DOI: 10.3745/KIPSTC.2009.16-C.4.535

모바일 센서노드들의 협동형 단계적 이동기법 기반의 에너지 효율적인 동적 센서네트워크 커버리지 관리

한 녹 손† · 김 성 환††

요 약

무선센서네트워크는, 다수의 센서노드가 넓은 지역에 배치되어 자신의 전원에 의해 동작하고 다른 센서노드와 협력하여 환경검침 또는 물리량 검침을 수행하는 무선네트워크로 정의한다. 각각의 센서노드들은 자신의 배터리전원소모를 최소화하여, 장기간 사용할 수 있어야 하며, 센서노드를 적절하게 배치하여 전체 네트워크 커버리지를 제공할 수 있어야 무선센서네트워크의 수명을 늘일 수 있다. 초기 센서노드의 배치로는 센서들의 배터리 수명 등의 이유로 제약이 있으며, 모바일센서노드는 센서가 이동하여 새로운 커버리지를 제공함으로써, 이러한 제약조건을 완화할 수 있다. 본 논문에서는 perimeter coverage property를 만족하는 모바일 센서들의 단계적인 이동을 통한 커버리지 제공기법을 제안한다. 각각의 모바일 센서노드들은 이웃한 센서노드들이 dead 노드인지 판단하게 되며, dead 노드인 경우, 센서네트워크의 커버리지 hole을 만드는지 여부를 판단한 후, 각각의 모바일 센서노드들은 hole의 중심점을 계산하고, 관련된 센서노드들이 협동하여, 단계적으로이동하여, 최종 hole을 커버하는 새로운 센서네트워크를 형성하게 된다. 본 제안기법을 시뮬레이션하여 DCM 기법과 비교한 결과, 에너지 효율을 결정하는 전체 움직임거리 측면에서 최소 50% 이상의 성능향상을 보임을 확인하였다.

키워드 : 무선센서네트워크, 이동성, 커버리지 문제

Collaborative Stepwise Movement of Mobile Sensor Nodes for Energy Efficient Dynamic Sensor Network Coverage Maintenance

Ngoc Soc Han† · Seong Whan Kim††

ABSTRACT

Wireless Sensor Network (WSN) is a wireless network consisting of spatially distributed autonomous devices, using sensors to cooperatively monitor physical or environmental conditions. WSNs face the critical challenge of sustaining long-term operation on limited battery energy. Coverage maintenance has been proposed as a promising approach to prolong network lifetime. Mobile sensors equipped with communication devices can be leveraged to overcome the coverage problem. In this paper, we propose a stepwise movement scheme using perimeter coverage property for the coverage maintenance problem. In our scheme, each sensor monitors neighboring dead nodes, determines vulnerable node (i.e. dead node which makes uncovered area), computes the center of uncovered area HC, and makes a coordinated stepwise movement to compensate the uncovered area. In our experimental results, our scheme shows at least 50 % decrease in the total moving distance which determines the energy efficiency of mobile sensor.

Keywords : Wireless Sensor Network, Mobility, Coverage Problem

1. Introduction

WSN is considered for several critical application scenarios including battlefield surveillance, habitat monitoring, traffic monitoring, and security applications[1]. Sensor nodes, and hence sensor applications, are subject to constraints such as limited processing, storage, communication capabilities, and limited power supplies. Recent advances in micro-sensor and communication technologies enlarge the possibility of manufacturing inexpensive small wireless sensors with simple sensing, processing and wireless communication capabilities. Limited by their size, small wireless sensors are equipped with restricted power source and storage capacity. Such small wireless sensors are usually deployed in an ad-hoc manner to monitor a specified region of interest for various applications, such as environment monitoring, target tracking and distributed data storage.

† 준 회 원 : 서울시립대학교 컴퓨터통계학과 석사과정
†† 정 회 원 : 서울시립대학교 컴퓨터과학부 부교수(교신저자)
논문접수 : 2009년 3월 10일
수 정 일 : 1차 2009년 5월 8일
심사완료 : 2009년 5월 9일

The wireless sensors in a network collaborate with eachother to monitor surrounding environment and transmit data to sink points through multi-hop communication. Despite mobility or not, WSNs face the critical challenge of sustaining long-term operation with limited battery energy. To prolong network life time requires we should minimize the energy consumption of sensors and maximize the coverage. Traditional coverage maintenance schemes aim at replacement or partial re-deployment of sensor nodes as and when coverage loss is detected. Previous works on the sensor network coverage focuses on algorithms to redistribution sensors in order to achieve a static configuration with an enlarged covered area [2, 3]. In many applications, sensors remain stationary after their initial deployment. The coverage of such a static sensor network is determined by the initial network configuration. Once the deployment strategy and sensing characteristics of the sensors are known, the network coverage can be computed and remains unchanged over time. Recently, there has been a strong requirement to deploy sensors mounted on mobile platforms such as mobile robots. Such mobile sensor networks are extremely valuable in situations where traditional deployment mechanisms fail or are not suitable, for example, a hostile environment where sensors cannot be manually deployed or air-dropped [1, 4, 5, 6].

In this paper, we focus on the sensor network deployment using sensors with limited mobility. In this configuration, we propose a coverage maintenance scheme based on perimeter coverage property. When we detect an uncovered area left by dead node, each sensor initiates stepwise movement (i.e. small movement of sensor nodes) to compensate the uncovered area. Each sensor node keeps the perimeter coverage information and broadcast relevant information over network area to identify uncovered area. To satisfy the objectives of coverage maintenance which are to minimize energy consumption of sensors and to maximize the coverage, we minimize energy usage (in terms of moving distance) and minimize the lost coverage. This paper consists of five sections. In section 2, we review the previous research works for coverage maintenance and mobility in wireless sensor networks. Section 3 presents our proposed scheme Stepwise movement. We show our experimental results in section 4, and conclude in section 5.

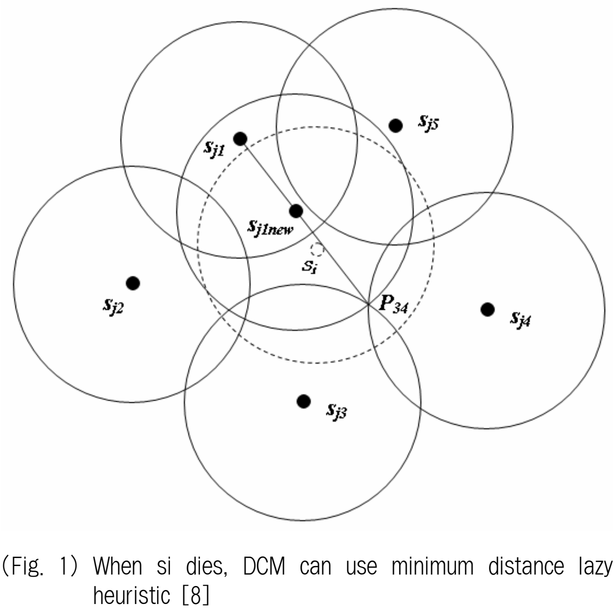
2. Related Works

One of the fundamental issues in sensor networks is the coverage problem, which reflects how well a sensor network is monitored or tracked by sensor. Coverage maintenance arises when sensor nodes die due to battery drain or environmental causes. The area which was covered by the dead node may be left uncovered, which is a potential point of vulnerability in the field. Hence, coverage maintenance is required to compensate for the death of the node. Traditional coverage maintenance schemes aim at replacement or partial redeployment of sensor nodes as and when coverage loss is detected. In static sensor networks, distributing a large number of redundant sensors is the only way to tolerate node failures [8]. Recently, mobility in sensor networks has increased, especially in the context of deployment, because it is useful for tracking of intrusion, and event-based mobility control. Mobile sensors called Robomotes [6] have already been constructed. The authors of [9] describe a method of self-deployment of sensors with mobility. The objective of deployment is to achieve near-uniform coverage over a previously unknown environment. Nodes are treated as virtual particles and are repelled by virtual forces from other nodes, and from obstacles. As a result of these virtual forces, nodes from an initial compact configuration covering a very small area eventually spread out uniformly. As in [10] Co Grid (Coordinating Grid) organizes the network into coordinating fusion groups located on overlapping virtual grids. Through coordination among neighboring fusion groups, CoGrid can achieve comparable number of active nodes as a centralized algorithm, while reducing the network reconfiguration time by orders of magnitude. CoGrid is especially suitable for large and energy-constrained sensor networks that require quick re-configuration in response to node failures and environmental changes. Another grid-based approach but for mobile WSN is mesh-based sensorrelocation protocols [11] that can be employed as fault tolerance approach to offset the coverage loss caused by node failures. The authors proposed a mesh-based sensor relocation protocol (MSRP) for mobile sensor networks. MSRP maintains a sensor network's sensing coverage by replacing failed sensors with nearby redundant ones using near optimal time delay and balanced energy consumption. Simulation indicates that it guarantees closest node replacement with high probability (> 96%) and with considerably low message overhead.

Another approach is dynamic coverage maintenance (DCM) schemes [8] that exploit the limited mobility of the sensor nodes. The main objective of DCM is to compensate the loss of coverage with minimum expenditure of energy. DCM schemes can be executed on individual sensor nodes having knowledge of only their local neighborhood topology. Four heuristics were used to decide which neighbor to migrate, and to what distance, such that the energy expended is minimized and the coverage obtained for a given number of live nodes is maximized. In maximum energy based (MEB) heuristic, the neighboring node for migration is determined by comparing their maximum available energy; in min-max distance (MMD) heuristic the neighbor with minimum of moving range (the maximum distance a node can move) is chosen for migration; while minimum distance per energy (MDE) combines the objectives of MED and MMD heuristics by calculating the ratio between the maximum distance they can move and available energy. The last heuristic, minimum distance lazy (MDL) carries out computations to locate the new position (s_{j1new} in Figure 1) of neighboring nodes that doesn't break the coverage of the network (P_{34} is limit of migration for sensor s_{j1}). Then the neighbor which has to move the minimum distance to reach its new position is chosen.

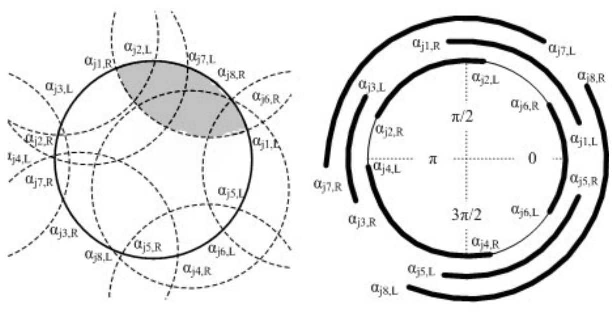
K-coverage problem [3, 4, 7] is to determine whether every point in the service area of the sensor network is covered by at least k sensors, where k is a given parameter. For example, we can check whether the arc $\widehat{\alpha_{j1,R}\alpha_{j1,L}}$ between two sensor nodes as shown in Figure 2(a) is k-perimeter-covered.

The coverage can be represented from targeting sensor s perspective as shown in Figure 2(b). For the given



(Fig. 1) When si dies, DCM can use minimum distance lazy heuristic [8]

sensor s, we can make a list of covered segments (arcs) and by traversing that list of segment, and we can determine that each point on perimeter belong to at least, for example k, segments. Once all the points on perimeter belong to k segments, we can assure that the sensor node is k perimeter-covered. In our scheme, we use a special case (k = 1) of perimeter coverage introduced in [7] to deal with coverage problem.



(a) (b)

(Fig. 2) K-perimeter-coverage maintenance [7]. (a) segment of sensor s_i 's perimeter covered by s_{jn} ($n=\overline{1..8}$), and (b) the perimeter-coverage for the sensor s_i's perimeter. For each sensor s_{jn} such that d(s_i, s_{jn}) $\leq$ 2r , determine the angle of s_i's arc, denoted by [$a_{jn,L}$, $a_{jn,R}$], that is perimeter-covered by s_{jn}

3. Collaborative Stepwise Movement Scheme for Dynamic Coverage Maintenance

In our configuration, we have a sensor network that consists of a large number of small, light-weight, highly power-constrained wireless nodes. Sensors are equipped with processors, memory and power sources. Each sensor periodically broadcasts HEARTBEAT signal. If any node sd stops broadcasting the signal to the network, neighboring sensor nodes can determine that the node sd is dead. Then, neighboring sensor nodes starts to check s whether it is vulnerable node v (i.e. dead node which makes uncovered area) and compute the hole center HC (i.e. center of uncovered area) to make them stepwise move for the lost coverage maintenance.

3.1 Computation of Perimeter Coverage

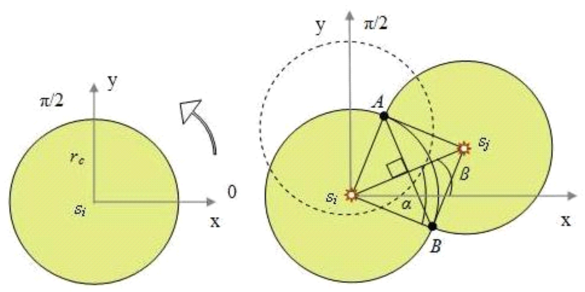
The first step to deal with coverage maintenance problem is to determine the network coverage (or detect the coverage loss). We use the perimeter coverage property presented in [7], in which, (thatis usually in complicated shape), the approach observes how the perimeter of each sensor's sensing range is covered instead of determining the coverage of each sub-region. By collecting the perimeter coverage information from neighboring nodes, each sensor can detect the coverage loss which occurs at its surrounding. The advantage of using perimeter coverage property is that each sensor node only stores and manages perimeter coverage, which is easily determined by simple geometry calculation.

To consider the whole sensing network area, we use co-ordinates system *Oxy*. Each sensor node *s* also uses a polar coordinate system (PCS) as shown in Figure 3(a). Segments of sensor s covered by neighboring sensors are determined in PCS. We set $\alpha(\alpha > 0)$ as the half of angle created by intersection of sensing ranges of two neighboring sensor nodes; and β as angular coordinate of sensor s_i (radial coordinate of sensor is always its sensing range r_c) in the sensor s_i's PCS. The sensor s_i is under the calculation. s_j is one of the neighbors of s_i, s_j covers the segment AB in the perimeter of s_i, we compute α and β as equation (1). After that, we can compute the two ends of sensor s_i's segment AB (created by neighboring sensor s_j) as the rotation of $s_j$counter-clockwise and clockwise respectively (in PCS), therefore A and B coordinates are simply calculated by values of α and β as $A = \beta + \alpha, B = \beta - \alpha$.

$$\alpha = \arccos\frac{d(s_i, s_j)}{2r_c}$$

$$\beta = \begin{cases} \frac{\pi}{2}\frac{dy}{|dy|}, & dx = 0 \\ \arctan\frac{dy}{dx} + \pi, & dx < 0 \\ \arctan\frac{dy}{dx} + \left(1 - \frac{dy}{|dy|}\right)\pi, & dx > 0, dy \neq 0 \\ 0, & dx > 0, dy = 0 \end{cases} \quad (1)$$

where $dx = x_j - x_i, dy = y_j - y_i$

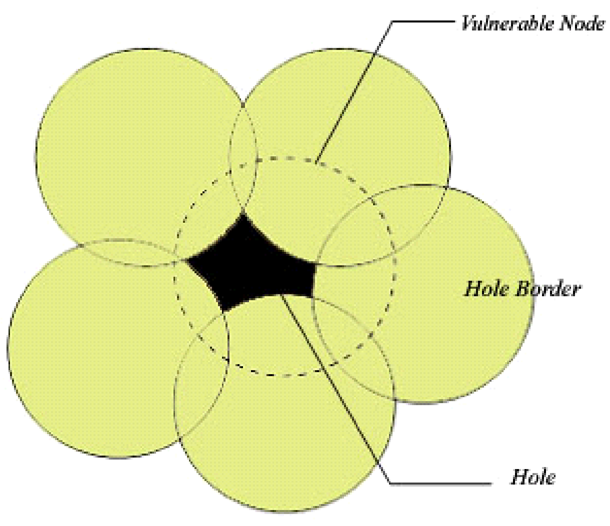


(Fig. 3) Computation of A and B for perimeter coverage: (a) sensor s_i (x_i, y_i)'s PCS, (b) determination of sensor s_i's perimeter segment covered by $s_j(x_j, y_j)$, $dx = x_j - x_i, dy = y_j - y_i$

3.2 Detection of Vulnerable Node

When a node dies, it is possible that a part of the network which was earlier covered by the node now does not belong to the coverage area of any sensor, as shown in Figure 4. The dead sensor node leaving an uncovered network area (*Hole*) is called *vulnerable node* of the network. And the hole is bound by perimeter segments of neighbor nodes of the dead node. The nodes making up the border of the *Hole* are called *Hole Border*. The *Hole* is made up by segments of circles, we try to determine the center of the *Hole* in the sense of the optimized inner point of the Hole that makes the proposed stepwise movement optimal, denote it H_C.

Because a dead node does not always make uncovered area (i.e. when we have high density of sensor nodes in a network, the dead node may not make coverage loss), we should determine the dead node makes uncovered area before we ask neighboring sensors to move for coverage maintenance. Vulnerability check for the dead node ensures that the dead node's neighbor only moves when the dead node actually makes coverage loss. If we determine that the dead node makes uncovered area (i.e. the dead node is vulnerable node in the network), we ask the neighboring sensors nodes to move for coverage maintenance. This task is done by each neighboring nodes by checking its perimeter coverage with assumption that the node is dead. By that coverage property of neighboring sensor reflects vulnerability of the node. Specifically, if neighboring node is covered then the node is not vulnerable, if neighboring node is uncovered, it is vulnerable.

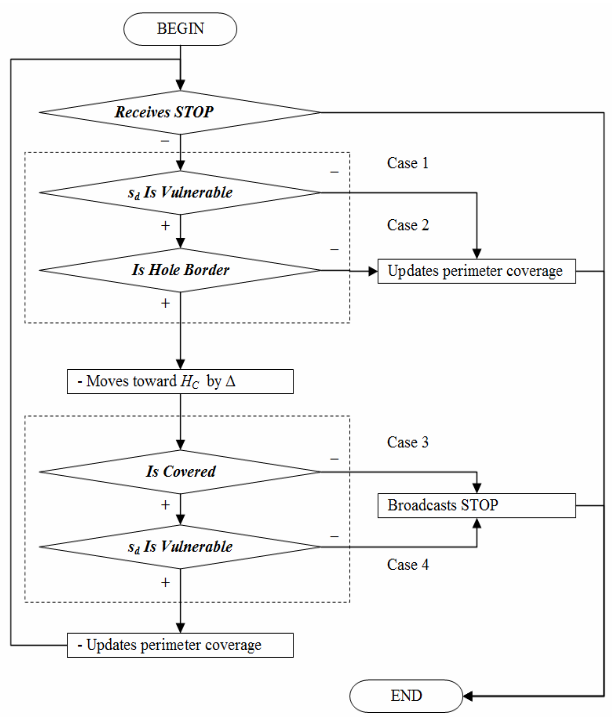


(Fig. 4) The dead node makes uncovered area, it is identified as vulnerable node

3.3 Collaborative Stepwise Movement

In our scheme, sensors movesto compensate the loss of coverage over the hole left by dead node as shown in Figure 5. When neighboring sensor nodes determine the s_d is dead, each neighboring sensor nodes first determines whether the to-be-dead node is *vulnerable* or not. If the to-be-dead node is not a *Vulnerable Node* it updates new perimeter coverage property and there is no movement. If to-be-dead node is a *Vulnerable Node*, it determines itself be *Hole Border* or not. In case the neighbor is not *Hole Border* then it updates new perimeter coverage property and stand still. In other case it makes Stepwise movement toward H_C by predefined Δ until one of the following conditions is reached: (1) any neighboring node becomes uncovered; (2) the *Hole* covered.

We determine the stepwise movement of sensors by defining *stepwise value* Δ that is predefined in deployment period of the sensor network as an environmental parameter. It depends on the scale of network as well as sensing range of sensors, in our simulation we choose $\Delta = r_c/10$, where r_c is sensing range of sensors. Also we determine the direction H_C of



(Fig. 5) Flowchart of vulnerability check and differential movement for coverage maintenance

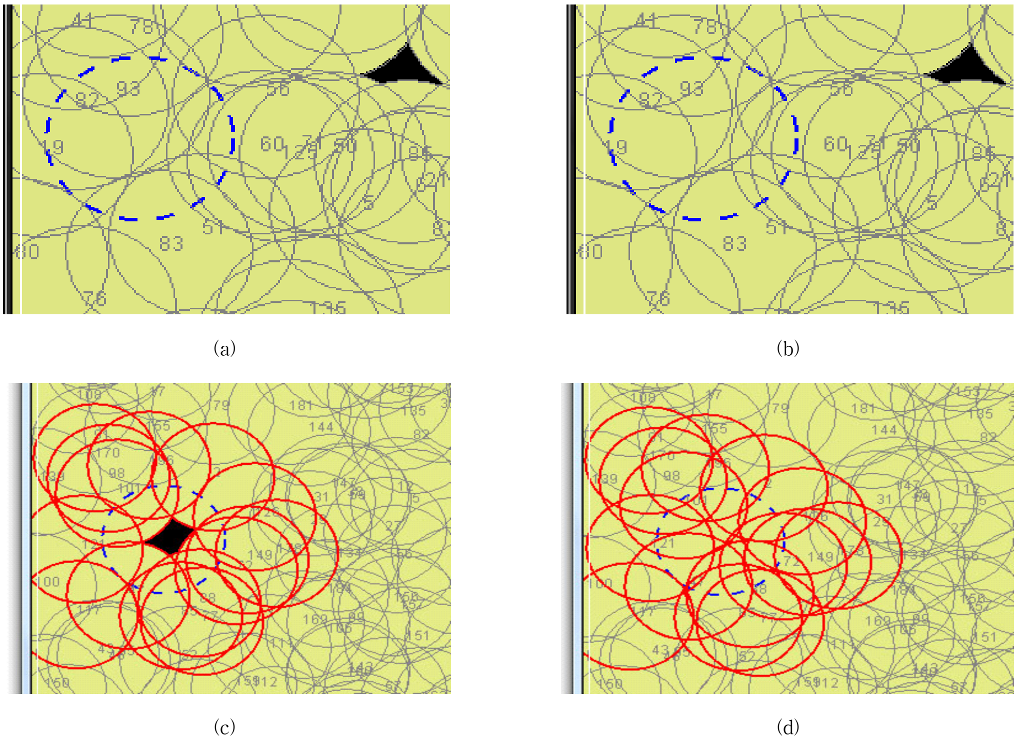
movement of neighboring sensor nodes. $H_C$is a point inside the *Hole* (uncovered network area) that minimizes the total moving distance of neighboring sensor nodes. However, it is difficult to identify the position of such point due to the various shape of the *Hole* as well as the arbitrary location of *Hole Border*s. We use a heuristic which sets H_C as the position of dead node. We use this heuristic as defaults because it is simple and easy to manipulate.

4. Experimental Results

For experimental configuration as shown in Figure 6, we simulate WSNs consisting of a large number of small, light-weight, highly power-constrained wireless nodes. Sensors are equipped with processors, memory and power sources. The area covered by a sensor is assumed to be circular with a sensing range (coverage radius) r_c. The lifetime of sensor nodes is modeled as an exponential distribution (λ is the failure rate) which is used for birth-and-death process [8]. The probability distribution function over time t is, where. To reflect the fact that a node with less remaining energy is more likely to die due to battery failure than a node with higher reserves of energy, the failure rate is inversely proportional to the available energy i.e., $\lambda \propto 1/E$, where E is the energy available at that point of time. In our simulation, we deployed numbers of sensors with sensing range of 50m over the network area of 600m × 600m. The number of sensors *n*varies from 100 to 200. The lifetime of sensor is formulated as the uniform random distribution (0, 1) [8] without loss of generality, Δ = 5m.

After doing 100 experiments for each deployment of sensor network, we obtain (1) the total distance that all related neighboring nodes moved, and (2) the area of the Hole before and after compensation process. Figure 7shows the comparison of total movement, which is a sum of distance moved by all nodes. We compared our stepwise movement scheme (default heuristics) with DCM (four heuristics) [8]. Simulation results with 150 sensor nodes show that our scheme moves less than other four heuristics, especially, our scheme moves half of the distance which MDL (the best among four DCM heuristics) moves.

To evaluate coverage maintenance and energy saving efficiency, we introduce two measures: CER (coverage efficiency ratio) and MER (movement efficiency ratio) as defined in equation (2). The former is ratio between the width of covered areas in the hole and the width of hole.

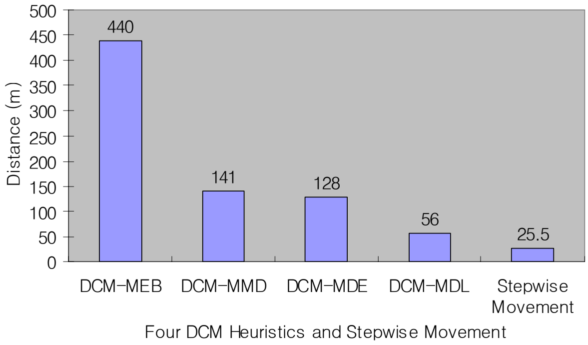


(Fig. 6) Simulation of collaborative stepwise movement: (a) and (b) - The dead node is not vulnerable node, neighbors need not move after the node dies; (c) and (d) - the dead node is vulnerable node, neighbors need to move for the lost coverage compensation

It reflects how much the uncovered area (S_U) left by dead node is covered (S_C) in percentage. The latter is ratio between square of total moving distance of sensors $(\sum d_{moving})^2$ over the area of the hole S_U.

$$CER = 100 \times \frac{S_C}{S_U}$$
$$MER = \frac{(\sum d_{moving})^2}{S_U} \quad (2)$$

<Table 1> shows the simulation result for the



(Fig. 7) Comparison of average of total moving distance between four DCM heuristics and stepwise movement (case for 150 nodes)

<Table 1> Experimental results for coverage maintenance and energy efficiency

No	Total moving Distance (m)	Hole Area S_U (m^2)	Covered S_C (m^2)	CER (%)	MER
1	60	1996	1852	92.78	1.80
2	30	1718	1718	100	0.52
3	10	15	15	100	6.67
4	15	318	318	100	0.71
5	5	1073	1073	100	0.02
6	15	421	421	100	0.53
7	10	125	125	100	0.80
8	5	1181	1181	100	0.02
9	45	229	229	100	8.84
10	20	583	583	100	0.69
11	85	1623	1623	100	4.45
12	20	3911	3911	100	0.10
13	30	420	420	100	2.14
14	75	2054	2054	100	2.74
15	15	2510	2510	100	0.09
16	10	2454	2454	100	0.04
17	10	137	115	83.94	0.73
18	15	3684	3684	100	0.06
19	15	386	386	100	0.58
20	20	2676	2676	100	0.15
	Average: 25.5		Success Probability: 90 (%)		

comparison of total moving distance, CER, and MER. Our scheme shows full coverage capability with 90% of success probability (the percentage of full coverage case over missing coverage case), which means that our scheme gives full coverage in the most cases as shown in Table 1. That high percentage of success probability can be explained as the optimized collaborative stepwise movement with all neighboring nodes involved.

5. Conclusions

In this paper, we proposed stepwise movement scheme to compensate uncovered area. Previous approach makes specific designated neighboring nodes to move and compensate the uncovered area. Our scheme makes all the neighboring nodes of the vulnerable node (i.e. dead node which makes hole) move toward the identified hole center H_C. We defined the PDU for the information exchange between mobile sensor nodes, and each sensor node can utilize the information to compute perimeter coverage property and determine whether the sensor is covered or not. Simulation results show that we get considerable performance gain in the sense that our scheme can achieve the full coverage maintenance with success probability 90% and more than 50% decrease of moving distances than DCM schemes.

References

[1] E. Ekici, Y. Gu, and D. Bozdag, “Mobility-Based Communication in Wireless Sensor Networks,” IEEE Communications Magazine, vol.44, no.6, pp.56-62, July 2006.

[2] J. Lu, J. Wang; T. Suda, “Scalable Coverage Maintenance for Dense Wireless Sensor Networks,” Sensor and Ad Hoc Communications and Networks, 2006. SECON ’06. 2006 3rd Annual IEEE Communications Society, vol.2, pp.651-660, Sept., 2006.

[3] V. Coskun, “Relocating Sensor Nodes to Maximize Cumulative Connected Coverage in Wireless Sensor Networks”, Sensors Journal, Vol.8, April 2008.

[4] Z. Butler and D. Rus, “Event-based Motion Control for Mobile Sensor Networks,” in IEEE Pervasive Computing Magazine, vol.2, pp.34-42, Oct., 2003.

[5] K. Ma, Y. Zhang, W. Trappe, “Managing the Mobility of a Mobile Sensor Network Using Network Dynamics,” IEEE Transactions on Parallel and Distributed Systems, Vol.19, Issue1, pp.106-120, Jan., 2008.

[6] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, “Robomote: A Tiny Mobile Robot Platform for Large-Scale Ad-hoc Sensor Networks,” in Proc. of ICRA2002, vol.2, pp.1143-1148, May, 2002.

[7] C.-F. Huang, Y.-C. Tseng, “The coverage problem in a wireless sensor network,” in ACM International Workshop on Wireless Sensor Networks and Applications, pp.115–121, 2003.

[8] A. Sekhar, B. S. Manoj, and C. S. R. Murthy, “Dynamic Coverage Maintenance Algorithms for Sensor Networks with Limited Mobility,” in Proc. IEEE International Conference on Pervasive Computing and Communications, pp.51-60, USA, Mar., 2005.

[9] M. Kamat, A.S. Ismail, S. Olariu, “Optimized-Hilbert for Mobility in Wireless Sensor Networks,” in IEEE International Conference on Computational Science and its Applications, ICCSA 2007, pp.554-560, 2007.

[10] G. Xing, C. Lu, R. Pless, and J. A. “O’Sullivan, Co-Grid: an efficient coverage maintenance protocol for distributed sensor networks,” in Proc. of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN ’04), pp.414-423, Berkeley, Calif, USA, April 2004.

[11] X. Li, N. Santoro and I. Stojmenovic, “Mesh-based sensor relocation for coverage maintenance in mobile sensor networks,” in Proc. Ubiquitous Intelligence and Computing, Hong Kong, China, July, 2007.

한 녹 손

e-mail : han.ngoc.son@gmail.com
2006년 Hanoi University of Technology Applied Mathematics 학사
현 재 서울시립대학교 컴퓨터통계학과 석사과정
관심분야 : WSN, 영상복원, 게임기술

김 성 환

e-mail : swkim7@uos.ac.kr
1999년 한국과학기술원 전자전산학과 박사
1996~2000년 LG전자/정보통신 이동통신 연구소 선임연구원
2000~2002년 Cisco Systems EWBU Senior Research Engineer
현 재 서울시립대학교 컴퓨터과학부 부교수
관심분야 : 컴퓨터 보안, 영상통신, 무선멀티미디어, 게임기술