A Distributed Node Scheduling Protocol Considering Sensing Coverage in Wireless Sensor Networks

Jaekyu Cho, Gilsoo Kim, Taekyoung Kwon and Yanghee Choi
School of Computer Science and Engineering
Seoul National University, Seoul, Korea
Email: {jkcho, gskim, tk, yhchoi}@mmlab.snu.ac.kr

***Abstract*— A crucial issue in deploying wireless sensor networks is to perform a sensing task in an area of interest in an energy-efficient manner since sensor nodes have limited energy power. The most practical solution to solve this problem is to use a node scheduling protocol that some sensor nodes stay active to provide sensing service, while the others are inactive for conserving their energy.**

In this paper, we present a distributed node scheduling protocol, which can maintain sensing coverage required by applications and yet increase network lifetime by turning off some redundant nodes. In order to do this, we use the concept of an *effective sensing area* (*ESA*). The *ESA* of a sensor node refers to the sensing area that is not overlapping with other sensor's sensing area. A sensor node determines whether it will be active or not after calculating its own *ESA*. The proposed protocol allows sensor nodes to sleep opportunistically while satisfying the required sensing coverage. Through extensive simulation experiments, we have observed significant improvement in terms of network lifetime, which we have compared with the existing protocols.

I. Introduction

Wireless sensor networks consist of a large number of sensor nodes which they have sensing and communication functions. Sensor nodes deployed in an area of interest gather information from the physical world, process it and deliver a report message to a remote user in a multi hop fashion. Wireless sensor networks have some unique characteristics compared to ad hoc wireless networks such as limited resource, large and dense deployment, dynamic topology, etc. Especially, sensor networks should operate in a high energy-efficient manner. Therefore, energy efficiency in the wireless sensor networks is a critical issue. There has been a number of approaches proposed to prolong the network lifetime [1] [2].

Typically, there is a substantial redundancy in terms of node density in wireless sensor networks since deploying a large number of sensor nodes is more economic than recharging batteries of sensor nodes. Multiple sensor nodes monitor the same region, generate and transmit redundant data to a user. Therefore, the wireless sensor networks can achieve reliability and availability. On the other side, there are unnecessary energy consumptions during monitoring, which generating and transferring redundant data. Hence, operating redundant sensor nodes simultaneously cannot increase the network lifetime, which motivates to employ a node scheduling protocol in which only some nodes perform the above functions while the others can sleep. In this way, the energy consumption of the network will become efficient by the node scheduling protocol, and hence the sensor network performs the sensing task for a prolonged time. However, finding the minimum subset of sensor nodes to activate while satisfying the required sensing coverage is NP-hard [3] [4]. Thus, we propose the concept of an *effective sensing area* by which a node can locally determine whether it will be activated or not.

In the literature, it is widely accepted that the network lifetime is more important than providing the perfect sensing coverage in many applications. A sensing coverage is defined as the ratio of the area covered by sensor nodes awake to the whole area of interest. For instance, a sensor network is deployed to monitor environmental variable (e.g. temperature, humidity etc.) in the area of interest, such as mountain, farmland and warehouse. In this case, a sensor network may provide the perfect sensing coverage in the whole area if all sensor nodes are active. However, the applications may require partial coverage of the area of interest. Then, it may be sufficient that the fraction of deployed sensors wakeup and report collected data to users. As a result, sensor nodes reduce energy consumption and the sensor network increases network lifetime.

In this paper, we propose a distributed node scheduling protocol that only a subset of sensor nodes is active while satisfying a required sensing coverage. The design issue in a node scheduling protocol is as follows. First, a sensor node chooses its state, either active or inactive, based on its neighbor sensor nodes' information in a distributed manner. Second, a sensor node schedules its active and inactive intervals considering its own remaining energy to achieve network-wide energy-efficiency. Finally, the scheduling protocol will satisfy the required sensing coverage.

The remainder of this paper is organized as follows. Section II reviews the related work and Section III introduces preliminaries. Section IV details the proposed protocol operations. In Section V, simulation results are given to validate our protocol. Lastly, Section VI concludes the paper.

II. Related work

In this section, we examine the current approaches to develop sensor node scheduling scheme in wireless sensor networks.

PEAS [5] is a probing based scheme to schedule sensor nodes in wireless sensor networks. PEAS selects a subset of nodes which stay in working state until they deplete their energy. Other sensor nodes fall asleep and start working if there are no working nodes within its probing range. In this scheme, the density of the working nodes can be controlled by the range of probing messages. In our experiments, probing range is set to sensing range. In PEAS, a sensing hole takes place permanently once it occurs. This is because a working node performs sensing operation until its energy depletes. Furthermore, it may cause partitioning of the network or isolation of nodes. PECAS [6] is a collaborating adaptive sleeping scheme to improve PEAS. Unlike PEAS, PECAS informs the probing node of the next sleep time of a current working sensor node in the reply message. It allows probing nodes to substitute for the current working node right after the working nodes goes to sleep to reduce the permanent sensing holes. [7] [8] focused on coverage and connectivity. [7] presented a distributed approach based on a connected dominating set (CDS). The selected set of active nodes provides full coverage and connectivity. [8] considered a trade-off between coverage and data latency. This paper presented a minimum sensor selection scheme to achieve the partial coverage, which is the application requirement. In [9] [10], the authors proposed a sensing coverage preserving scheme to reduce the overall system energy consumption by turning off redundant sensor nodes in wireless sensor networks. Sensor nodes determine periodically whether they turn on or not using local neighbor information without affecting the sensing coverage. [11] [12] solved the coverage problem that ensures every point in the area is contained within at least k sensors' sensing ranges. Then a local and simple method was proposed to construct k separate sets, each set achieving 1-coverage. Three kinds of algorithms are presented to construct k-cover set for resolving this problem. In [13], a cluster-based sensor network is considered, and the sensor nodes communicate directly with the cluster head. The authors adjusted the sleeping probability of each sensor node according to its distance and hence transmission power from the cluster header. Hence, the sensor node farthest from the cluster head is put to sleep more frequently than the rest of the nodes.

III. Preliminaries

A. Problem definition

We consider that a set of sensor nodes, N, are deployed over an area of interest, A_{int}, with a random distribution. Each sensor node i is monitoring its sensing area denoted by $S(i)$. The objective of the proposed protocol is to make the subset of N awake in order to provide the required sensing coverage and to extend the sensor network's lifetime.

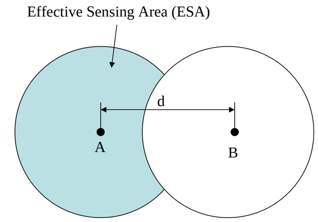

Fig. 1. Effective sensing area.

$$\frac{\bigcup_{i \in K} S(i)}{A_{int}} \geq required\ sensing\ coverage \quad (1)$$

where K is the set of active nodes.

To solve this problem, we will use the concept of an *effective sensing area* to be explained in Section III-C.

B. Assumptions

We make the following assumptions in designing the proposed protocol. According to several recent papers, sensor nodes know their geographical locations by Global Positioning System (GPS) [14] [15] or virtual coordinates [16]. In either way, we assume that a node has a means to obtain its geographical location.

We also assume that all the nodes have the same communication and sensing ranges. To ensure communication connectivity between two nodes having non-overlapping sensing ranges, the communication range, r_c, is at least twice as long as the sensing range, r_s [9]. In the following description, we assume that r_c=2r_s. For simplicity, the communication and sensing ranges are concentric with the node at the center. In the protocol description, all the nodes are deployed in a two-dimensional plane and synchronized and stationary.

How a node transfers sensory data to a user or gateway node is the issue of a routing protocol which is out of the scope of this paper.

C. Effective Sensing Area (ESA)

An *effective sensing area* of a sensor node refers to its own sensing area which is not overlapping with the sensing areas of its neighbor nodes, as illustrated in Figure 1. The grey area of node A in Figure 1 is A's *ESA* when node *B* is working at the same time.

Let S_i denote the sensing area of node i with the sensing range r_s, and $N(i)$ be the sensing neighbor set of node i defined as $N(i) = \{j \in N | d(i,j) \leq 2r_s, j \neq i\}$ where $d(i,j)$ is the distance between node i and node j. Thus the *ESA* of node i is defined as

$$ESA(i) = S(i) - (\bigcup_{j \in N(i)} S(i) \cap S(j)) \quad (2)$$

For instance, if there are two sensor nodes, *A* and *B*, as in Figure 1, node *A* can calculate its *ESA* when node *B* has the overlapping sensing range. The *ESA* of node *A* is calculated by

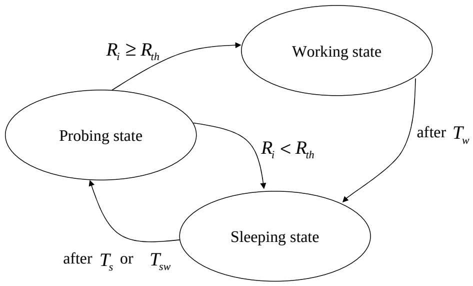

Fig. 2. State diagram.

$$ESA(A) = r_s^2 \left[\frac{d}{2r_s} \sqrt{4 - \left(\frac{d}{r_s}\right)^2} + 2\arcsin\left(\frac{d}{2r_s}\right) \right] \quad (3)$$

where d, $(0 \leq d \leq 2r_s)$, is the distance between node *A* and *B*. If there are more than one working neighbor node, the computation of the *ESA* of node *A* is more complicated [9]. We omit how to compute the *ESA* of a node when there are two or more working neighbor nodes due to the limitation of space.

In this way, we can obtain the ratio of the *ESA* to the whole sensing range of sensor node i, R_i, by

$$R_i = \frac{ESA(i)}{\pi r_s^2} \quad (4)$$

Then each node checks the following condition to decide whether working or not at a decision epoch:

$$node\ i\ is \begin{cases} working, & if\ R_i \geq R_{th}, \\ sleeping, & otherwise. \end{cases} \quad (5)$$

where R_{th} is a threshold that is determined by the required sensing coverage.

IV. ESA-based Scheduling protocol (ESP)

We now describe a distributed node scheduling protocol called ESP. ESP operates in a distributed manner in the sense that every node obtains the information of one-hop neighbor nodes. ESP is also based on the *effective sensing area* to allow each node to decide its states for prolonging network lifetime while maintaining a required sensing coverage.

There are three states (probing, working and sleeping) for each sensor node which are illustrated by Figure 2. At the beginning of the network operation, all the nodes are in the sleeping state by default. After random back-off time, node i changes its state from sleeping state to probing state.

In probing state, node i makes a decision whether it is working or not by calculating R_i. To do so, node i requires its active neighbor nodes information. That is, node i broadcasts a probing message (containing node's ID and location) in order to know the positions of the working neighbor nodes and when its working neighbor nodes will go to sleep. If there is no response for the probing message for a predefined time, T_r, node i figures out that no working nodes exist in its vicinity. Then node i transitions from probing state to working state and begins sensing operations for its working time, T_w. After T_w, node i will go to sleep state.

If there are working nodes within the communication range, node i receives reply messages from its working neighbor nodes. Each reply message contains the node ID, the node location and remaining working time. On receiving the reply messages, node i stores this information into its neighbor table. Then it calculates R_i as mentioned in Section III-C.

Node i goes to working state from probing state if calculated R_i is greater than or equal to a predefined threshold, R_{th}. Otherwise, node i will go back to sleep state. This time, node i will sleep until a timer expires after T_{sw}. T_{sw} is calculated as the time until node i's R_i becomes higher than R_{th}.

In working state, node i is responsible for sensing its vicinity and delivering sensory data to the sink. This time, node i should determine how long it is working. The energy level is considered to compute its working time. For load balancing in wireless sensor networks, a node whose remaining energy is higher than other nodes can work for a longer time. Thus we can derive the working time of node i, $T_w(i)$, from an exponential distribution by

$$T_w(i) = T_{max}[\alpha(1 - \sqrt{\frac{e - e^{(\frac{E_r(i)}{E_{ini}(i)})}}{e-1}}) + \beta] \quad (6)$$

where T_{max} is the predetermined maximum working time. $E_r(i)$ and $E_{ini}(i)$ are the amount of remaining energy and initial energy of node i, respectively. α and β are tunable system parameters $(\alpha + \beta = 1)$. After node i is working for $T_w(i)$, it will go to sleep state.

When node i enters into sleeping state, there are two cases; (1) node i should turn itself off and stay in sleeping state for T_s which is the time predefined by system if node i transits its state to sleeping state from working state. (2) node i stay in sleep state for T_{sw} mentioned above if node i transits its state to sleeping state from probing state.

V. Simulation Results

To verify the effectiveness of ESP, extensive simulation experiments are performed. In this paper, our experiments focus on the comparison of ESP, PEAS and PECAS in terms of the network lifetime and the number of alive nodes. We evaluate ESP in different environments varying the node density and the sensing range. We set our simulation parameters as shown in Table I. For each data point shown, we have calculated the average of 10 runs with different random seeds. Sensor nodes are randomly deployed over an area of size 200×200 m^2. For the energy model, we assume that the energy consumed by a node is proportional to the length of time over which the node is working. That is, a working node consumes 1 unit power per time unit in our simulation while a sleeping node does not consume energy. For simplicity, we do not consider packet flows and hence the energy consumption for communication and processing is disregarded.

TABLE I
SIMULATION PARAMETERS

Parameter	Values
Two-dimensional area	200 × 200 m^2
Number of sensor nodes	200 and 400
Sensing range (r_s)	20 and 30 m
Initial energy of each sensor node	10 unit power

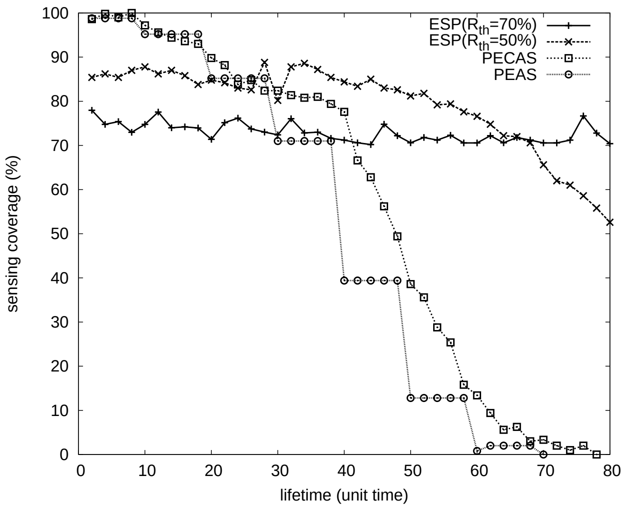

Fig. 3. Sensing coverage of three scheduling protocols over time (r=20m, 200 nodes).

In Figures 3 and 4, we compare ESP with the existing scheduling protocols, PEAS and PECAS in terms of the network lifetime and the number of alive nodes. An alive node is a node which has a remaining energy and is in one of three states (working, sleeping and probing). In PEAS, more sensors are in working state but a large part of their sensing area is redundantly overlapped by their neighbors' sensing area. Initially, the sensing coverage is better than ESP since more sensors maintain working state until their energy is depleted. However, the sensing coverage is not sufficient over time due to the fact that sensors die rapidly. In PECAS which is an advanced version of the PEAS in terms of energy balance, more sensors also maintain working state in its early stages. PECAS has a slightly longer lifetime than PEAS; nevertheless its sensing coverage is similar to that of PEAS with time. We observe that ESP performs better than PEAS and PECAS in terms of sensing coverage and number of alive nodes. For instance, the network lifetime is longer when the threshold (R_{th}) is 70%. Although the sensing coverage of ESP is poorer (about 75%) than those of PEAS and PECAS in early stages, its coverage is satisfactory to some application requirements. In addition, ESP provides a long network lifetime. Sensor nodes die more slowly since nodes determine their states according to their neighbors' states.

We observe that the sensing hole does not take place continuously at the same point in ESP as illustrated in Figure 5. At time t_0, some area are not covered by working nodes (marked by the crosses on the left side in Figure 5). However,

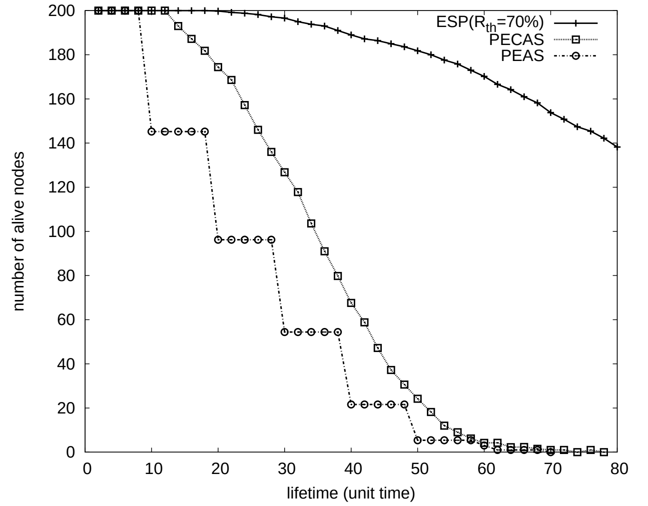

Fig. 4. Number of alive nodes of three scheduling protocols over time (r=20m, 200 nodes).

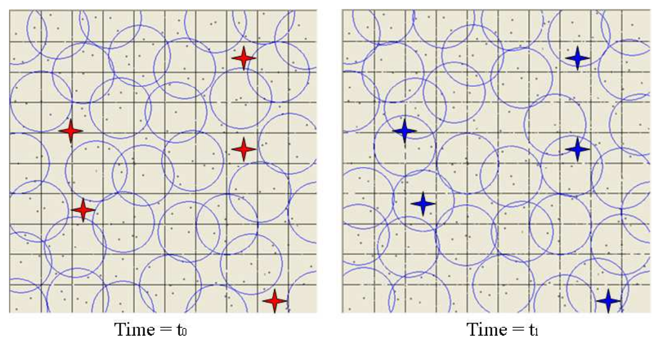

Fig. 5. Sensing hole by time flow (r=20m, 200 nodes).

these places, not covered at time t_0, are covered at time t_1 (marked by the crosses on the right side in Figure 5). Thus, sensing hole at a moment will be covered after a while.

We also measure the network lifetime for denser sensor networks. We vary the number of sensors from 200 to 400 to obtain better sensing coverage for a long time. Although the number of deployed nodes is doubled, Figure 6 shows that both ESP and PECAS cannot guarantee two times network lifetime. ESP performs better than PECAS for both scarce and dense scale sensor networks.

In Figure 7, we change the sensing range from 20m to 30m. We observe that the sensing range does not significantly affect the sensing coverage and the network lifetime. When the sensing range is increased, the sensing coverage is decreased a little and the network lifetime is not prolonged.

VI. CONCLUSION

Wireless sensor networks are extremely resource-constrained networks. It is important to conserve energy for prolonging the network lifetime. In order to extend the network lifetime, a proper number of the sensor nodes is working in turns while the others are sleeping. In this paper, we have proposed the distributed node scheduling protocol

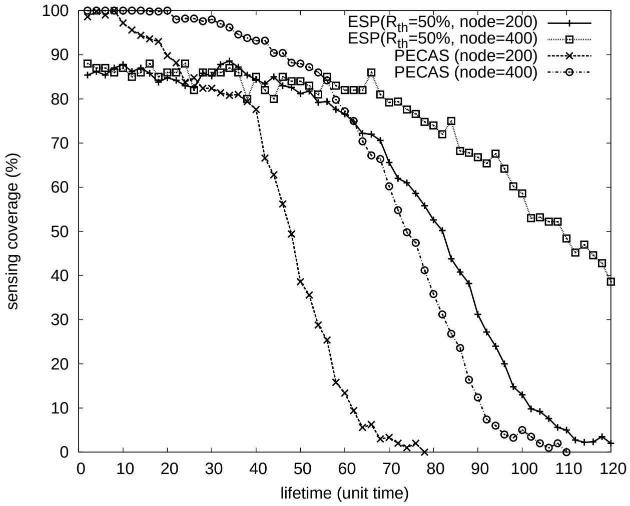

Fig. 6. Coverage over time with varying node density (r=20m).

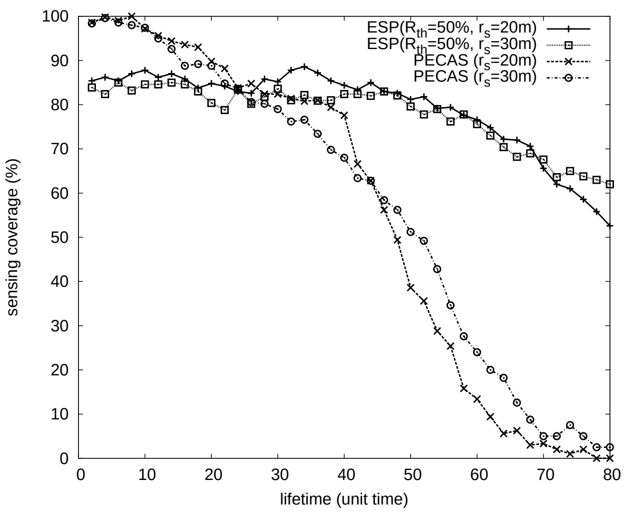

Fig. 7. Coverage over time with varying sensing range (200 nodes).

to prolong network lifetime while maintaining the sensing coverage required by applications. This protocol utilizes the concept of an *effective sensing area* to decide whether nodes are working or not. In addition, the protocol takes advantage of remaining energy in working state to achieve load balancing. Thus, the network lifetime can be significantly increased due to the fact that the number of working nodes is reduced. Simulation results show that the proposed protocol achieves better performance in comparison with PEAS and PECAS in terms of the network lifetime and the number of alive nodes under comprehensive environments. Our future research work will include performance analysis.

ACKNOWLEDGMENT

This research is supported by the Ubiquitous Computing and Network (UCN) Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

REFERENCES

[1] M. Cardei, M. Thai, Y. Li, W. Wu, " Energy-efficiency target coverage in wireless sensor networks," In proc. IEEE Infocom 2005, Mar. 2005.

[2] I. Howitt, J. Wang, "Energy balanced chain in distributed sensor networks," In proc. IEEE WCNC 2004, Mar. 2004.

[3] V. Bharghavan and B. Das, "Routing in Ad Hoc Networks Using Minimum Connected Dominating Sets," In proc. IEEE Int'l Conf. Comm, 1997.

[4] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets," Algorithmica, vol. 20, no. 4, 1998.

[5] Fan Ye, Gary Zhong, Songwu Lu, Lixia Zhang, "A Robust Energy Conserving Protocol for Long-lived Sensor Networks," In proc. IEEE ICDCS, 2003.

[6] Chao Gui and Prasant Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks," In proc. Mobicom '04, Sep. 2004.

[7] Y. Zou, K. Chakrabarty, "A distributed coverage-and connectivity-centric technique for selecting active nodes in wireless sensor networks," IEEE Transactions on Computers, Aug. 2005.

[8] W. Choi and S. K. Das, "Coverage-Adaptive Random Sensor Scheduling for Application-Aware Data Gathering in Wireless Sensor Networks," Computer Communications, 2006.

[9] D. Tian and N. D. Georganas, "A Node Scheduling Scheme for Energy Conservation in Large Wireless Sensor Networks," Wireless Communications and Mobile Computing Journal, May. 2003.

[10] D. Tian and N. D. Georganas, "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks," In proc. ACM Workshop on Wireless Sensor Networks and Applications, Oct.2002.

[11] Z. Abrams, A. Goel, and S. Plotkin, "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks," In proc. Information Processing in Sensor Networks, 2004.

[12] C. F. Huang and Y. C. Tseng, "The Coverage Problem in a Wireless Sensor Network," In proc. WSNA '03, 2003.

[13] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney, "Balanced-energy sleep scheduling scheme for high density cluster-based sensor networks," Computer Communications: special issue on ASWN04, Sep. 2005.

[14] Y. Yu, R. Govindan and D. Estrin, "Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks," UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023, May. 2001.

[15] M. Heissenbuttel, T. Braun, M, Walchli and T. Bernoulli, "Optimized Stateless Broadcasting in Wireless Multi-hop Networks," In proc. Infocom 2006, Apr. 2006.

[16] T. Moscibroda, R. O'Dell, M. Wattenhofer and R. Wattenhofer, "Virtual Coordinates for Ad hoc and Sensor Networks," In proc. DIALM-POMC 2004, Oct. 2004.