

# An Integrated Self-Deployment and Coverage Maintenance Scheme for Mobile Sensor Networks

Xu Li and Nicola Santoro

School of Computer Science, Carleton University  
xlii@connect.carleton.ca  
santoro@scs.carleton.ca

**Abstract.** In mobile sensor networks, the *coverage improvement* problem, i.e., maximizing and/or maintaining overall sensing coverage, is a fundamental research issue attracting many researchers. Existing coverage improvement algorithms such as sensor self-deployment algorithms and sensor relocation protocols enhance coverage with limitations due to their specialized design purposes. In this paper, we propose an integrated self-deployment and coverage maintenance scheme, which solves the coverage improvement problem in a complete sense. The proposed scheme is an integration of four algorithms: a redundancy determination algorithm, a sensor relocation protocol, a sensor self-deployment algorithm, and a sensor replenishment protocol. By this scheme, redundant sensors are placed together with non-redundant ones in the target field at random; non-redundant sensors autonomously scatter to form a network with maximal coverage after initial placement; all the sensors collaborate to compensate coverage loss throughout network lifetime. Mentionably, we notice that no existing schemes besides ours take into account the impact on coverage from nodal sensing range diminishment. At the end of the paper, we briefly address some evaluation issues.

## 1 Introduction

Mobile sensor networks (MSNs), as a new paradigm of wireless sensor networks (WSNs), emerged approximately five or six years ago. They inherit all the properties such as the resource constraint and the infrastructureless nature from WSNs, and meanwhile, they are featured with their own particularity, i.e., node mobility. This feature allows sensors to act in a much intelligent way and make MSNs more flexible and adaptive to unknown/hazardous environment compared to their static counterpart. An increasing number of research activities are currently being carried out for MSNs. One of the fundamental and attractive issues is *coverage improvement*. In a sensor field, a point is said to be covered iff it falls into at least one sensor's sensing range. The overall sensing coverage of a sensor network is just the aggregation of the areas covered by all the network nodes. A MSN with maximal coverage can timely capture the interesting events happening in the sensor field; a MSN with constant coverage is able to offer

sensing service without quality degradation. Hence, the coverage improvement problem aims to find optimal solutions to maximizing and/or maintaining the overall sensing coverage of a network.

There are two main streams of algorithms, i.e., *sensor self-deployment*[1–7] and *sensor relocation*[8–10], for coverage improvement in MSNs. Other streams include, for example, robot-assisted approaches[11, 12]. Since uniform sensor distribution may yield optimal coverage, sensor self-deployment focuses on the way of converting a randomized sensor distribution to a uniform one without human assistance. As for sensor relocation, it concentrates on how to strategically move sensors to maintain existing coverage in the presence of node failure. Due to their specialized design purposes, the two types of approaches supplement each other and may combine to solve the coverage improvement problem on a complete basis. However, to our knowledge, no such an integrative solution has been presented in literature. In this paper, we propose an integrated sensor self-deployment and coverage maintenance scheme to fill the blank.

The proposed scheme is designed to empower MSNs to maximize their overall sensing coverage and operate without coverage degradation in the scenarios (e.g., Mars exploration) where human assistance is infeasible or too costly. It involves the utilization of redundant sensors and requires the original network size and the expected network operating period to be known a priori. The proposed scheme is composed of four algorithms: a redundancy determination algorithm (RDA), a virtual-force-based self-deployment algorithm (VFSD), a zone-based sensor relocation protocol (ZONER)[10], and a sensor replenishment protocol (SRP). The execution of the scheme spans the entire networking process from pre-deployment to post-deployment. First of all, the RDA is run to determine the number of redundant nodes (or, R-nodes for short) to be dropped together with the initial set of network nodes, i.e., non-redundant nodes (or, NR-nodes for short). After node dropping, NR-nodes autonomously spread out by executing the VFSD to form a network covering the target field as much as possible. During the operating period of the network, some R-nodes are activated by the ZONER to replace failed NR-nodes; the other R-nodes are gradually injected into the network by the SRP to compensate the coverage loss due to sensing range diminishment. On a periodical basis, the network is geographically reorganized through the VFSD to eliminate the gaps and overlapping between nodal sensing ranges. The novelty of the proposed scheme resides in the following four aspects:

1. the introduction to the affect on coverage from sensing range diminishment;
2. the development of the RDA that determines node redundancy in advance;
3. the design of the VFSD that is adaptive to nodal sensing radius difference;
4. the design of the SRP capable of activating a specified number of R-nodes.

The remainder of this paper is organized as follows: Section 2 reviews some existing work on sensor self-deployment and sensor relocation; Section 3 introduces the two main reasons for coverage loss; Section 4 presents the proposed scheme in detail; Section 5 addresses some evaluation issues; Section 6 summarizes the paper and point out our future work.

## 2 Related Work

In this section, we will briefly review some existing sensor self-deployment algorithms and sensor relocation protocols.

### 2.1 Sensor Self-deployment

Howard, Mataric, and Sukhatme[1] proposed an incremental deployment algorithm for mobile sensor networks. Under the assumptions of global coordination, location awareness, and nodal visibility, this algorithm deploys nodes one-at-a-time and maintains a line of sight relationship between nodes. For a node to be deployed, it first constructs a occupancy grid using the information gathered from previously deployed nodes. The cell in a occupancy grid is either free or occupied or unknow. Then, the node converts the occupancy grid to a configuration grid, in which a cell is free iff all nearby cells are free, or occupied if at least one nearby cell is occupied, or unknow otherwise. Afterward, the configuration grid is transformed to a reachability grid, where cells are marked either reachable or unreachable. According to different cell selection policies, the node choose and move to different reachable cell, and thereby yielding different network topology.

Howard, Mataric and Sukhatme[2] introduced a potential field based approach to sensor self-deployment problem. In their approach, nodes are assumed to be able to measure the range and bearing of its nearby nodes and seeable obstacles. A node receives virtual repulsive force from a potential field generated by other nodes and seeable obstacles. This virtual force depends only on the relative position of the node itself and the force sources. A node is driven by its received virtual force to move. When the virtual force exerted on a node is zero, the node stops moving. The network reaches a static equilibrium status when all the nodes stay still. The nice property of this algorithm is that neither global knowledge nor message transmission is required. Similar algorithms include the VEC[4], the one proposed in [3] and the DSSA/IDCA[5].

Heo and Varshney[5] proposed a deployment algorithm VDDA based on Voronoi diagram under the assumptions of global coordination, location-awareness, and pre-knowledge to network size and target field area. In their approach, the effective area of a node is defined as the intersection of the node's sensing range and its Voronoi polygon, and coverage is improved by increasing each node's effective area with minimal energy consumption. To do so, a node by the algorithm examines the points from its current location to the centroid of the Voronoi polygon and then to the center of the Voronoi range to find the point that can maximize the node utility metric as its target location. A node's utility metric is defined as the product of the node's effective area and estimated lifetime. Similar algorithms include the VOR presented in [4].

Wu and Yang[6] proposed a scan-based sensor deployment scheme (SMART) by treating sensor deployment as a load balancing problem. This algorithm works under the assumptions of sufficient node density, pre-knowledge to the target field, global coordination and location-awareness. By the algorithm, the target field is partitioned into a 2-D mesh. The nodes in a cell of the 2-D mesh together

form a cluster covering that cell. Each cell is covered by one and only one cluster. If a cell does not contain any nodes, it is considered covered by an empty cluster. The nodes in each cell is viewed as load. In a pre-processing step, empty clusters are filled with redundant nodes of non-empty clusters using the approach of recursive doubling expansion. Then, two rounds of balancing are performed by cluster heads to balance the number of nodes in each cell.

## 2.2 Sensor Relocation

Wang, Cao and Porta[8] presented a proxy-based sensor relocation protocol for the sensor networks containing both statics and mobiles under the assumptions of global coordinates and location-awareness. The protocol estimates the size of the coverage hole generated by the mobile node in the case that the node leaves its current location, and assigns the node a base price accordingly. Each static node independently identifies coverage holes using Voronoi diagram and bids the closest mobile node with smallest base price. A mobile node is bid by the largest hole when receiving multiple bids. By this means, a mobile node always intend to move to large holes from small ones until no larger holes can be detected. To save energy, mobiles perform logical move for transient locations, and they conduct actual movement is conducted only when final location is found.

Wang, Cao, Porta and Zhang[9] proposed a grid-quorum based sensor relocation protocol under the assumptions of global coordinates, location-awareness, and pre-knowledge to network field. In this protocol, the network field is geographically partitioned into grids, in each of which, a node is elected as grid head and responsible for gathering the location information about all the grid members. Based on grid members' location, a grid head determines redundant nodes and detect sensing holds. Each grid head publishes redundant node information inside its grid row (demand quorum). When a grid head finds a sensing hole, it broadcasts a request in its grid column (demand quorum). Because every demand quorum intersects with all the supply quorums, redundant nodes are then discovered. The closest redundant node is then relocated in a cascaded way along a carefully selected path to fill the sensing hole.

Li and Santoro[10] proposed a zone-based sensor relocation protocol (ZONER) under the assumptions of global coordinates and location awareness. This protocol shares similar idea with the grid-quorum based protocol[9] in node registration and node discovery, but it outperforms the grid-quorum based protocol in that it requires zero knowledge about the network field and has the immunity to the void-areas caused by obstacles or unbalanced node distribution. Protocol ZONER will be explained in more detail later, in Section 4.3.

## 3 Inevitable Coverage Loss

From Quality of Service (QoS) point of view, it will be advantageous if the overall sensing coverage of a mobile sensor network (MSN) can keep undecreased during network lifetime. Unfortunately, it property can not be practically achieved

without making extra efforts, since coverage loss is an inevitable phenomenon in real world scenario. There are two coverage loss factors, i.e., *node failure* and *sensing range diminishment*. In this section, we will discuss them in detail.

### 3.1 Node failure

A node is said to be a failed node if it is no longer able to deliver sensing service. Failed nodes may possibly generate sensing holes in a network since the coverage provided by these nodes is completely lost. The reasons why a node fails could be multifold: hardware defects, harsh environmental condition, and so on. Let a random variable  $T_f$  represent the time instant at which a node fails. Let  $f(\tau)$  be the probability density function of  $T_f$ , and  $f(\tau) = 0$  for  $\tau < 0$ . Clearly,  $\int_0^\infty f(\tau)d\tau = 1$ . Define *node unreliability*  $Q(t)$  as the conditional probability that a node does not correctly operate throughout interval  $[0, t]$  given that the node is operational at time 0. *Node reliability*  $R(t)$  is then defined as the conditional probability that a node functions properly throughout interval  $[0, t]$  given that it is operational at time 0.  $Q(t)$  and  $R(t)$  can be represented using  $f(\tau)$  as follows:  $Q(t) = Prob\{T_f < t\} = \int_0^t f(\tau)d\tau$  and  $R(t) = Prob\{T_f > t\} = 1 - \int_0^t f(\tau)d\tau$ . If node failure follows exponential distribution, i.e.,  $f(\tau) = \lambda e^{-\lambda\tau}$ , then we have  $Q(t) = 1 - e^{-\lambda t}$  and  $R(t) = e^{-\lambda t}$ . In this case, by the definition of node reliability, the numbers  $Num_o(t)$  of operational nodes in a network at time  $t$  is

$$Num_o(t) = Num \times e^{-\lambda t} , \quad (1)$$

where  $Num$  is the number of operational nodes in the network at time 0, and thus, the number  $Num_f(t + \Delta t)$  of nodes that fails during interval  $[t, t + \Delta t]$  is

$$Num_f(t + \Delta t) = Num_o(t)(1 - e^{-\lambda\Delta t}) . \quad (2)$$

### 3.2 Sensing Range Diminishment

There exist two sensor models. One is the most commonly used binary sensor model[1, 2, 4–10]. In this model, a sensor detects with probability 1 (resp., 0) the target events happening inside (resp., outside) its sensing range, a disc centered at itself. The other is so-called stochastic sensor model[3], where the target detection probability however follows a decaying function of the distance between a target and a sensor. In this paper, we use the binary sensor model.

After a sensor is placed in the target field, it starts sensing its surroundings and participating in network operations. As the sensor operates, its battery power decreases, and its hardware wears out, therefore resulting in the performance decline of its sensing module: an originally-detectable target becomes undetectable. We model this sensibility degradation phenomena as nodal sensing range diminishment. For an arbitrary wireless sensor having been operating for  $t$  time units, its sensing range can be computed by a monotonically decreasing *sensing range function*  $f(E, t)$ , where  $E$  denotes the sensor's remaining energy level. This sensing range function is heavily affected by the material and the

hardware technology that the sensor uses. Under this circumstance, sensor sensing range function is very likely to be different for different types of sensors and should be determined on an empirical basis rather than through theoretical analysis. Nodal sensing range diminishment can be easily computed once sensing range function is defined.

## 4 The Proposed Scheme

In this section, we will present an integrated self-deployment and coverage maintenance scheme. We first state assumptions, give an overview on the scheme, and then elaborate on scheme detail.

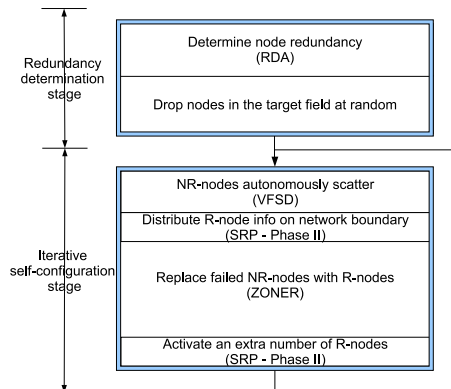
### 4.1 Assumptions

The proposed scheme works under the following assumptions:

1. Nodes are homogeneous. They initially have the same amount  $E$  of energy, and their communication radii are at least twice their sensing radii.
2. Each node is associated with a unique ID and aware of its global coordinate as well as its remaining energy level.
3. Nodes fail following exponential distribution at failure rate  $\lambda$ .
4. Nodal sensing range decreases over time, while nodal communication range keeps constant.
5. Every node executes an effective routing protocol and a sleeping/wakeup protocol enabling R-nodes (i.e., redundant nodes) to receive messages from NR-nodes (i.e., non-redundant nodes).
6. The number  $n$  of NR-nodes and the expected network operating period  $T$  are known a priori.
7. The sensing range function  $f(., .)$ , the average per-time-unit energy consumption  $\Delta E$  of a NR-node and that  $\Delta E'$  of a R-node are empirically determined beforehand. And,  $\Delta E \geq \Delta E'$ .

### 4.2 Overview

The proposed scheme is a framework constructed on top of four algorithms: a redundancy determination algorithm (RDA), a virtual-force-based self-deployment algorithm (VFSD), a zone-based sensor relocation protocol (ZONER)[10], and a sensor replenishment protocol (SRP). Its objective is to enable a mobile sensor network (MSN) to achieve maximal sensing coverage after initial node placement and maintain the achieved coverage in the presence of coverage loss. The execution of the proposed scheme is composed of two stages, a *redundancy determination stage* and an iterative *self-configuration stage*. The redundancy determination stage involves human interference and takes place foremost. During this stage, the network administrator runs the RDA to estimate the number  $n'$



**Fig. 1.** The full execution cycle of the scheme

of R-nodes needed for coverage maintenance during the expected network operating period  $T$ ; afterward, he/she drops  $n$  NR-nodes together with  $n'$  R-nodes in the target field at random.

What follows is the iterative self-configuration stage. Throughout this stage, each NR-node maintains a neighboring map by listening to a periodical HELLO message carrying sender's coordinate and sensing range from its every neighboring NR-node; R-nodes stay "sleeping" most of time by executing a sleeping/wakeup protocol. All the iterations of this stage have equal length, and they together constitute the whole network operating period. In an arbitrary iteration, three algorithms, the VFSD, the ZONER, and the SRP, are executed. The VFSD is run only by NR-nodes at the beginning of the iteration. Through the VFSD, NR-nodes moves around to close the gap and open the overlapping between their sensing ranges, therefore maximizing the network overall coverage. After the VFSD terminates, both the ZONER and the SRP starts. By the ZONER, failed NR-nodes are timely replaced with R-nodes in a one-to-one fashion; by the SRP, boundary nodes collect R-node information and active R-nodes to make up the coverage loss caused by sensing range diminishment. We would indicate that the coverage obtained by the VFSD during the very first iteration is the so-called *target coverage* that the scheme tries to maintain in the rest of network lifetime. The full execution cycle of the scheme is shown in Figure 1.

### 4.3 Scheme Detail

From Section 4.2, we know that the four algorithms, i.e., the RDA, the VFSD, the ZONER[10] and the SRP, constitute the core of the proposed scheme. In this section, we are going to go through the detail of these algorithms.

**Redundancy Determination Algorithm** This algorithm, denoted by RDA, is designed for estimating coverage loss and determining node redundancy in

advance of actual node dropping. Before going into the detail of the algorithm, Let us first look at the following important definitions:

1. *Target coverage*  $\mathcal{C}$  is the coverage that a mobile sensor network (MSN) achieves by the VFSD during the very first iteration of the self-configuration stage of the scheme.
2. *Potential coverage*  $\mathcal{P}$  is the maximal coverage that a MSN could possibly obtains through geographical reorganization. It must not be less than the network's actual coverage.
3. *Coverage gain*  $\mathcal{G}$  is the difference between the target coverage and the potential coverage of a MSN.

The RDA splits the operating period  $T$  of the network evenly into  $k$  consecutive time slots, each of which is composed of  $t$  time units and matches an iteration of the self-configuration stage of the scheme. Denote the  $j$ -th time slot (or, the  $j$ -th iteration) by  $TS_j$ . Let  $S_j$  represent the set of R-nodes, which are injected in the network by the ZONER[10] and the SRP during  $TS_j$  and are still functioning at the consideration time. Because of node failure, the size of  $S_j$  reduces over time. Denote the initial size of  $S_j$  and the size of  $S_j$  at the beginning of  $TS_i$  ( $i > j$ ) respectively by  $Sz_j^j$  and  $Sz_j^{i-1}$ . To be consistent with above denotation, let  $S_0$  represent the initial set of NR-nodes, and apparently,  $Sz_0^0 = n$ . Taking into account R-node failure and according to formula (1), the total number  $n'$  of R-nodes needed for maintaining the target coverage  $\mathcal{C}$  during entire  $T$  should satisfy the inequality  $(\dots((n'e^{-\lambda t} - Sz_1^1)e^{-\lambda t} - Sz_2^2)\dots)e^{-\lambda t} - Sz_k^k \geq 0$ . Solving this inequality, we get

$$n' \geq \sum_{j=1}^k Sz_j^j e^{jt\lambda} . \quad (3)$$

Now, let us focus our attention on how to determine  $Sz_i^i$  for ( $1 \leq i \leq k$ ). As all the failed NR-nodes are replaced with R-nodes in a one-to-one fashion, the original size  $Sz_i^i$  of the set  $S_i$  of R-nodes added in the network during  $TS_i$  will be at least the number  $Num_f^i$  of failed NR-nodes during  $TS_i$ . Namely,

$$Sz_i^i = Num_f^i + X^i , \quad (4)$$

where  $X^i$  is a non-negative number whose value depends solely on whether the network's potential coverage after node replacing is smaller than the target coverage  $\mathcal{C}$ . The set of nodes constituting the network at the beginning of  $TS_i$  is the union of all the  $S_j$ 's ( $j < i$ ). According to formula (2),  $Num_f^i$  will be

$$Num_f^i = \sum_{j=0}^{i-1} (Sz_j^j (1 - e^{-\lambda t})) , \quad (5)$$

where  $Sz_j^i = Sz_j^j e^{-(i-j)\lambda t}$  represent the size of  $S_j$  at the beginning of  $TS_i$ . Let  $\mathcal{G}^{i-1}$  be the coverage gain during time slot  $TS_{i-1}$ . Denote by  $L^i$  the total coverage



loss, by  $C_f^i$  the compensating coverage from the replacements of failure NR-nodes, and by  $A^i = f(E - (i - 1)t\Delta E', (i - 1)t)$  the average sensing range of a R-node, during  $TS_i$ . Then,  $X_i$  is given by

$$X^i = \begin{cases} 0, & \text{if } L^i \leq (\mathcal{G}^{i-1} + C_f^i); \\ \left\lceil \frac{(L^i - \mathcal{G}^{i-1} - C_f^i)}{A^i} \right\rceil, & \text{otherwise.} \end{cases} \quad (6)$$

For time slot  $TS_i$  ( $i \geq 1$ ), let  $A_j^i$  represent the average sensing range of a NR-node in  $S_j$  ( $j \leq i$ ) and  $\Delta A_j^i = A_j^i - A_j^{i+1}$  its average sensing range diminishment. Define  $A_j^i = \begin{cases} f(E - (j - 1)t\Delta E' - (i - j)t\Delta E, (i - 1)t), & \text{for } j \geq 1; \\ f(E - (i - 1)t\Delta E, (i - 1)t), & \text{for } j = 0. \end{cases}$  Then, the coverage loss due to node failure is  $L_f^i = \sum_{j=0}^{i-1} (Sz_j^i A_j^i (1 - e^{-\lambda t}))$ , and the coverage loss due to sensing range diminishment is  $L_d^i = \sum_{j=0}^{i-1} (Sz_j^i \Delta A_j^i e^{-\lambda t})$ . Consequently, the total coverage loss  $L^i$ , i.e., the summation of  $L_f^i$  and  $L_d^i$ , is

$$L^i = \sum_{j=0}^{i-1} (Sz_j^i (A_j^i (1 - e^{-\lambda t}) + \Delta A_j^i e^{-\lambda t})) . \quad (7)$$

Assume that the VFSD yields a node distribution with no sensing range overlapping. Additionally, we define  $A_0^0 = A_0^1$ . The target coverage is just the sensing range aggregation of all the NR-nodes in  $S_0$  during  $TS_1$ , namely,  $\mathcal{C} = nA_0^1$ . Hence, the coverage gain  $\mathcal{G}^{i-1}$  during time slot  $TS_{i-1}$  is

$$\mathcal{G}^{i-1} = \sum_{j=0}^{i-1} (Sz_j^{i-1} A_j^{i-1}) - nA_0^1 . \quad (8)$$

The compensating coverage  $C_f^i$  from failure node replacements in  $TS_i$  is

$$C_f^i = Num_f^i A^i = \sum_{j=0}^{i-1} (Sz_j^i A^i (1 - e^{-\lambda t})) , \quad (9)$$

where where  $A^i$  is the average sensing range of a R-node during  $TS_i$ .

By formula (4) – (9), the RDA estimates each  $Sz_i^i$  ( $1 \leq i \leq k$ ) in the increasing order of  $i$  and then computes  $n'$  using formula (3). Besides, a *redundancy table* as side-product is created and stored at every single node during the execution of the RDA. This table records the mapping between time slot  $TS_i$  and its corresponding  $X_i$  for every possible  $i$ , and it is going to be used by the SRP to determine how many extra R-nodes need to be activated in each time slot.

**Virtual-Force-based Self-Deployment Algorithm** All the existing distributed sensor self-deployment algorithms (e.g., [1–6]) assume equal and constant nodal sensing range and thus is not suitable for our scheme where nodal sensing radii decrease over time. We develop a Virtual-Force-based Self-Deployment algorithm, denoted by VFSD, without such an assumption.

The VFSD is executed only by NR-nodes. It makes NR-nodes able to autonomously spread out to form a network, and in order for the network to have as-large-as-possible coverage, it attempts to keep the distance between any two neighboring NR-nodes equal to the summation of their sensing radii. Because the virtual-force-based type of self-deployment algorithms are so sensitive to node failure as to cause frequent topology change and thus large amount of energy loss, in our scheme, the VFSD does not keep effective all the time but run only at the beginning of each iteration of the self-configuration stage.

By the VFSD, a NR-node receives virtual force only from its neighboring NR-nodes. Consider an arbitrary pair of neighboring NR-nodes  $N_i$  and  $N_j$ . Let  $r_i$  and  $r_j$  respectively denote the sensing radii of  $N_i$  and  $N_j$ , and let  $XY_i$  and  $XY_j$  respectively represent the coordinates of  $N_i$  and  $N_j$ . Furthermore, define

$$\delta_{i,j} = |d_{i,j}^{\vec{}}| - r_i - r_j ,$$

where  $d_{i,j}^{\vec{}}$  stands for the distance from  $N_i$  to  $N_j$  and is given by  $d_{i,j}^{\vec{}} = XY_i - XY_j$ . In the case of  $\delta_{i,j} < 0$ , we model the two NR-nodes as electriferous particles that exert repulsive force on each other, while in the case of  $\delta_{i,j} > 0$ , we model them as massive matters that exert gravitational force on each other. In either of the two cases, the magnitude of virtual force is computed following Newton's Law of Gravitation. We consider that there is no virtual force between  $N_i$  and  $N_j$  if  $\delta_{i,j} = 0$ , because their total coverage is maximized when their sensing ranges adjoin without overlapping.

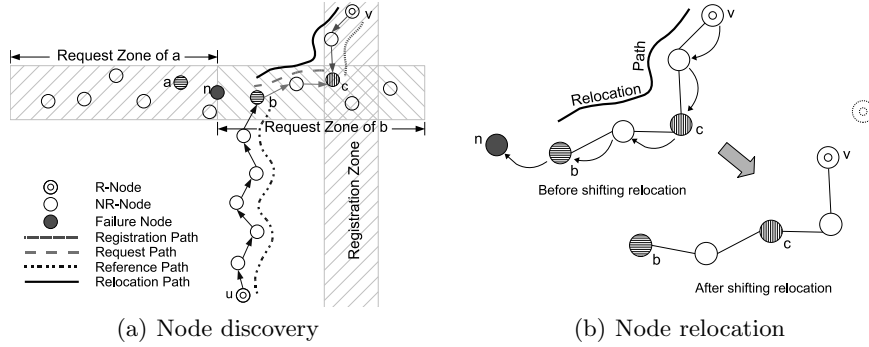
Since Newton's Law of Gravitation is a function of mass, we treat a node as a massive sphere of its sensing radius. Suppose that the density of a node is  $\rho$ . The virtual mass  $M_i$  of  $N_i$  is  $M_i = 4\pi r_i^2 \rho$ . If we define the *virtual force constant*  $K$  as  $K = G(4\pi\rho)^2$  where  $G$  is Newton's constant, for any two neighboring NR-nodes  $N_i$  and  $N_j$ , the force  $\vec{F}_i^j$  that  $N_j$  exerts on  $N_i$  will be

$$\vec{F}_i^j = \begin{cases} K \left( \frac{r_i r_j}{\delta_{i,j}} \right)^2 \frac{d_{j,i}^{\vec{}}}{|d_{j,i}^{\vec{}}|} , & \text{if } \delta_{i,j} > 0 ; \\ \vec{0} , & \text{if } \delta_{i,j} = 0 ; \\ -K \left( \frac{r_i r_j}{\delta_{i,j}} \right)^2 \frac{d_{j,i}^{\vec{}}}{|d_{j,i}^{\vec{}}|} , & \text{if } \delta_{i,j} < 0 . \end{cases} \quad (10)$$

The total virtual force  $\vec{F}_i$  exerted on node  $N_i$  is the vector summation of the virtual force that  $N_i$  receives from all its neighboring nodes. Let  $NS_i$  denote  $N_i$ 's neighbor set. Then,  $\vec{F}_i$  is given by

$$\vec{F}_i = \sum_{N_j \in NS_i} \vec{F}_i^j . \quad (11)$$

To compute  $\vec{F}_i$  using formula (10) and (11), node  $N_i$  must know both the  $r_j$  and the  $XY_j$  of every  $N_j$ , which are in fact available in its neighborhood map. Driven by  $\vec{F}_i$ ,  $N_i$  moves toward the direction of  $\vec{F}_i$ . The movement of  $N_i$  in turn causes the change in  $\vec{F}_i$ . This mutual effect leads to  $N_i$ 's unpredictable migration itinerary. Node  $N_i$  stops moving when it reaches either a static or a



**Fig. 2.** An illustration of how the ZONER works

dynamic equilibrium status. The former is the situation that  $\vec{F}_i = \vec{0}$ ; the latter is the situation that  $N_i$  fluctuates between several positions, and in this case,  $N_i$  stops at the centroid of those positions. Once  $N_i$  stops moving, it notifies all its NR-node neighbors. When  $N_i$  finds that its neighborhood is stabilized, it becomes fixed and starts the relocation protocol ZONER[10].

**ZONE-based Sensor Relocation Protocol** In each iteration of the self-configuration stage, the ZONER[10] starts after the termination of the VFSD and stops after the termination of the SRP. Its execution consists of three core processes, i.e., node registration, node discovery, and node relocation. These processes are performed using a restricted flooding technique, ZFlooding, to save energy and messages. The node registration process is executed first. During this process, a R-node floods its unbounded vertical *registration zone* with a registration message to register with all the NR-nodes inside the zone. After a NR-node failed, its westmost neighbor and eastmost neighbor respectively initiates a node discovery process by flooding their bounded horizontal *request zones* with a request message to find a replacement for it. The westmost neighbor and the eastmost neighbor are called *discovery partner* of each other, and their request zones are adjacent by an imaginary line vertically across the failed node. During a node discovery process, the process initiator first searches its local memory space for the registered R-node with shortest relocation path, and then takes this R-node as reference to inquires all the NR-nodes inside its request zone for a R-node with yet shorter relocation path. For message-saving purpose, the length of the request zone is made subject to the reference node's relocation path length. Because the request zone intersects with a number of registration zones, the NR-nodes in the intersection areas may be able to reply the initiator's request as *recommender*. Finally, the initiator chooses the one with shortest relocation path among all the discovered available R-nodes as the failure node's replacement candidate. Having found the replacement candidate, the initiator communicates with its discovery partner to determine the official replacement

node. Figure 2(a) is a big picture about a discovery process. Sequentially, the replacement discoverer triggers a relocation process by a relocation message. In this process, the nodes along the replacement node’s relocation path relocate in a shifting manner to replace the failed node. That is, every node in the path simultaneously moves to the location of its path neighbor toward the replacement node discoverer, while the replacement discoverer moves to the location of the failed node as illustrated in Figure 2(b). After such a relocation process, the failed node is in fact replaced by the replacement node discoverer rather than by the replacement node itself. Once a R-node actually involves in a relocation process, it becomes active and transforms to a NR-node.

**Sensor Replenishment Protocol** The sensing holes caused by failure NR-nodes are filled with R-nodes by the relocation protocol ZONER[10], while the other factor of coverage loss, i.e., nodal sensing range diminishment, still remains untreated. To compensate the coverage loss due to sensing range diminishment, extra R-nodes may have to be released into the network. However, with the absence of centralized controller, where to look for R-nodes and how to release R-nodes become an issue. Under this circumstance, we devise a sensor replenishment protocol, denoted by SRP. The execution of the SRP consists of two phases, i.e., the *node registration phase* and the *node activation phase*, respectively answering the “where” and the “how” question.

*Node Registration Phase* starts at the beginning of an iteration of the self-configuration stage. In this phase, the SRP, through a Greedy-Face-Greedy (GFG) routing mechanism[13, 14], distributes R-node information onto the outer face perimeter of a Gabriel graph (GG) constructed over the network.

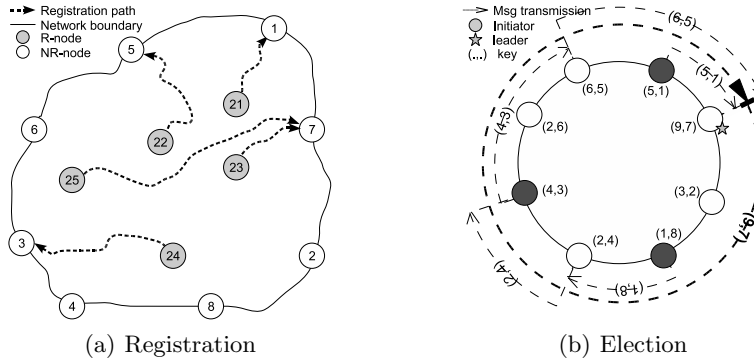
A gabriel graph (GG) is a planar graph, where the closed diametral disc of each edge contains no other vertices than the two edge ends[15]. A GG-construction algorithm, which takes a connected graph  $G$  as input and outputs a GG  $G'$  spanning  $G$ , can be the following: remove non-GG edges from  $G$  by testing every edge using the GG definition; an edge  $e$  remains in  $G$  iff it passes the GG test; finally,  $G$  becomes  $G'$ . Hence, a GG can be easily built over a connected network in a localized and distributed fashion without message transmission, as long as each network node knows about the position (coordinate) of its every neighboring node. This is just the case in our proposed scheme since each NR-node maintains its neighborhood map. In a GG network, the outer face perimeter is called network boundary. Without losing generality, the network boundary can be modeled as a ring, denoted by  $R$ . The network boundary has a special property, that is, *it contains all the global directional optima*. What it is trying to say is that the globally foremost node in certain direction, e.g., the northmost node, must be on the network boundary. This property is referred to as *network boundary property* by us. Its correctness follows the fact that all the nodes but boundary nodes reside in the area surrounded by the network boundary. To avoid ambiguity, directional foremostness must be explicitly defined beforehand, and tie must be broken according to some policy.

When a R-node  $RN_r$  wishes to register on the network boundary, it randomly picks a direction as its registration direction  $RD_r$ , and sends a *registration message* carrying its ID, coordinate, energy remaining level and its registration direction  $RD_r$  to its foremost NR-node neighbor in  $RD_r$ . The randomization here is for the purpose of load balancing among boundary nodes. This registration message is routed in a GFG manner [13, 14]. Specifically, after a NR-node  $N_i$  receives the registration message of  $RN_r$ , it first obtains  $RD_r$  from the message and then greedily forwards the message to its own foremost NR-node neighbor in  $RD_r$ . In the case that  $N_i$  itself is the foremost in  $RD_r$  among its neighborhood, it attaches its ID and coordinate to the registration message and retransmits the message in face routing mode, and thereafter, the message keeps being processed in face routing mode until it reaches a yet-furthest NR-node  $N_j$ , which will resume the greedy message transmission. When the globally furthest NR-node  $N_k$ , which is a boundary node according to the *network boundary property*, in direction  $RD_r$  receives the registration message, there are two cases to be explored. One is that  $N_k$  knows about the fact that it itself is a boundary node, while the other is that it does not. In the former case,  $N_k$  just records the information about  $RN_r$  retrieved from the message. In the latter case,  $N_k$  tries to find a node yet furthest in  $RD_r$  by retransmitting the message along  $R$  in face routing mode. Since  $N_k$  is actually the global directional optimum, the message will traverse all the way  $R$  and get back to  $N_k$  at the end. After  $N_k$  receives the message back, it becomes aware of its role of boundary node, and then stores  $RN_r$ 's information as well as notifies all the other boundary nodes of their role through message relay along  $R$ . Figure 3(a) shows an example of the node registration phase.

Note that, if the VFSD algorithm (refer to Section 4.3) does not yet globally terminate, the constructed GG will not be stable, resulting in the failure of the node registration process introduced above. Hence, the SRP requires that NR-nodes ignore any registration message before they become fixed, and that boundary nodes reply R-nodes' registration request to confirm their successful registration. Under this circumstance, if a R-node does not receive any response after sending a registration message, it "sleeps" for a while and then tries to register once again. When many registration retrials happens, the time interval between two successive ones has incremental length. Once a R-node finds that it succeeds in registration, it turns off to save energy.

*Node Activation Phase* starts at the end of each iteration of the self-configuration stage. In this phase, the SRP elects a boundary node as leader, which then activates a specified number  $k$  of R-nodes. The number  $k$  is determined by the leader using the index of current iteration and its locally stored *redundancy table* (see Section 4.3). Considering the possible insufficiency in the R-nodes that a single boundary node (i.e., the leader) can activate, the node activation phase is executed recursively until  $k$  R-nodes are successfully injected into the network.

Denote by  $N_i$  an arbitrary boundary node, by  $id_i$  the ID of  $N_i$ , and by  $v_i$  the number of R-nodes currently registering with  $N_i$ . Furthermore, define the key  $K_i$  of  $N_i$  as the value pair  $(v_i, id_i)$ . For two keys  $K_i$  and  $K_j$ , we define  $K_i < K_j$  for  $(v_i < v_j) \vee (v_i = v_j \wedge id_i < id_j)$ . When  $t$  time units elapse since the start of



**Fig. 3.** An illustration of how the SRP works

current iteration,  $N_i$  spontaneously initiates the node activation phase. Taking into account the inaccuracy of local lock,  $N_i$  first polls among a collection of NR-nodes. This collection of NR-nodes can be randomly selected or predefined (e.g., one-hop neighbors). An extreme case is that it includes all the NR-nodes in the network.  $N_i$  actually initiates the node activation phase iff majority of polled NR-nodes agree.

Node  $N_i$  starts the node activation phase by sending a *start message* carrying its key  $K_i$  and the  $k$  along  $R$ . After a NR-node  $N_j$  receives a start message, it compares its own key  $K_j$  with the key, denoted by  $K$ , retrieved from the message. If  $K_j < K$ ,  $N_j$  simply forwards the message to its next hop; otherwise,  $N_j$  updates the key in the message with  $K_j$  and retransmits the message along  $R$  iff it is not an initiator. The start message with largest key will traverse entire  $R$  and get back to its generator, which is then becomes the leader. Figure 3(b) shows an example of the leader election process. We would like to indicate that this lead election method is by no means the optimal one. We use it only because of its simple description. Leader election is a classic and well-studied problem of distributed computing, and a number of solutions have been proposed for ring networks in literature. Reference [16] provides a systematical study on existing leader election algorithms.

The elected leader picks  $k$  closest registered R-nodes, sends them an *activation message*, and waits for their replies. If the number of replying R-node is less than  $k$ , the leader will try to activate other locally registered R-nodes in the same way. Both replying R-nodes and unreplying R-nodes are removed by the leader from future consideration. The leader's activation attempt stops when the total number of replies is equal to  $k$ , or when no more registered R-nodes are available. In the latter case, the leader updates  $k$  with  $k - v$  where  $v$  represents the total number of replies it receives and restarts the leader election process.

The last elected leader in above recursive process notifies all the NR-node and R-node of the termination of current iteration via a flooding process. Thereafter, the SRP terminates, and the self-configuration stage enters its next iteration.

## 5 Evaluation Issues

We expect that the proposed scheme is able to maximize the coverage of a mobile sensor network (MSN) and maintain achieved coverage for a given period of time using redundant sensors. The RDA estimates coverage loss based on probability and assumes that the VFSD yield a node distribution with no sensing range overlapping. However, in reality, the degree of coverage loss can not be exactly as what the RDA expects, and the VFSD is not likely to generate such a perfect node distribution either. There must be certain level of difference between the actual performance of the scheme and our expectation. We would like to study the difference by experiments.

There exists two simple variant of the proposed scheme. The first variant  $V_1$  does not use the sensor relocation protocol in its self-configuration stage and thus reduce a large amount of communication overhead. In this variant, at the end of an iteration of the self-configuration stage, R-nodes are injected into the network by the SRP, and then the network is reorganized by the VFSD at the beginning of the next iteration. Since the coverage loss caused by failed nodes is compensated in a deferred manner, the average coverage in each iteration of  $V_1$  should be lower than that of the original scheme. As for the second variant  $V_2$ , it uses the VFSD to obtain the target network coverage, and then employs only the ZONER in each iteration of its self-configuration stage to compensate coverage loss. Compared with the original scheme,  $V_2$  maintains coverage in a more timely and energy-efficient fashion; but, due to the lack of network reorganization, sensing range overlapping may happens everywhere in the network, wasting redundant nodes. Under these circumstances, we will conduct an extensive set of experiments to evaluate the proposed scheme in comparison with its two variants  $V_1$  and  $V_2$ .

## 6 Conclusion and Future Work

In this paper, we discussed the two main reasons, *node failure* and *sensing range diminishment*, for coverage loss in sensor networks, and proposed an integrated self-deployment and coverage maintenance scheme for mobile sensor networks (MSNs). The proposed scheme is a combination of an existing algorithm, i.e., the ZONER[10], and three new algorithms, i.e., the RDA, the VFSD and the SRP. It provides a guidance to systematically estimate coverage loss and determine node redundancy in advance of actual node dropping, and it empowers a MSN to gain maximal coverage and maintain the achieved coverage with the assistance from redundant sensors for an expected period of time. We noticed that our scheme is the first one that considers the impact from nodal sensing range diminishment when analyzing coverage loss.

From Section 4.3, we can find the proposed scheme has the following incompleteness: 1) the ZONER[10] and the SRP functionally overlap each other to some extent in their node registration processes; 2) the shifting relocation strategy of the ZONER may impair network coverage because of the sensing range difference among the nodes along a relocation path; 3) that the SRP is vulnerable to boundary node failure. The smooth integration of the ZONER into the

proposed scheme and the development of a fault-tolerance enabling mechanism for the SRP will be part of our future work. As well, we plan to evaluate the performance of the scheme through experiments, in accordance with the evaluation issues depicted in Section 5.

## Acknowledgments

The authors would like to thank Dr. Ivan Stojmenovic for his valuable discussions on improving this work, and the anonymous reviewers for their useful comments.

## References

1. A. Howard, M. J. Mataric, and G. S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks". *Autonomous Robots*, 13(2):113-126, 2002.
2. 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 *Proc. of DARS*, pp. 299-308, 2002.
3. Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces". In *Proc. of IEEE INFOCOM*, vol. 2, pp 1293-1303, 2003.
4. G. Wang, G. Cao, and T. L. Porta, "Movement-Assisted Sensor Deployment". In *Proc. of IEEE INFOCOM*, vol. 4, pp. 2469-2479, 2004.
5. N. Heo and P. K. Varshney, "Energy-Efficient Deployment of Intelligent Mobile Sensor Networks". *IEEE Tran. on Systems, Man, and CyberNetics - Part A: Systems and Humans*, 35(1):78-92, 2005.
6. J. Wu and S. Yang, "SMART: A Scan-Based Movement-Assisted Sensor Deployment Method in Wireless Sensor Networks". In *Proc. of IEEE INFOCOM*, vol. 4, pp. 2313- 2324, 2005.
7. S. Chellappan, X. Bai, B. Ma, and D. Xuan, "Sensor Networks Deployment Using Flip-based Sensors". In *Proc. of IEEE MASS*, 2005.
8. G. Wang, G. Cao, and T. L. Porta, "Proxy-Based Sensor Deployment for Mobile Sensor Networks". In *Proc. of IEEE MASS*, pp. 493-502, 2004.
9. G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks". In *Proc. of IEEE INFOCOM*, pp. 2302-2312, 2005.
10. X. Li and N. Santoro, "ZONER: A ZONE-based Sensor Relocation Protocol for Mobile Sensor Networks". In *Proc. of IEEE LCN/WLN*, 2006. To appear.
11. L. E. Parker, B. Kannan, X. Fu, and Y. Tang, "Heterogeneous Mobile Sensor Net Deployment Using Robot Herding and Line-of-Sight Formations". In *Proc. of IEEE IROS*, vol. 3, pp. 2488- 2493, 2003. the 2003.
12. Y. Mei, C. Xian, S. Das, Y. C. Hu, and Y. Lu, "Replacing Failed Sensor Nodes by Mobile Robots". In *Proc. of ICDCS/WWASN*, 2006. To appear.
13. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks". In *Proc. of ACM DIALM*, pp. 48-55, 1999.
14. H. Frey and I. Stojmenovic. "On Delivery Guarantees of Face and Combined Greedy-Face Routing Algorithms in Ad Hoc and Sensor Networks". In *Proc. of ACM MobiCom*, 2006. To appear.
15. J. W. Jaromczyk and G. T. Toussaint. "Relative neighborhood graphs and their relatives". In *Proc. of the IEEE*, vol. 80, pp. 1502-1517, 1992.
16. N. Santoro. "Design and Analysis of Distributed Algorithms". In Publication, 2006.