Probabilistic Self-Scheduling for Coverage Configuration in Wireless Ad-hoc Sensor Networks

Jun Lu, Lichun Bao and Tatsuya Suda
School of Information and Computer Sciences
University of California, Irvine, CA 92697, USA
{lujun, lbao, suda}@ics.uci.edu

Abstract

Sensing coverage is a critical issue in sensor network deployments. We propose a novel scheme to maintain the sensing coverage in sensor networks, which we call *CASS (Coverage-Aware Self-Scheduling)*. Different from the existing work on coverage maintenance, CASS probabilistically schedules sensing activities according to the sensor's contribution to the sensing coverage of the whole sensor network. CASS is designed to allow sensors with higher coverage contribution to have more chance to sense. In this way, CASS reduces the number of active sensors to maintain certain coverage. Besides the sensing coverage, the connectivity of the network topologies is required for the purpose of communicating among sensors to collect sensing data. Therefore, we describe a generic unifying framework to incorporate different connectivity and coverage maintenance schemes. Simulations are carried out under the framework by integrating CASS with an existing connectivity maintenance scheme - LEACH (the Low Energy Adaptive Clustering Hierarchy). Simulation results show that CASS can considerably improve the energy efficiency of sensing coverage with low communication and computation overhead.

Index Terms

sensing coverage, sensor scheduling, self-organization, wireless sensor networks

I. INTRODUCTION

Wireless sensor networks have emerged rapidly to provide surveillance functions in a variety of applications, *e.g.*, environment monitoring and target tracking. In wireless sensor networks, a large number of small wireless sensors are employed to monitor a target area and report sensing data through wireless communications.

A previous version of this article was published at IEEE Computer Communication Workshop (CCW) 2003 and later at International Conference on Sensing Technology (ICST) 2005.

Due to their limited size, wireless sensors are highly restricted in capabilities (*i.e.*, the sensing, computation and communication abilities) and constrained in resources (*i.e.*, the storage and power capacities). For example, the crossbow mica mote MPR300CB [1] has a low-speed $4MHz$ processor equipped with only $128kB$ flash, $4kB$ SRAM and $4kB$ EEPROM. It has a maximal data rate of only $40kbps$ and a transmission range of about 100 feet, using the power of two AA batteries. Due to the limited capabilities and constrained resources of wireless sensors, wireless sensor networks are usually deployed in high density, which not only helps to improve their reliability, but also extends their longevity. Moreover, wireless sensor networks are usually employed to monitor a large area, which makes manual deployment infeasible. In this research, we assume a large-scale wireless sensor network deployed in an ad-hoc manner.

Given such a randomly and densely deployed wireless sensor network, it is desirable to have sensors autonomously schedule their duty cycles according to local information of connectivity and coverage. This is regarded as a self-organization problem, which has drawn intense research attention recently. As Xing *et al.* [2] pointed out, self-organization to maintain network connectivity and sensing coverage are two different issues and both are essential for wireless sensor networks.

Extensive work has been done on the connectivity maintenance issue. For example, Heinzelman [3] proposed LEACH as a dynamic clustering algorithm for wireless sensor networks. Sensors elect themselves as cluster heads with certain predefined probability. Other sensors choose the closest cluster head and become cluster members. Every cluster member periodically sends data to its cluster head, which relays data to a base station after aggregating all the data received. The research in [4] focuses on energy conservation by controlling transmission power while maintaining global network connectivity. It demonstrated that global network connectivity can be maintained if each node has at least one neighbor in every cone of $2\pi/3$. In ASCENT [5], sensors measure local connectivity and decide whether to join the routing infrastructure based on application requirements. Xu *et al.* [6] proposed two algorithms to identify and turn off redundant nodes while preserving connectivity.

The other issue, coverage maintenance, has also driven lots of research efforts. For example, Tian *et al.* [7] presented a node-scheduling algorithm to turn off redundant sensors if their sensing areas are covered by their neighbors. Randomized as well as coordinated sleep algorithms were proposed in [8] to maintain network coverage using low duty-cycle sensors. The randomized algorithm enables each sensor to independently sleep under a certain probability, while the coordinated sleep algorithm allows a sensor to enter sleep state if its sensing area is fully contained by the union set of its neighbors. A K-coverage maintenance algorithm was proposed in [9] so that each location of the sensing area is covered by at least K sensors. A sensor decides whether it is redundant by merely checking the coverage state of its sensing perimeter. In [10], the redundancy of the sensing coverage is analyzed, and the relation between the number of neighbors and the coverage redundancy is studied. Co-Grid proposed in [11] schedules sensors by adopting a distributed detection model based on data fusion. Abrams *et al.* studied a variant of the NP-hard SET K-COVER problem in [12], partitioning the sensors into K covers such that as many areas are monitored as frequently as possible. Yan *et al.* [13] proposed an adaptable energy-efficient sensing coverage protocol, in which each sensor broadcasts a random time reference point, and decides its duty schedule according to the time reference points received from neighbors. Xing *et al.* [2] studied the relationship between coverage and connectivity, and proposed a coverage maintenance scheme, Coverage Configuration Protocol (CCP), which, when integrated with an existing connectivity maintenance scheme, is able to provide both coverage and connectivity guarantees. Zhang and Huo [14] presented a scheme to optimize coverage maintenance while providing global connectivity by keeping a minimum number of active sensors to minimize coverage redundancy.

We propose a new coverage maintenance scheme called CASS (Coverage-Aware Self-Scheduling). Different from the existing work, CASS analyzes the probabilistic sensing behaviors of sensors caused by signal fading and background noise to schedule sensing activities, therefore conserving the energy consumed for sensing, computing and communicating data. Instead of providing strict guarantee of network coverage (*e.g.*, every location is covered by at least K sensors, namely K-coverage), CASS takes a probabilistic approach, in which sensors autonomously adjust their probability of turning off sensing units according to their local coverage information. Meanwhile, in order to maintain the network connectivity, a unifying framework is proposed to seamlessly incorporate the operations of connectivity and coverage maintenance. The framework allows any connectivity and coverage maintenance schemes to coexist and work independently, so that different requirements for connectivity and coverage can be easily satisfied. Under the unifying framework, CASS is integrated with an existing connectivity maintenance scheme called Low Energy Adaptive Clustering Hierarchy (LEACH) [3]. Using the unifying framework, it is possible to investigate CASS with other connectivity maintenance schemes and the choice of different connectivity schemes shall have little effect on the performance of CASS.

The rest of this paper is organized as follows. Assumptions are described in Section II. Section III specifies CASS in details. In Section IV, we describe the framework for connectivity and coverage maintenance. Simulation results are presented in section V for performance evaluations. Section VI concludes the paper.

II. ASSUMPTIONS

We assume that sensors are static and each sensor knows its own location. Sensors can acquire the location of neighbors through one-hop communication. Such assumptions are conveniently taken by other work (*e.g.*, [7], [9] and [13]) and are supported by the existing research on location services (*e.g.*, [15], [16] and [17]). The location can be absolute or relative to neighbors.

We assume that sensors can *separately* control the state of RF and sensing units, *i.e.*, RF unit state is independent from sensing unit state.

Due to the signal attenuation and noise, a sensor's measurement can be modeled by a probability density function, which varies with the type of signals and the propagation channel. A sensor detects an event if the sensor measurement is above a preset threshold. Thus, the sensing ability of a sensor can be modeled as the probability of a successful detection of the event of interests. Apparently, a sensor's sensing ability is a function of the distance between the sensor and the event (a similar concept of *sensor field intensity* is presented in [18]). Compared with the boolean sensing model (*i.e.*, sensors can only detect an event happening within a certain range) assumed by the existing work (*e.g.*, [7], [9], [13], and [14]), the probabilistic sensing model better reflects a sensor's sensing behavior. We assume that the probabilistic sensing model is given before deployment through calibration process.

We use $S_j(P_i)$ to describe sensor j's sensing ability at location P_i. A sensor's sensing range, denoted by SR, is defined as the range, beyond which the sensor's sensing ability can be neglected. For clarity of algorithm discussion, we further assume that sensors' communication range, denoted by CR, is at least twice the sensing range. This assumption is usually true for real sensors. For example, ultrasonic sensors have a sensing range of approximately $0.2 - 6m$ [19] while the transmission range of MICA motes is about 30 meters [1]. In the case that CR is less than twice SR, CASS can work by propagating control beacons through multiple hops.

III. COVERAGE-AWARE SELF-SCHEDULING

A. Sensing coverage metrics

Based on our assumptions, sensing ability of sensor j at location P_i represents the probability of sensor j to detect an event at P_i. We derive the sensing coverage $C(P_i)$ of a sensor network $\mathcal{A}$ at location P_i as the probability of detecting an event by any of the sensors in the network, or

$$C(P_i) = 1 - \prod_{j \in \mathcal{A}} (1 - S_j(P_i)) \tag{1}$$

We propose two metrics to evaluate sensing coverage over a target area. The first metric, *gross coverage*, is defined as the summation of $C(P_i)$ over target area $\mathcal{R}$, or

$$\begin{aligned} GC &= \sum_{P_i \in \mathcal{R}} C(P_i) \\ &= \sum_{P_i \in \mathcal{R}} (1 - \prod_{j \in \mathcal{A}} (1 - S_j(P_i))) \end{aligned} \tag{2}$$

The second metric, *coverage extensity*, is defined as the probability that $C(P_i)$ is larger than a certain threshold τ, or

$$CE = \Pr(C(P_i) > \tau \mid \forall\, P_i \in \mathcal{R}) \tag{3}$$

When τ is set to 0, coverage extensity represents the percentage of covered area.

Gross coverage and coverage extensity reflect two different aspects of sensing coverage quality: gross coverage gives the overall intensity of coverage over the entire area while coverage extensity represents the distribution of coverage at different places.

B. Coverage Contribution

We use the loss of gross coverage when the sensor is removed from the network to evaluate a sensor's contribution to sensing coverage. Sensor m's coverage contribution is calculated by

$$\begin{aligned} CC_m &= GC - GC' \\ &= \sum_{P_i \in \mathcal{R}} (1 - \prod_{j \in \mathcal{A}} (1 - S_j(P_i))) - \sum_{P_i \in \mathcal{R}} (1 - \prod_{\substack{j \in \mathcal{A} \\ j \neq m}} (1 - S_j(P_i))) \\ &= \sum_{P_i \in \mathcal{R}} (S_m(P_i) \prod_{\substack{j \in \mathcal{A} \\ j \neq m}} (1 - S_j(P_i))) \end{aligned} \tag{4}$$

where GC and GC' are the gross coverage with and without sensor m, respectively.

Since the existence of a sensor only monitors the area covered by the sensor, its coverage contribution can be calculated by only considering the area within its SR. In addition, it is obvious that a sensor's coverage contribution is affected by the neighboring sensors within $2 \cdot SR$ only. For computation simplicity, CC_m is calculated in polar coordinates:

$$\begin{aligned} CC_m &= GC - GC' \\ &= \int_0^{2\pi} \int_0^{SR} S_m(\theta, r) \prod_{j \in \mathcal{N}} (1 - S_j(\theta, r)) r d\theta dr \end{aligned} \tag{5}$$

where location P_i is denoted by (θ, r) and $\mathcal{N}$ is the set of neighbors within $2 \cdot SR$ of sensor m.

According to Eq. (5), the coverage contribution of a sensor can be calculated given the sensing ability model and the location of the active neighbors within $2 \cdot SR$.

C. The Self-Scheduling Algorithm

CASS is designed to maximize gross coverage per unit of consumed energy while fulfilling the requirement for coverage extensity. To achieve this goal, each sensor computes coverage contribution in a distributed manner and chooses the sensing state in a probabilistic way. Sensors with higher coverage contribution are more preferred to be active so that higher gross coverage can be achieved with the same number of active sensors. The details of CASS are described in the rest of the section.

Initially, sensors acquire neighbor information by exchanging IDs and location among neighbors within CR, which is approximated by $2 \cdot SR$. The neighbor information is stored in an active neighbor list. Afterward, each sensor sets a back-off timer to make its decision. When a sensor times out, the sensor computes coverage contribution according to Eq. (5) using the active neighbor list. The coverage contribution is used to compute the probability to turn off the sensing unit,

$$P_m = \frac{(CC_{base} - CC_m)}{CC_{base}} \tag{6}$$

where CC_m is the coverage contribution of sensor m, and CC_{base} is a system parameter defined by

$$CC_{base} = \varepsilon \cdot \int_0^{2\pi}\int_0^{SR} S_m(\theta, r) r d\theta dr \tag{7}$$

where ε ($\varepsilon > 0$) is a tunable parameter and the double integral yields the maximum possible coverage contribution when sensor m has no neighbor within $2 \cdot SR$. Intuitively, when a sensor has the maximal coverage contribution, which implies it has no active neighbor, it should always stay active to provide coverage. Thus, ε should be less or equal to 1. With a larger ε, CC_{base} is larger and a sensor becomes more aggressive to turn off the sensing unit.

With probability P_m, the sensor turns off the sensing unit and broadcasts a `SENSINGOFF` message to its neighbors. When a neighbor receives a `SENSINGOFF` message before its timer expires, it removes the sender of the message from the active neighbor list.

According to Eq. (6), sensors with relatively high coverage contribution have more chance to keep their sensing units active. Note that when CC_m is larger than CC_{base}, P_m is negative, which means sensor m has a negative probability to turn off its sensing unit and should keep the sensing unit active.

The CASS is again specified with pseudo codes as Alg. III.1.

IV. THE CONNECTIVITY AND COVERAGE MAINTENANCE FRAMEWORK

The self-organization in sensor networks involves two distinct issues, the connectivity maintenance and the coverage maintenance. In a real sensor network deployment, however, an application may have different requirements on coverage and connectivity. For instances, an application may require low-quality monitoring with high-bandwidth data transmission, or high-quality monitoring with low-bandwidth data transmission. Furthermore, sensors may have various configurations of sensing and communication capabilities. The above observations imply that the necessary sensor densities to fulfill the coverage and connectivity requirements are usually different. Thus,

```
Algorithm III.1: CASS()

 comment: $\mathcal{N}$ is the set of active neighbors within $2 \cdot SR$
 global $State, \mathcal{N}$
 $State \leftarrow active$
 $\mathcal{N} \leftarrow neighbors\ within\ 2 \cdot SR$

 procedure CoverageContribution()
   comment: To calculate coverage contribution
   $CC \leftarrow 0$
   comment: $S(\theta, r)$ is the sensing ability of the current sensor at $(\theta, r)$
   comment: $S_j(\theta, r)$ is the sensing ability of neighbor $j$ at $(\theta, r)$
   $CC \leftarrow \int_0^{2\pi}\int_0^{SR} S(\theta, r) \prod_{j \in \mathcal{N}} (1 - S_j(\theta, r)) r d\theta dr$
   return $(CC)$

 procedure TimeOut()
   comment: The timer expires
   $CC \leftarrow$ CoverageContribution()
   $P \leftarrow \frac{(CC_{base} - CC)}{CC_{base}}$
   if Random.Uniform$(0, 1) < P$
     then { BroadcastSENSINGOFF()
            $state \leftarrow inactive$

 procedure BroadcastSENSINGOFF()
   comment: Broadcast SENSINGOFF message to neighbors within $2 \cdot SR$
   $Broadcast$ SENSINGOFF $message$

 procedure RecvSENSINGOFF($i$)
   comment: Received a SENSINGOFF message from neighbor $i$
   $\mathcal{N} \leftarrow \mathcal{N} - i$

 main
   comment: Each sensor sets a random timer $T$ to make decision
   $T$.Random.Uniform()
```

we propose to separate the control of RF units from the management of sensing units. Under our framework, the connectivity maintenance protocol decides the active/inactive state of RF units and the coverage maintenance protocol determines the active/inactive state of sensing units. Jointly, there are four possible sensor states:

1) Relaying state. A relaying node keeps its RF unit active so that it can relay data from other nodes to maintain global connectivity. Sensing units of relaying nodes are off, which means relaying nodes do not

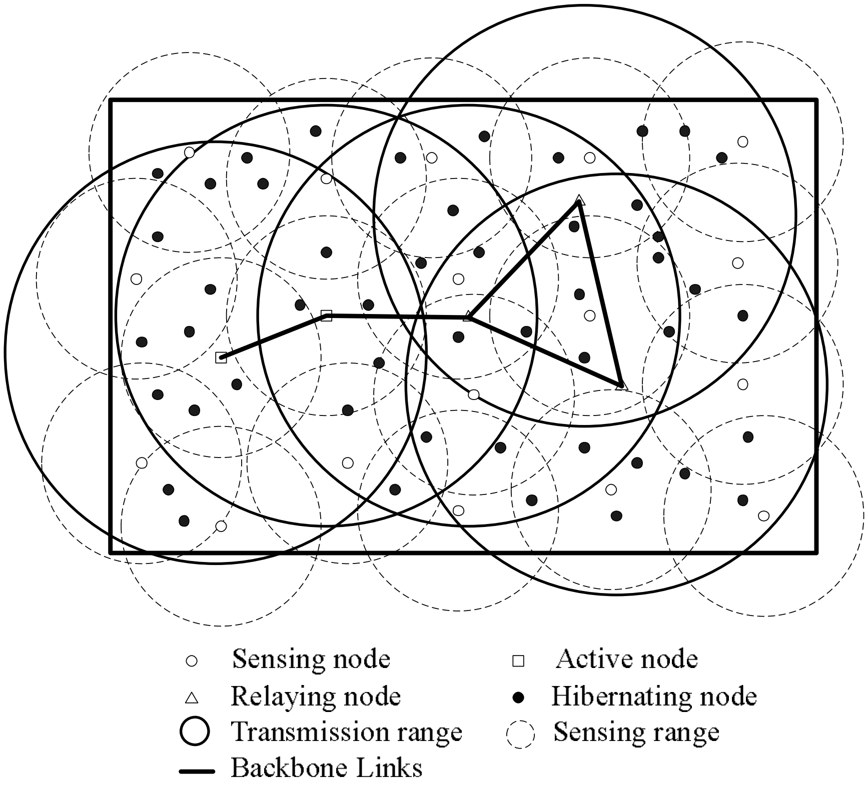

Fig. 1. An example network

generate data.

2) Sensing state. A sensing node keeps its sensing unit active in order to provide sensing coverage. The RF unit of a sensing node is usually off but is activated when the node has sensing data to transmit. When data collection nodes are not within communication range, sensing nodes send the data to nearby relaying nodes. Since RF units are off most of the time, sensing nodes cannot relay data.
3) Active state. Active nodes perform both the sensing and relaying functions.
4) Hibernating state. Both the RF unit and sensing unit of a hibernating node are off. Hibernating nodes perform no function of sensing or relaying.

Fig. 1 illustrates an example network under the proposed framework to provide 1-connectivity (*i.e.*, any sensor is in the communication range of a relaying or active node) and 1-coverage (*i.e.*, any location is in the sensing range of a sensing or active sensor). We can see that three relaying nodes (denoted by triangles) and two active nodes (denoted by squares) are enough to provide a backbone for 1-connectivity, while many more sensing nodes (denoted by blank circles) are needed to support 1-coverage.

The advantages of the framework are: 1) Sensors only keep their RF units active when they have data to send or need to relay data from other sensors, and only turn on their sensing units in order to fulfill the coverage requirements. Thus, only necessary RF units and sensing units are kept active and energy is conserved; 2) Under the framework, any connectivity and coverage maintenance schemes can coexist and work independently. Therefore, different requirements on connectivity and coverage can be easily satisfied.

V. SIMULATION EVALUATIONS

To verify the validation of the framework proposed in Section IV, we integrate CASS with LEACH [3] and evaluate their performance through simulation experiments.

Note that we do not compare CASS with other existing work on coverage maintenance (*e.g.*, [7], [9] and [13]) because they all assume the boolean sensing model and do not work under the probabilistic sensing model. Specifically, the working schedule setup algorithm in [13] and the eligibility rules of sensor redundancy described in [7] and [9] are tightly coupled with the boolean sensing model, and thus cannot be applied to the probabilistic sensing model.

A. Experiment Setup

The probabilistic sensing models depend on the sensor capabilities and environments. Although CASS shall work with any realistic sensing model, for simplicity, we assume a virtual probabilistic sensing model, two examples of which are shown below,

$$S_j(P_i) = f(D_{ij}) = \frac{1}{1 + \alpha D_{ij} + \beta D_{ij}^2 + \cdots + \gamma D_{ij}^k}$$

$$S_j(P_i) = f(D_{ij}) = \frac{1}{\chi^{D_{ij}}}$$

where D_{ij} is the distance between sensor j and location P_i, and α, β, γ and χ ($\chi > 1$) are system parameters reflecting the physical characteristics of sensor j and deployment environments.

Specifically, we assume the following virtual probabilistic sensing model in the simulations:

$$S_j(P_i) = \frac{1}{(1 + \alpha \cdot D_{ij})^{\beta}} \tag{8}$$

If not explicitly specified, α and β are default set to 0.1 and 3, respectively. We regard the sensing ability less than 6% as negligible and set the SR of the default model to 15 meters. In the simulations, ε is set to 0.75 by default and CC_{base} is calculated according to Eq. (7).

According to [20], executing one instruction consumes about $1pJ$ (pico-Joule) of energy. Assuming that the calculation of coverage contribution needs about 100,000 instructions, we set the energy consumption of coverage contribution computation to $100nJ$ (nano-Joule). We assume thermal sensors in the simulations and each thermal sampling of 10 bits costs $4nJ$ [20]. The sampling rate is set to $25Hz$ and the initial energy of each sensor is set to $2J$.

In the simulations, we integrate LEACH with CASS, which is referred as *LEACH-CC*. In LEACH, time is slotted into rounds. At the beginning of each round, sensors probabilistically choose to become cluster heads. Sensors that do not become cluster heads join the closest cluster head as cluster members. Within each round, every cluster member periodically sends data to its cluster head, which relays data to a base station after aggregating all the data received from cluster members. In LEACH-CC, each sensor executes the CASS algorithm right before the cluster formation phase in each round. Following the unifying framework, RF units and sensing units are controlled by LEACH and CASS, respectively. As described in Section IV, a sensor could be in one of the four possible states (*i.e.*, relaying, sensing, active and hibernating), which is jointly decided by the states of the RF and sensing units.

In the simulations, sensors are randomly deployed over a square area from (0, 0) to (100, 100). We ran the simulations in three different network densities, *i.e.*, 150-node, 200-node and 250-node. In each network density, 5 scenarios are randomly generated, and simulations are run 10 times per scenario. The simulation results show the average performance in these 50 runs.

B. Result Analysis

We evaluate the performance of CASS by comparing LEACH with LEACH-CC in terms of sensing coverage, communication and computation overhead, and robustness against location error.

1) Sensing coverage: Two metrics are collected in all the scenarios for the sensing coverage statistics - the gross coverage and the coverage extensity. Before forwarding the data to the base station, LEACH aggregates

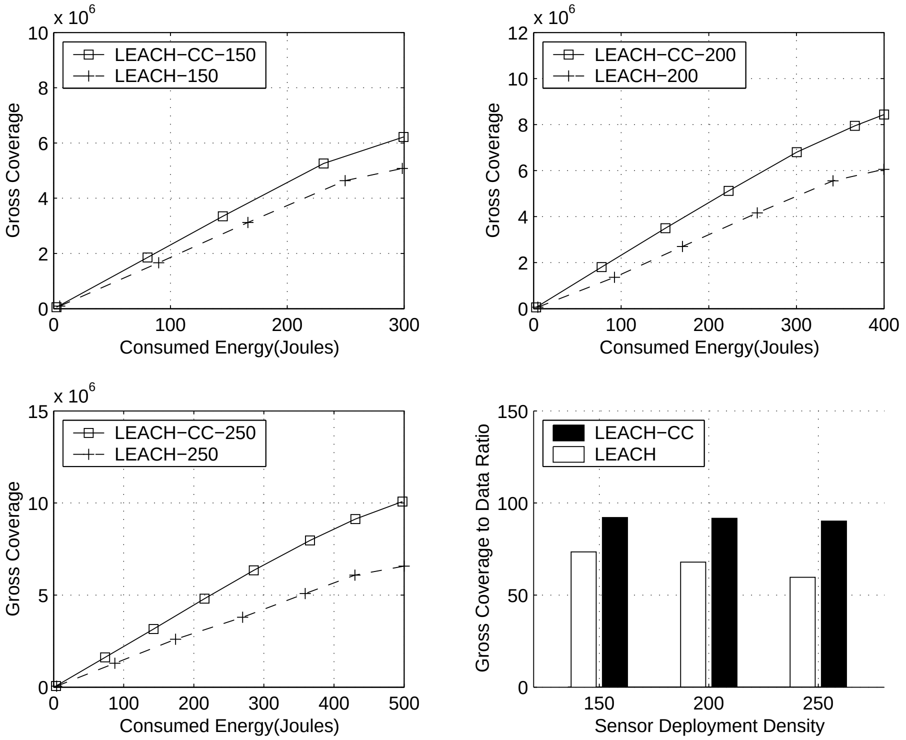

Fig. 2. Gross coverage - The first three diagrams correspond to three different network densities, 150-node, 200-node and 250-node, respectively. The last one shows the gross coverage to data ratio.

data at cluster head nodes. For each aggregated data packet received, the data collection node calculates the gross coverage of the network that consists of the sensors contributing to the aggregated packet using Eq. (2). The gross coverage is accumulated for each aggregated data packet received.

In Fig. 2, the gross coverage of LEACH and LEACH-CC is compared. The first three diagrams show the gross coverage achieved by LEACH and LEACH-CC in different network densities (*i.e.*, 150-node, 200-node and 250-node). We can see that LEACH-CC outperforms LEACH in all the cases. In LEACH-CC, sensors adjust their sensing behaviors according to their local coverage information (*i.e.*, coverage contribution). Thus, LEACH-CC is able to reduce the redundancy among data packets from different sensors and improve the gross coverage achieved per unit of energy consumed. This is further confirmed by the fourth diagram of Fig. 2, which shows the ratio of gross coverage to the number of data packets. We can see that the ratio of LEACH decreases with the increment of network density, while the ratio of LEACH-CC is almost stable because sensors become more aggressive to turn off sensing units due to the lower coverage contribution in a higher network density. As a

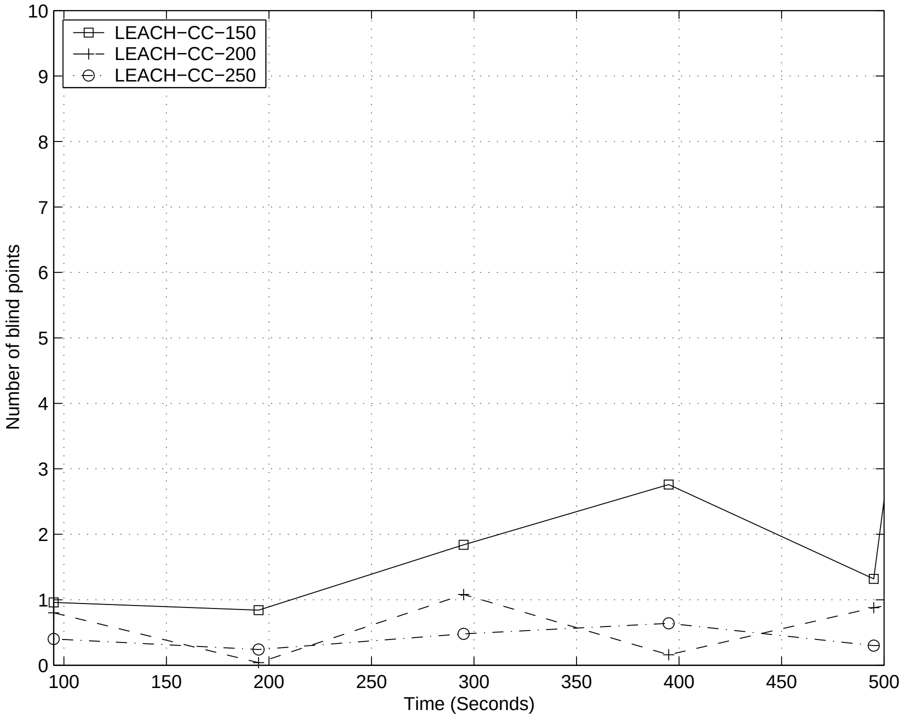

Fig. 3. Coverage extensity - number of uncovered points (out of 10,000 sampling points)

result, LEACH-CC achieves about 22% more gross coverage than LEACH in 150-node networks, while about 53% improvement is obtained for 250-node networks. We have carried out simulations with higher network densities, and similar results were obtained, *e.g.*, about 100% improvement in 450-node networks.

The coverage extensity is measured by setting the network sensing coverage threshold τ to 0. For presentation convenience, Fig. 3 shows the number of uncovered points out of $10,000$ sampling points in the field. We can see that LEACH-CC has less than 3 uncovered points, or better than 99.97% coverage extensity in all the cases. By powering off more sensing units, the spatial correlation of data packets decreases (*i.e.*, sensors have less probability to detect the same event), thus better gross coverage performance can be expected. However, coverage extensity becomes worse since more uncovered sub-area may be exposed by turning off more sensing units. Thus, there is a tradeoff between the gross coverage and coverage extensity.

The tradeoff is demonstrated by Fig. 4. From the upper diagram, the achieved gross coverage increases with the increase of ε. This is because sensors become more aggressive to turn off sensing units with a larger CC_{base}, which reduces the redundancy of data. In the lower diagram, however, more blind points are observed with the

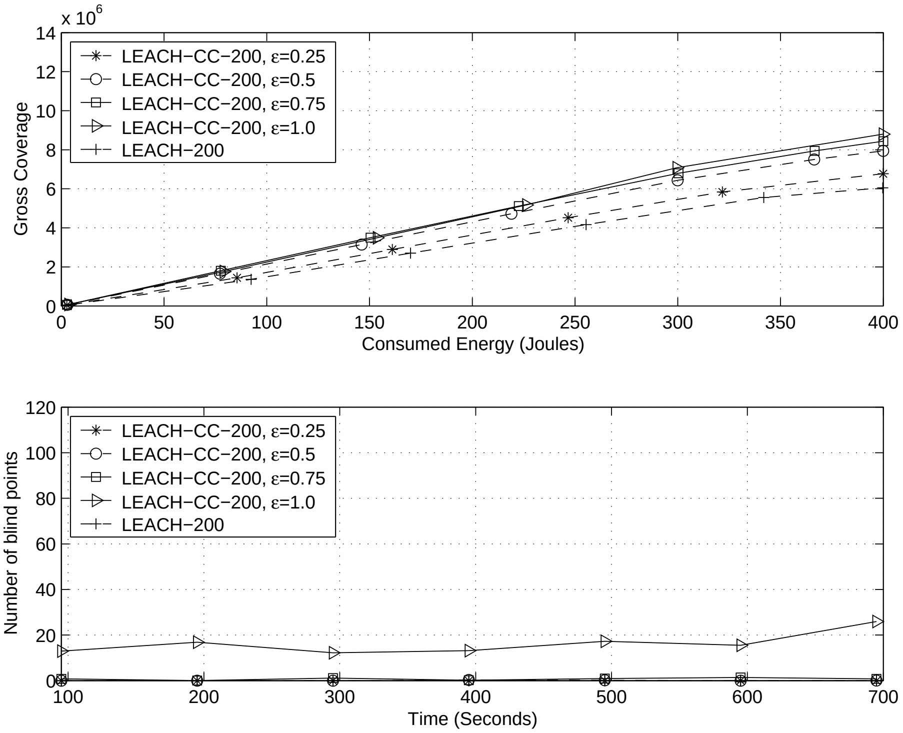

Fig. 4. The effect of CC_{base} (200 nodes)

increase of ε. Thus, it is very important to properly choose the value for ε in order to maximize gross coverage achieved while fulfilling the requirement for coverage extensity. Although not studied in the current stage of this research, it is possible to have CASS dynamically adjust ε value according to local factors, such as the event happening frequency.

2) Communication and computation overhead: The upper diagram of Fig. 5 depicts the theoretical and measured communication overhead of CASS in 250-node networks. The total measured communication overhead incurred by CASS is about $4.43J$. The lower diagram of Fig. 5 shows the theoretical and measured computation overhead of CASS in 250-node networks. We can see that the total measured computation overhead is only about $1.47mJ$. The theoretical value curve and measured curve grow apart after about $1,000$ seconds because sensors begin to die due to power outage. The overall overhead of CASS including the communication and computation overhead is about $4.43147J$, only 0.8% of total energy consumption.

3) The effect of location error: We assume that each sensor is aware of its own location. In practice, however, location obtained by a sensor is often noisy and incurs error. For example, autonomous civilian GPS are typically

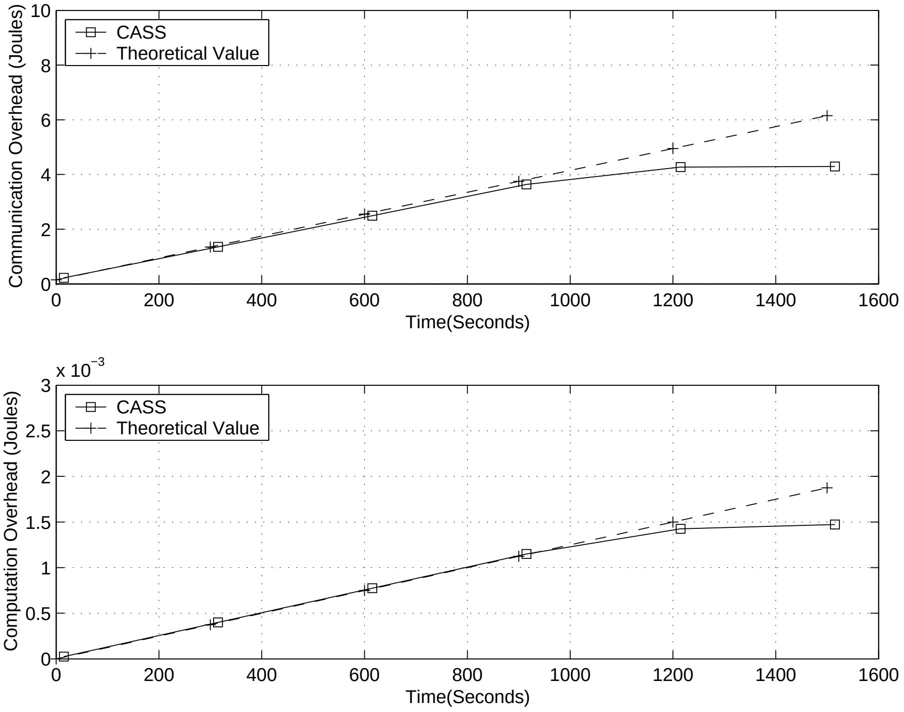

Fig. 5. Communication and computation overhead of CASS (250 nodes)

accurate to about 15 meters. Thus, it is necessary to study the performance of CASS against location error. Fig. 6 shows the coverage performance of LEACH-CC by varying location errors in 250-node networks. We assume that the location error is uniformly distributed between 0 and the maximal location error, which is configured as 0, 5, 10 and 20 meters in the simulation. As show in Fig. 6, both gross coverage and coverage extensity degrades when location is less accurate. This is because that inaccurate location information incurs errors in coverage contribution computation, which, in turn, causes sensors to make wrong decisions on sensing state. We can see that, even with the maximal location error of 20 meters, LEACH-CC still provides decent sensing coverage, *e.g.*, about 30% more gross coverage and 99.9116% coverage extensity at worst.

VI. CONCLUSIONS

We have described a new coverage maintenance scheme, which allows sensors to decide the state of their sensing units in a distributed manner according to local coverage information. A unifying framework is proposed to incorporate different connectivity and coverage maintenance schemes. Under such a framework, we evaluated the performance of the proposed scheme by integrating it with a well-known existing connectivity maintenance

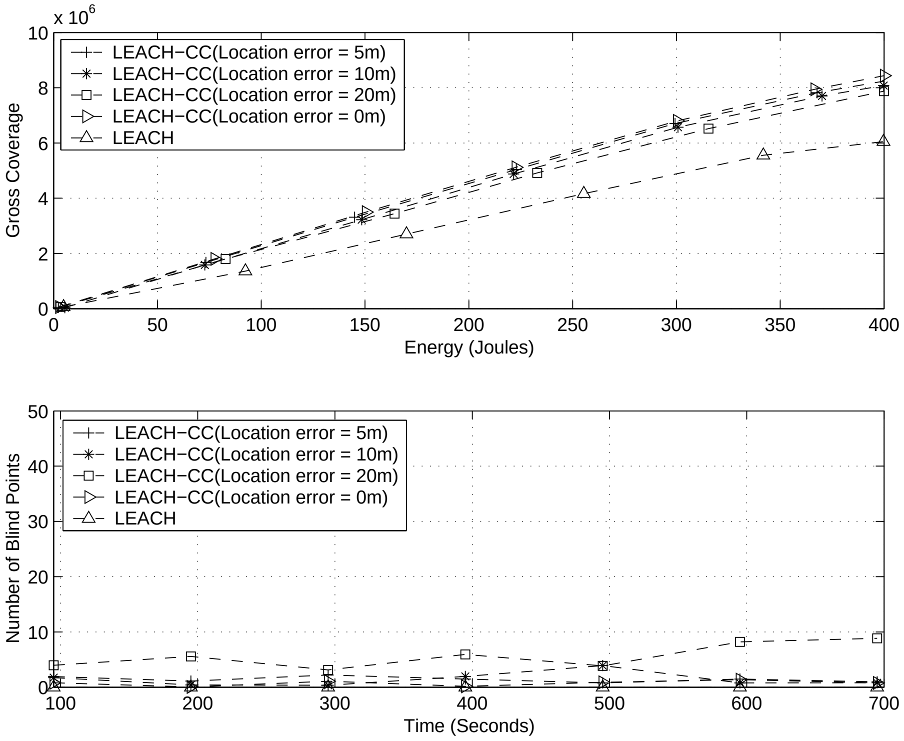

Fig. 6. The effect of location error on network coverage performance (250 nodes)

protocol. The simulation results verified the validation of the proposed framework and showed that the proposed scheme achieves considerable improvements on energy efficiency for coverage maintenance with low overhead while presenting sufficient robustness against location error.

ACKNOWLEDGMENT

This research is supported by the NSF through grants ANI-0083074, ANI-9903427 and ANI-0508506, by DARPA through grant MDA972-99-1-0007, by AFOSR through grant MURI F49620-00-1-0330, and by grants from the California MICRO and CoRe programs, Hitachi, Hitachi America, Hitachi CRL, Hitachi SDL, DENSO IT Laboratory, NICT (National Institute of Communication Technology, Japan), Nippon Telegraph and Telephone (NTT), NTT Docomo, NS Solutions Corporation, Fujitsu and Novell.

REFERENCES

[1] XBOW Inc., http://www.xbow.com/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf. *MPR/MIB User's Manual*, Last visited on 06/30/2005.

[2] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, *Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks*, ACM Transaction on Sensor Networks, 1(1):36–72, August 2005.

[3] W.R. Heinzelman, A. Chandrakasan and H. Balakrishnan, *Energy-Efficient Communication Protocol for Wireless Microsensor Networks*, Proceedings Hawaii International Conference on System Sciences (HICSS), Maui, Hawaii, Janurary 2000.

[4] R. Wattenhofe, L. Li, P. Bahl and Y. Wang, *Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks*, Proceedings IEEE INFOCOM, Anchorage, Alaska, Apri 2001.

[5] A. Cerpa and D. Estrin, *ASCENT: Adaptive Self-Configuring Sensor Network Topologies*, Proceedings IEEE INFOCOM, New York, June 2002.

[6] Y. Xu, S. Bien, Y. Mori, J. Heidemann and D. Estrin, *Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks*, Technical Report 6, University of California, Los Angeles, 2003.

[7] D. Tian and N.D. Georganas, *A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks*, Proceedings ACM WSNA, Atlanta, Georgia, September 2002.

[8] C. Hsin and M. Liu, *Network Coverage Using Low Duty-Cycled Sensors: Random and Coordinated Sleep Algorithms*, Proceedings IPSN, Berkeley, California, April 2004.

[9] C. Huang and Y. Tseng, *The Coverage Problem in a Wireless Sensor Network*, Proceedings ACM WSNA, San Diego, California, September 2003.

[10] Y. Gao, K. Wu and F. Li, *Analysis on the Redundancy of Wireless Sensor Networks*, Proceedings ACM WSNA, San Diego, California, September 2003.

[11] G. Xing, C. Lu, and R. Pless, *Co-Grid: an Efficient Coverage Maintenance Protocol for Distributed Sensor Network*, Proceedings IPSN, Berkeley, California, April 2004.

[12] Z. Abrams, A. Goel and S. Plotkin, *Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks*, Proceedings IPSN, Berkeley, California, April 2004.

[13] T. Yan, T. He and J.A. Stankovic, *Differentiated Surveillance for Sensor Networks*, Proceedings ACM SenSys, Los Angeles, California, September 2003.

[14] H. Zhang and J.C. Huo, *Maintaining Sensing Coverage and Connectivity in Large Sensor Networks*, Ad Hoc & Sensor Wireless Networks, 1(1-2):89–123, March 2005.

[15] J. Albowicz, A. Chen and L. Zhang, *Recursive Position Estimation in Sensor Networks*, Proceedings the IEEE International Conference on Network Protocols (ICNP), Riverside, California, November 2001.

[16] N. Bulusu, J. Heidemann and D. Estrin, *GPS-less Low Cost Outdoor Localization For Very Small Devices*, IEEE Personal Communications, 2000.

[17] N.B. Priyantha, A. Chakraborty and H. Balakrishnan, *The Cricket Location-Support System*, Proceedings ACM MOBICOM, Boston, Massachusetts, August 2000.

[18] S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, *Exposure In Wireless Ad-Hoc Sensor Networks*, Proceedings ACM

MOBICOM, Rome, Italy, July 2001.

[19] Robosoft Advanced Robotics Solutions, http://www.robosoft.fr/SHEET/02Local/1001LAUN/LAUN.html, Last visited on 09/07/2005.

[20] J. Pister, *Energy and Performance Considerations for Smart Dust*, International Journal of Parallal and Distributed System and Networks, 4(3):121–133, 2001.