *Active* and *Passive*. Initially a mobile node participates in the VFA as an active node until its current remaining energy falls below $K_1 \times E_i$, where K_1 is a tunable parameter that controls the amount of energy consumption permitted for this deployment phase and E_i is the initial energy. After becoming passive, HELLO messages broadcast by the node have a special passive flag set indicating that the node is in passive mode and cannot move from its current location. Neighbors of this passive node calculate and absorb all of the virtual force resulting from the passive node.

$$F_{push} = (\frac{Th_{pull} + Th_{push}}{2}) - d_{ij} \tag{12}$$

$$F_{pull} = d_{ij} - (\frac{Th_{pull} + Th_{push}}{2}) \tag{13}$$

Equations 12 and 13 are used for passive mobile neighbors while Equations 2 and 3 are used for all active neighbors.

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 00:42 from IEEE Xplore. Restrictions apply.

Algorithm 1 Virtual Force Algorithm (VFA)

```
**Notations** :
$M_j$ = Set of mobile neighbors, $B_k$ = set of B-nodes
$T_{round}$ = Maximum time allowed for each round
**INIT** :
broadcast HELLO message
if HELLO message received then
    store NodeId, Location and flag in $M_j$ or $B_k$
**Process** :
set Round = 0
while Round $\leq$ MaxRounds do
    for each neighbor $j$ in $M_j$ do
        Calculate $\vec{F}_{ij}$ using $d_{ij}, Th_{push}, Th_{pull}$, and passive flag
    for each B-node $k$ in $B_k$ do
        Calculate $\vec{F}_{ib}$ using $d_{ij}, Th_{push}$, and $Th_{pull}$
    $\vec{F}_i = \sum \vec{F}_{ij} + \sum \vec{F}_{ik}, j$ in $M_j, j \neq i, k$ in $B_k$
    Calculate $(x, y)_{next}$ based on $\vec{F}_i$
    start moving toward $(x, y)_{next}$ and start timer $T_{round}$
    if $T_{round}$ expires then
        set Round = Round + 1
        if $E_{current} \leq K_1 \times E_i$ then
            set passive flag = true
        broadcast HELLO message
```

4) Oscillation control: Node oscillation can occur in successive rounds of VFA because of distributed decision making. A node may discover new neighbors, after the movement, that may force it back towards its initial position. As an oscillation control measure, a node should remember its last position from the previous round of movement. If the current resultant virtual force drives it towards the same direction, it only move half of the calculated distance i.e. resultant virtual force become $\vec{F}_i/2$. This will result in mobile node rapidly converging to its final position without excessive movements due to oscillations.

B. Virtual Force Algorithm with Simulated Movement

Mobile nodes move after each round of VFA due to the virtual force exerted by its neighbors before settling down to their final position in the topology. If we could calculate the final position of a mobile node and move directly to that final position, we can save energy by moving much lesser distances, i.e. through the cut-through paths instead of the zig-zag paths that result from the round-by-round operation of VFA.

VFA with Simulated Movement (VFA-SM) attempts to use the cut-through paths for movement. Nodes go through the VFA iterations and calculate new position after each round. The difference is that node do not physically move after each iteration rather they assume the new calculated virtual position. The HELLO messages at the start of the next round contain the new virtual positions enabling the recipients to use this updated position information for the upcoming round of VFA. Nodes only move once they have calculated their final positions. This approach has a few disadvantages. Recall that new neighbors in the unknown environment are discovered by the per round movements in VFA. Consider a scenario where initial mobile node deployment results in islands of communications. If a fully simulated run of VFA is used, the mobile nodes will never find out the presence of other mobile nodes in the region. Similarly, during simulated movements it is not possible to discover new B-nodes that model the boundary forces for an unknown environment.

One way to offset this disadvantage is to use intermittent simulated movement instead of fully simulated movement. In VFA Intermittent Simulated Movement (VFA-ISM) approach, nodes simulate the movement for x number of rounds. New neighbors including the B-nodes are discovered with actual movement after every x number of rounds e.g. in VFA-ISM2 nodes physically move after every second round.

V. SIMULATION RESULTS

TABLE I
SIMULATION PARAMETERS

Parameter	*Value*
Transmit power P_t	24.5 dBm
Receiving threshold (γ)	-65.0 dBm
Path loss exponent n (free space)	2
σ	4 dBm
Th_{push}/Th_{pull}	25m/33m
Initial Energy	4528J (3.7v, 345mAh)
Energy consumed in movement	8.274 J/m (Robomote [16])
Region	100m x 100m

We implemented the B-node selection algorithm, the VFA and the simulated variants in NS2 with parameters listed in Table I. Maximum number of rounds was set as 12. Each simulations was run for VFA and different variants of ISM (ISM2, ISM3, ISM4, and ISM6). Figures 4 and 5 show the total distance covered by all mobile sensors with normal and island initial deployment. The results are average over three different topologies for each type of deployment. Topology distribution are also shown in Figures 4 and 5. Small filled circles represent static nodes, B-nodes are shown by connected virtual boundary. Unfilled big circles are the initial deployment position of the mobile nodes while filled big circles are their final position.

ISM6 consistently moves the lowest distance and consequently consumes the least energy than other movement algorithms for all types of deployment. Comparing the same movement algorithm across different initial deployments, island deployment results in the least movement. Observing the topology distribution achieved for different types of deployment in Figure 4 and 5 (only 100-20 static-mobile nodes scenario shown), VFA results in better uniform distribution than all other movement algorithms while consuming the most energy for both normal and island initial deployment.

The simulation results show that the virtual boundary formed by the identified B-nodes successfully guides the mobile sensor nodes and keeps them in the useful area. Results also show that the intermittent simulated movements save considerable amount of energy by moving lesser distances than the VFA for different types of initial deployment. This saving is at the cost of slight nonuniformity in the nodes distribution. Performance of ISM3 is comparable to ISM4 and ISM6 in terms of energy consumption and yet it achieves topology distribution closer to the VFA. To summarize, VFA achieves

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 00:42 from IEEE Xplore. Restrictions apply.

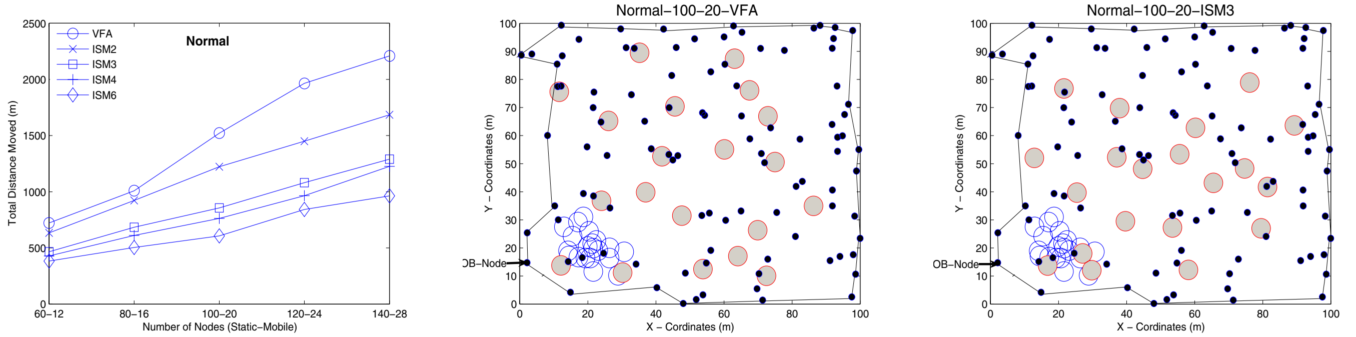

Fig. 4. Normal Initial Deployment, Distance moved, Topology VFA and ISM3

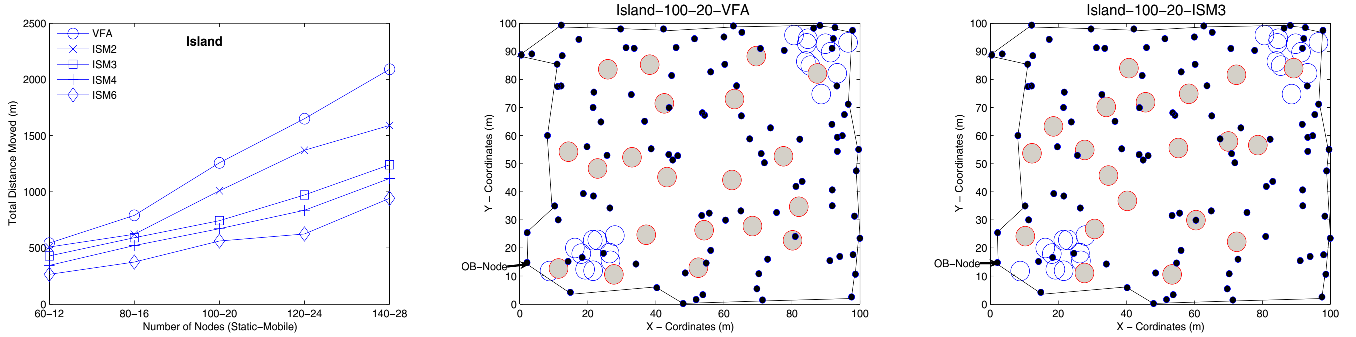

Fig. 5. Island Initial Deployment, Distance moved, Topology with VFA and ISM3

better topology distribution but if we want to restrict the amount of energy consumed in the deployment phase, ISM3 is a good compromise.

VI. CONCLUSION

In this paper, we have proposed a practical way for deploying mobile sensors in an unknown environment. Boundary of the unknown region is estimated by recognizing B-nodes, nodes lying on the boundary, by applying an algorithm based on the right-hand rule. The algorithm is localized, distributed and is executed only on a subset of all deployed static nodes. Mobile sensors spread out in the region by making distributed decisions based on local information of mobile neighbors and the B-nodes. Our simulation results demonstrate that mobile sensors are successfully deployed in the unknown region with the help of static sensor nodes while ensuring that low energy is expended by the mobile nodes.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, pp. 102–114, 2002.

[2] G. Wang, G. Cao, and T. L. Porta, “A bidding protocol for deploying mobile sensors,” in *11th IEEE International Conference on Network Protocol ICNP ’03*, pp. 315–324, Nov 2003.

[3] S. Ganeriwal, A. Kansal, and M. B. Srivastava, “Self aware actuation for fault repair in sensor networks,” in *Proceedings of the IEEE International conference on Robotics and Automation (ICRA ’04)*, May 2004.

[4] G. Wang, G. Cao, and T. L. Porta, “Movement-assisted sensor deployment,” in *IEEE INFOCOM 2004*, June 2004.

[5] 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 *6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, June 2002.

[6] 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*, vol. 13(2), pp. 113–126, Sep 2002.

[7] M. A. Batalin and G. S. Sukhtame, “Coverage, exploration and deployment by a mobile robot and communication network,” *Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks*, vol. 26(2), pp. 181–196, 2004.

[8] S. Poduri and G. S. Sukhatme, “Constrained coverage for mobile sensor networks,” in *IEEE International Conference on Robotics and Automation*, pp. 165–172, May 2004.

[9] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization in distributed sensor networks,” *ACM Transactions on Embedded Computing Systems*, vol. 2(3), pp. 1–29, 2003.

[10] R. Nowak and U. Mitra, “Boundary estimation in sensor networks: Theory and methods,” in *Proceedings of the 2nd IEEE IPSN’03*, April 2003.

[11] Q. Fang, J. Gao, and L. J. Guibas, “Locating and bypassing routing holes in sensor networks,” in *IEEE INFOCOM 2004*, June 2004.

[12] A. Kroller, S. P. Fekete, D. Pfisterer, and S. Fischer, “Deterministic boundary recoginition and topology extraction for large sensor networks,” in *Proceedings of the 17th ACM SODA’06*, 2006.

[13] S. Fekete, A. Kroller, D. Pfisterer, S. Fischer, and C. Buschmann, “Neighborhood based topology recognition in sensor networks,” in *Proceedings of the Algosensors’04, LNCS 3121*, pp. 123–136, 2004.

[14] R. A. Jarvis, “On the identification of the convex hull of a finite set of points in the plane,” *Inform. Process. Lett.*, vol. 2(No 1), pp. 18–21, March 1973.

[15] G. Wang, G. Cao, and T. L. Porta, “Proxy based sensor deployment in mobile sensor networks,” in *Ist IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS-04)*, pp. 493–502, Oct 2004.

[16] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, “Robomote: A tiny mobile robot platform for large scale ad-hoc sensor networks,” in *IEEE Conference on Robotics and Automation*, pp. 1143–1148, May 2002.

[17] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. Sukhatme, “Robomote: Enabling mobility in sensor networks,” Tech. Rep. CRES-04-006, University of Southern California, 2004.

[18] T. S. Rappaport, *Wireless communications : Principles and Practice*. New Jersey: Prentice Hall, 1996.

[19] M. Zuniga and B. Krishnamachari, “Analyzing the transitional region in low power wireless links,” in *1st IEEE Conference on Sensor and Ad Hoc Communications and Networks SECON ’04*, Oct 2004.

Authorized licensed use limited to: IEEE Xplore. Downloaded on October 30, 2008 at 00:42 from IEEE Xplore. Restrictions apply.