DEEPS: Deterministic Energy-Efficient Protocol for Sensor networks

Dumitru Brinza and Alexander Zelikovsky
Department of Computer Science, Georgia State University, Atlanta, GA 30303.
E-mail: dima@cs.gsu.edu, alexz@cs.gsu.edu

Abstract

Energy consumption in monitoring and communication protocols for wireless sensor networks became one of the most important performance objective. We assume a commonly accepted sensor network model in which sensors can interchange idle and active modes both for monitoring and communicating. We introduce a reliability requirement for distributed target-monitoring protocols and prove that previously considered protocols [1],[2] are reliable.

In this paper we propose a new Deterministic Energy-Efficient Protocol for Sensor networks (DEEPS) aimed at prolonging the lifetime. We prove that DEEPS is reliable and compare DEEPS with several known target-monitoring protocols in NS2 environment using LEACH[4] protocol for data delivery to the base. We implemented the full-fledged simulation of the monitoring protocols on NS2 combined with LEACH[4] as a communication protocol, and performed extensive experimental study of several protocols showing almost 2 times increase in the lifetime for DEEPS over known protocols.

1. Introduction

Advances in VLSI and wireless technologies have resulted in creation of affordable small wireless sensors which become essential providers and active participants of our information era. It is not surprising that wireless sensors networks have shifted into the focus of considerable research during the past few years.

A sensor network is composed of a large number of sensor that can be densely deployed close to the targeted environment which is required to monitor. The position of sensor nodes in general are not required to be engineered or predetermined. This allows fast random deployment in inaccessible terrains or hazardous environments. Some of the most important application areas of sensor networks include military, natural calamities such as forest fire detection and tornado motion, different kinds of surveillance.

When compared to traditional ad hoc networks, the most noticeable point about sensor networks is that, they are limited in power, computational capacities, and memory. A wireless sensor node typically consists of sensing hardware, embedded processor& memory, transceiver and batteries. In most applications, replenishment of power resources might be impossible. The energy density of batteries has only doubled every 5 to 20 years, depending on the particular chemistry, and prolonged refinement of any chemistry yields diminishing returns. This shows that energy saving will be a critical issue for the foreseeable future.

The main metric for measuring performance of sensor network protocols is the *sensor network lifetime* measured by total time during which the network perform its monitoring duties without recharging batteries. Other important parameters are *latency* (which is a specified upper bound on the delay with which collected data should be delivered to the base) and *fault-tolerance* (non-catastrophic individual sensor failures should be timely and reliably repaired).

Power consumption of a sensor can be divided into three domains corresponding to the main sensor tasks: data collection (monitoring events), data aggregation, and data delivery to the base. It has been shown that the energy necessary to perform 3000 instruction is the same as to send 1 bit for 100m by radio [10]. Most previous efforts on optimizing power consumption has been concentrated on routing protocols, in particular, delivery of data to the base and data requests from the base to sensor nodes.

In this paper we will concentrate on maximizing sensor network lifetime by using distributed algorithms for *continuous* and *event-driven* models.

Monitoring schedule protocols have been proposed in [9] and recently in [3]. The Probing Environment and Adaptive Sleeping (PEAS) proposed in [3] is solely based on random decisions and no load balancing is considered. In [9], the authors emphasize differentiated degree of coverage and their approach allows each sensor to decide itself a monitoring schedule. To balance the energy consumption, each sensor produces a number of schedules, transmits to its neighbors, and selects the most suitable schedule. The randomized heuristics of [9] are analyzed through simulations and no approximation ratio is given. In [1], first provably good centralized approximation algorithms as well as formally described distributed algorithms are given. Cardei et al. [3] propose the same model as [1] and present centralized heuristics without proving any approximation ratio. The three papers cited in this paragraph do not take into account the cost of communicating to the base the data collected by monitoring.

This work is a continuation and refinement of [1], [2]. Our contributions for continuous and event-driven monitoring models include:

1. a novel target-coverage model and defining reliable monitoring protocols,
2. a new provably reliable Deterministic Energy Efficient Protocol (DEEPS) based on upper bounds on target coverage, which is an improvement over Load-Balancing Protocol (LBP) from [1]
3. first full-fledged simulation of the monitoring protocols on NS2 with LEACH as a communication protocol, showing almost 2 times increase in the lifetime for DEEPS over existing protocols.

The rest of the paper is organized as follows. In the next section we will formally describe the sensor network model under consideration and introduce reliability constraint for monitoring protocols. In Section 3, we describe the load-balancing protocol from [1], show that it is reliable and give an instance of sensor networks showing limits of its efficiency. Section 4 describes our new proposed protocol DEEPS and proves its efficiency and reliability. NS2+LEACH simulation results for DEEPS, LBP and other protocols is described in Section 5.

2. Continuous and Event-Driven Sensor Network Model

Geometric Model. Our model of a sensor network is close to one described in [7, 8]. We assume that sensors are sprayed over the region R which is required to monitor, and each sensor p_i has its own *monitored region* R_i which it *covers*, i.e., p_i can collect the trustful data from R_i without help of any other sensor. Assuming the regions R_i are disks, it has been shown in [1] that an efficient data structure reduces the entire monitored region to the small set of *faces* ($O(n^2)$, where n is the number of sensors) which are the regions covered by the same subset of sensors.

We also consider a case when it is required to cover certain points (targets) rather than 2-dim regions. By putting a target in each face, the data structure above effectively shows that the case of covering targets is a more general case. Therefore, we assume that each sensor knows its own coordinates as well as the ID's of all its covered targets.

Communication and Monitoring Models. Each sensor can be in the following communication modes: sleeping, listening, receiving and sending and two monitoring modes: idle and active. In the sleeping mode, nodes do not hear any packets but it is possible to use wake-up mechanisms such as in [5]. In general we assume that the energy cost for supporting wake-up mechanism is low.

Following LEACH[4], when a sensor is active, then it is supposed to send data packets to the base (usually via clusterheads) with a prescribed frequency in case of continuous model or randomly in case of event-driven model.

We also assume that the number of sensors largely exceeds the amount necessary to monitor the required region R. Therefore, it is possible to turn some sensors in the sleepy idle mode saving their energy and prolonging the network lifetime. Sensors can interchange their mode multiple times and possible energy loss due to changing mode can be taken in account.

Monitoring Protocol Models. The main requirement for all monitoring protocols is to be *reliable*, i.e., always cover each target unless all sensors covering such target are exhausted. Surprisingly, a simple rule that a sensor should stay active if it is the last sensor covering a target does not guarantee reliability in the presence of lossy or slow communication. Therefore the reliability requirements imply that a protocol is reliable only if for any target $t \in T$ there is at least one sensor $s \in S$ which cannot become idle unless it knows that an active senor covers t.

We consider a class of monitoring protocols in which each sensor can broadcast just before its battery exhaustion. This is an important assumption which allows sensors to become sleepy and do not waste energy for listening. Assuming that messages from nearly-exhausted sensors do not happened frequently, it is possible to use wake-up mechanisms such as in [5]. So we further assume that neighboring sleepy sensors can wake-up in order to decide who will substitute the exhausted sensor.

We further assume that each sensor s can communicate with all sensors sharing faces with s, otherwise, we can, e.g., increase for this purpose communication range. The geometric data structure can be easily built in the distributed manner as follows: (1) each sensor broadcast to its neighbors its id and geographical position, (2) each sensor determines all the faces in its monitored area and associate with each face the id's of all sensors covering this face.

3. Load-Balancing Protocol for Sensing

In this section we briefly describe a revised load-balancing protocol (see [1]), show that it is reliable and give an instance of sensor networks illustrating limitations of its efficiency. Everywhere below, we assume that we do not have ties in energy supplies – the ties are broken using id's of targets or sensors.

There are two main questions which should be answered by any monitoring protocol: (1) What are the rules for each node to decide whether it should become idle or active (i.e., turn itself on or off)? and (2) When should nodes make such decision?

One main idea behind all sensing protocols is to keep at any time a minimal sensor cover, i.e., changing the mode of a sensor from active to idle would result in some target becoming un-monitored. While one cannot prove just based on the minimal property an approximation ratio, the duality paradigm from the previous section suggest that the minimal property must be maintained.

The Load-Balancing Protocol (LBP) for sensing improves over the protocol from [8] by allowing sensors to freely exchange on and off states (see [1]). As a result, if a certain sensor is overused comparatively with its neighbors, then it gets a break. The main intuition behind LBP is that by means of load-balancing one keeps alive a maximum number of sensors as long as possible. Then for a longer period sensors would have more freedom to choose who will stay active to cover targets. We first describe the three states of each node of LBP:

on = active and sleepy modes, i.e., the sensor decided to monitor its targets and does not waste energy for communication;

off = idle and sleepy modes, i.e., the sensor decided that it can stop wasting any energy (since, e.g., all his targets are taken care of);

alert = active and listening modes, i.e., the sensor monitors its targets and should change its state to either active or idle in order to reduce energy consumption.

Each alert sensor knows in which state all its neighbors – any state transition is immediately broadcasted to the neighbors together with the current energy (battery) supply.

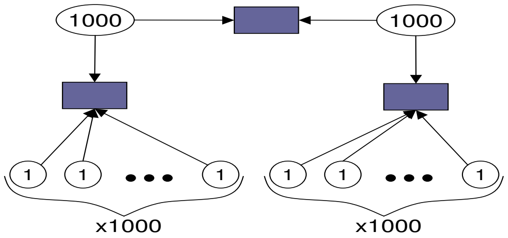

Figure 1. **An example with the optimal schedule equal to 2000 time units long. LBP schedule is twice shorter since it uses the 1000-battery sensors simultaneously for 999 time units. DEEPS schedule almost matches the optimum schedule.**

The state transitions from the alert state are described by the following two rules:

(1) (*on-rule*) whenever a sensor s has a target covered solely by itself (no alert- or on-sensor covers it), s switches itself on, i.e., changes its state to "on".

(2) (*off-rule*) whenever each target covered by a sensor s is also covered either by an on-sensor or an alert-sensor with a larger battery supply, s switches itself off, i.e., s changes its state to "off".

With a certain period (depending on energy spent on communication) all nodes are alerted using wake-up calls. After that a so called *global reshuffle* put all alert sensors to on/off states. During a global reshuffle, each sensor first broadcasts to its neighbors its targets and battery supply and in the second broadcast tells if it switches on or off.

Finally, if an active sensor exhausts its energy supply and is going to die, then using wake-up calls it alerts neighboring sensors and a *local reshuffle* puts alerted neighbors into on/off states effectively substituting the exhausted sensor. It has been shown in [1] that the LBP satisfies the following property.

Theorem 1 *[1] Each global reshuffle of LBP needs* 2 *broadcasts (to the neighbors) from each sensor and the resulted set of all active sensors form a minimal sensor cover.*

Theorem 2 *The LBP is a reliable protocol.*
Proof. The "off-rule" (2) allows a sensor to switch-off itself only if it does not have the maximum battery supply for any of its targets uncovered by on-sensors. In other words, for any target t, there is a sensor in charge of covering it, namely the sensor with the largest battery supply.

We will now describe an example showing that LBP can have an unbounded inefficiency. It consists of 3 targets (filled rectangles), 2 elliptic sensors each with 1000 batteries and 2 groups of 1000 sensors each with 1 battery (see Figure 1). The optimal monitoring schedule would be to use for 1000 time units bottom-left 1000-group and top-right 1000-battery sensor and for the next 1000 time units use the rest of sensors. Thus, the total lifetime is 2000. Instead, LBP will use the two 1000-battery sensors until they will be almost gone. As a result the top target can be covered during at most 1001 time units (even if we will continuously reshuffle free of charge). It is easy to see that the factor-2 loss can be generalized a factor-k loss by having k 1000-battery sensors covering a middle target, another k targets, each covered by one of the 1000-battery sensors and by $(k-1) \cdot 1000$ other sensors each with battery 1, such that the set of sensors covering these k targets are disjoint.

Even if the original battery supply is the same for all nodes, if ties are broken in a worst-case manner, one can end up with the example in Figure 1 when 1000 nodes of battery 1 replace each sensor of battery 1000.

The main drawback of LBP is that it balances the load of sensors (while it does not matter if they alive or not) instead of balancing energy of sensors covering the same target. In the next section we propose our new protocol which overcomes this drawback.

4. DEEPS Protocol for Sensing

This section describes our new Deterministic Energy-Efficient Protocol for Sensing (DEEPS) and proves its reliability.

We will first explain intuition behind DEEPS. It is easy to see that the best schedule for any individual target t is to make all covering sensors to be active in sequential time periods. Indeed, if any two covering sensors are active simultaneously, then the total battery supply for t is decreasing two times faster. Therefore, a good strategy is to minimize the energy-consumption rate for energy-poor targets while allowing higher energy consumption for sensors with higher total supply.

Let *sink* be a target t which is poorest for at least one sensor covering t. For sinks, one should use a single covering sensor and such sensor is better to be the richest in batteries since the intuition behind LBP (keep more sensors alive) is still valid. Thus the off-rule for DEEPS switches off poorer sensors covering a sink until a single left sensor (preferably the richest one) is switched on based on the same (as in LBP) on-rule.

A simple application of the rule above (i.e., switch off a poor sensor covering a sink) may result in loosing reliability requirement (see Figure 2 and the caption). The abandoned target is a *hill*, i.e., a target which is not the poorest for any of covering sensors.

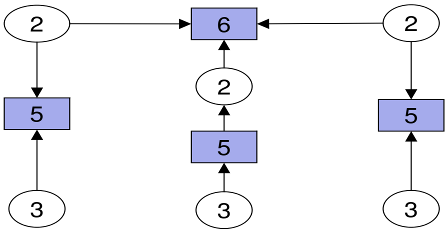

Figure 2. 3 lower sensors (circles) have 3 batteries each and the 3 uppers sensors have 2 batteries each. Therefore, 3 lower targets (rectangles) are sinks with 5 batteries each. The upper hill target will be abandoned if all three upper sensors will be switched off simultaneously.

In order to restore reliability it is necessary to place in charge of each target t at least one sensor. The sensor *in charge of* t should not switch off unless it learns that another switched-on sensor covers it. The following two rules determine which sensor should be in charge of target t:

(i) If target t is a sink, then the richest among sensors for which t is the poorest is placed in charge of t.

(ii) If target t is a hill then the sensor s covering t whose poorest target is the richest over all sensors covering t, is placed in charge of t. If there are several such sensors, i.e., several sensors with the same poorest target, then the richest among them is placed in charge of t.

Consider the example in Figure 2. Assume that the breaking-tie rule decides that the leftmost target is the richest among 3 lower targets. Then according to the rule (ii), the leftmost sensor with 2 batteries will be placed in charge of the topmost hill target and will turn itself on after two other 2-battery sensors will turn themselves off.

We are ready now to describe DEEPS. Sensors have the same states (on, off and alert) in as in LBP. The on-rule is the same as for LBP, but the off-rule is different. For any sensor, in order to learn for which targets it is in charge of, it is necessary to know the battery supply of all the targets of the neighbors. This can be achieved either by two broadcasts to the neighbors or simply by increasing communication range to 4 times of the sensing range. In our simulations we increased the range.

(1) (*on-rule*) whenever a sensor s has a target covered solely by itself (no alert- or on-sensor covers it), s switches itself on, i.e., changes its state to ”on”.

(2) (*off-rule*) whenever a sensor s is not in charge of any target except those already covered by on-sensors, s switches itself off, i.e., changes its state to "off".

DEEPS schedule for the example in Figure 1 almost matches the optimum schedule. Only the first shift will make the both 1000-battery sensors be active simultaneously. After that the top target becomes the sink and will be monitored only by one of the 1000-battery sensors at a time.

During each global reshuffle, each sensor first broadcasts to 2-neighbors (i.e., neighbors of neighbors) its targets and battery supply and in the second broadcast tells if it switches on or off.

Theorem 3 *DEEPS is a reliable protocol. Each global reshuffle of DEEPS needs* 2 *broadcasts (to the 2-neighbors) from each sensor and the resulted set of all active sensors form a minimal sensor cover.*

Proof. Reliability of DEEPS simply follows from the fact that each target has a sensor which is in charge of that target. The transition on-rule guarantees that the resulted sensor cover is minimal – each sensor s has a target covered only by s.

We now show that at some point all alert sensors will switch on or off. Indeed, consider a moment when each target is covered by at least two sensors (otherwise, one of the sensors should switch itself on by on-rule). Let t be the globally poorest target among not yet covered by on-sensors and s be the poorest sensor covering t. We will show that s is not in charge of any target and by off-rule should switch itself off. Indeed, s is not in charge of its poorest target t since there should be at least one more sensor covering t and the richest among sensors covering t will be in charge of t by rule (i). Also s is not in charge of any hill h since h is also covered by another sensor s'; either poorest target of s' is richer than t or s' also covers t but is richer than s – in the both cases by rule (ii) there is a sensor in charge of h which is different from s.

5. NS2 Simulations

For our simulations we have used the network simulator NS2 (version 2.26). The protocol LEACH[4] has been used for delivering data packets sensors to the base as well as for establishing sensor clusters. Our energy consumption model uses the following constants: transmitting energy is $0.005 \cdot d^{2.45}$(J/bit), energy for receiving $3 \cdot 10^{-6}$(J/bit), sensing energy is $5 \cdot 10^{-6}$(J/bit), idle energy consumption is 10^{-3}(W), energy consumption for listening is 0.01(W), the receiving threshold is 10^{-9}(W), success threshold is $6 \cdot 10^{-9}$(W), aggregation energy (per sensor/per packet) is 10^{-7}(J/bit). We have simulated the following scenarios: Area-1, Area-3 and Target-3.

Area-1. 10,000 targets and 100 sensors are placed uniformly at random in the square 100m×100m with the left-bottom corner at (0,0). Base coordinates are (75,150) and sensing radius is 10m. Communication radius for data packets transfer is flexible and is chosen by LEACH protocol. Communication radius for DEEPS and LBP control messages is equal to two and four sensing radiuses respectively. The initial energy of each sensor is either randomly chosen between 2J and 4J or the same (4J) for all sensors. Sensors monitor their targets continuously and send the data packets according to the TDMA scheduler determined by LEACH for each cluster. The size of data packet is proportional to sensing time.

Area-3 – same as Area-1 but all targets covered by less than 3 sensors are dropped.

Target-3 – same as area Area-3 but the number of targets is 100.

We have implemented 4 protocols: DEEPS, LBP, 1-DEEPS (which is a version of DEEPS with a single global reshuffle and local reshuffles when nodes run out of energy), and EUPS - Energy Unaware Protocol for Sensing (all sensors continuously monitor all targets).

Figure 3 gives results for the scenario Area-1 with uniform initial energy for all sensors. It is easy to see that DEEPS wins in all 4 metrics: it has less sensors active, it covers for longer time the larger portion of monitored area, its sensors are alive for longer time and its spends less total energy.

The similar advantage of DEEPS is also true for the case when the initial battery supply for each sensor is different (the energy is randomly distributed between 2J and 4J, results are not presented). For scenarios Area-3 and Target-3 DEEPS demonstrates even greater advantages comparative to results for Area-1 scenario(results are not presented).

Finally, we explored possibilities of varying reshuffle schedule for DEEPS and LBP protocols respectively. We found that for LBP protocol it is better to keep a certain uniform schedule while for DEEPS protocol it is better to increase the period between schedules from one period to another.

6. Conclusions

In this paper, we have formulated a more realistic Maximum Sensor Network Life Problem and suggested distributed algorithm for solving this problem. The distributed algorithms were tested by NS2 simulations with the LEACH[4] communication protocol.

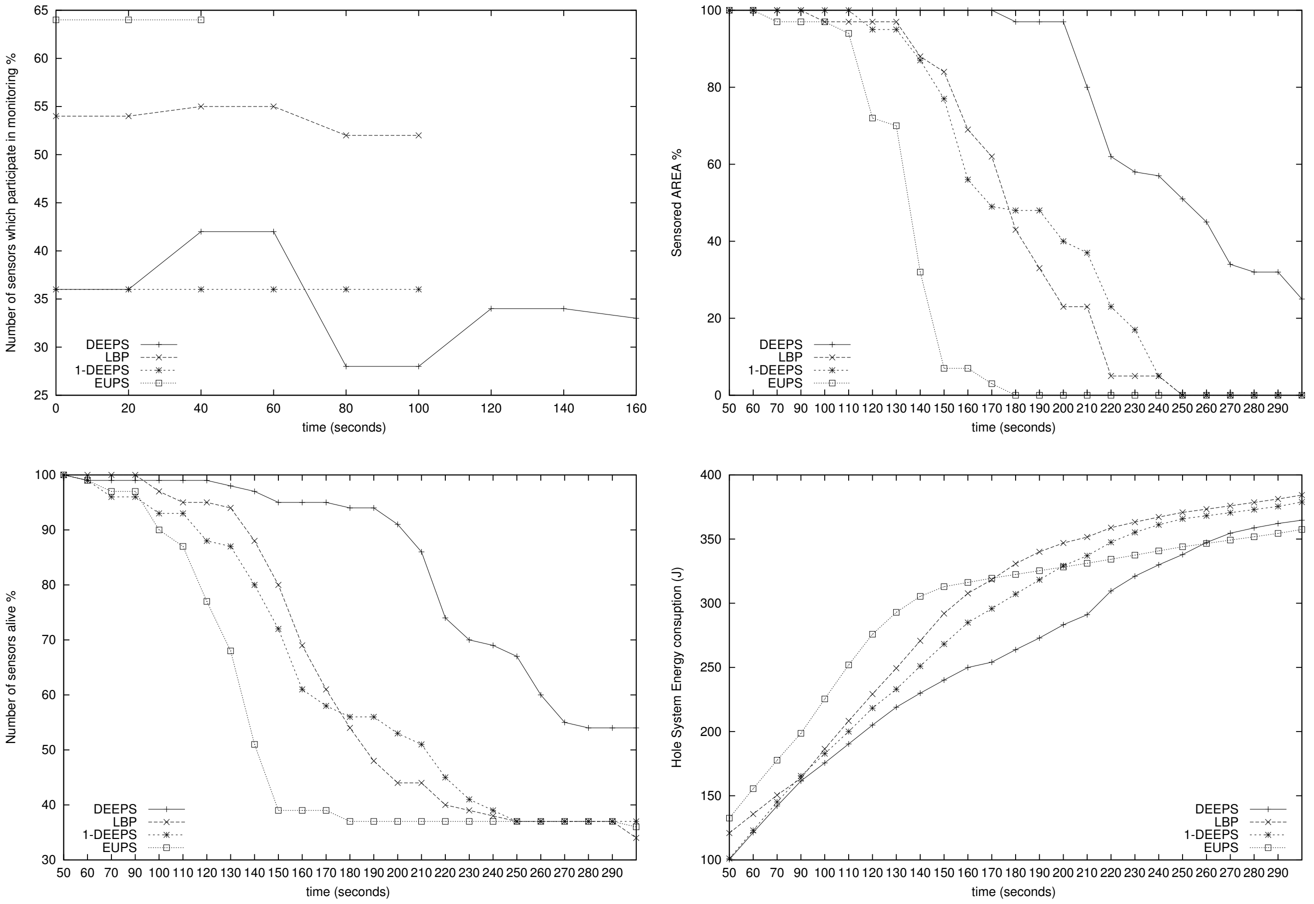

Figure 3. Number of active sensors, the percent of covered area, the percent of alive sensors and the total spent energy with respect to simulation time for Area-1 scenario with the initial energy of 4J for each sensor.

The proposed protocol DEEPS is shown to be superior to the previous monitoring schedulers.

References

[1] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky. Power efficient monitoring management in sensor networks. In *Proc. IEEE Wireless Communications and Networking Conference*, 2004.

[2] D. Brinza, G. Calinescu, S. Tongngam, and A. Zelikovsky. Energy-Efficient Continuous and Event-Driven Monitoring. In *Proc. 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*,2005

[3] M. Cardei, M.T. Thai, Y. Li, and W. Wu. Energy-efficient target coverage in wireless sensor networks. In *Proc. of IEEE Infocom*, 2005.

[4] W. Heizelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient routing protocols for wireless microsensors networks. In *Proc. Hawaii Int. Conf. on System Sciences*, 2002.

[5] L. Gu and J. Stankovic. Radio triggered wake-up capability for sensor networks. In *Real-Time Applications Symposium*, May 2004.

[6] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communication ACM*, 43(5):51–58, 2000.

[7] S. Slijepcevic and M. Potkonjak. Power efficient organization or wireless sensor networks. In *Proc. IEEE International Conference on Communications (ICC)*, pp. 472–476, 2001.

[8] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proc. ACM WSNA02*, 2002.

[9] T. Yan, T. He, and J. Stankovic. Differentiated surveillance for sensor networks. In *Proceedings of Sensys*, 2003.

[10] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communication ACM*, 43(5):51–58, 2000.