Chapter 1

Self-Organized Scheduling of Node Activity in Large Scale Sensor Networks

Sumathi Seetharaman and Ali A. Minai
Complex Adaptive Systems Laboratory
ECECS Department
University of Cincinnati
Cincinnati, Ohio 45221
seethasi,aminai@ececs.uc.edu

1.1 Introduction

Advances in microlectromechanical systems (MEMS), digital signal processing and wireless communication have led to the possibility of networkable sensors with dimensions of less than a few hundred microns. Tens or hundreds of thousands of these nodes can be randomly deployed over a region of interest. The sensors can communicate locally with each other and self-organize to form a functional sensing network.

Micro-sensor nodes can perform a limited set of functions: a) sense the environment, b) transmit and receive data packets, c) perform rudimentary information processing and d) stay inactive to conserve battery. Each node is required to perform a subset of these functions at any given time to ensure consistent network performance. A particular concern is the need to maintain *coverage*, i.e., to ensure that every location in the monitored environment is being sensed at all times. Another imperative is to conserve energy, since nodes typically have limited energy resources. Nodes in a large-scale sensor network are usually deployed randomly with relatively high density. Thus, even though each node can sense only in a limited region, the sensing regions of neighboring nodes overlap significantly. The *coverage problem* is to control the activity of nodes in a way that ensures sensing coverage while minimizing energy consumption [2].

Approaches to solving the coverage problem fall into two broad categories: a) Scheduling schemes, and b) Density control schemes. Scheduling schemes partition the set of nodes into subsets called *covers* such that each subset can completely cover the field. These covers share the burden of monitoring by taking turns at being acive. In contrast, density control schemes seek to maintain a uniform density of active nodes by activating an initial cover and activating new nodes as needed to replace those that run out of energy.

In this paper, we describe two intelligent scheduling schemes and compare their performance with some successful scheduling and density control algorithms. Based on the results, we draw some interesting conclusions regarding the ideal design space for each approach.

1.2 Coverage Algorithms

A number of interesting solutions to the coverage problem have been explored, including: a) Centralized scheduling approaches, e.g., integer linear programming[5], blue noise spatial sampling[3], etc.; b) Distributed scheduling techniques, e.g., graph coloring[4], differential coverage algorithm[7] etc., and c) Density control schemes, e.g., PEAS[8], OGDC[9], RACP[1], sponsored sector scheme[6] etc. Centralized scheduling schemes are not a feasible solution for large-scale sensor networks since the computational costs do not scale well and the energy overhead involved in collecting location information for all the nodes whenever the topology changes is unacceptable. Among the distributed schemes, the differential coverage algorithm proposed by Ting[7] exhibits good performance with a low overhead and is shown to perform better than the overlap-based sponsored sector scheme[6].

In the differential coverage algorithm, sensing periods are time sliced into intervals of duration T. Each node selects a reference time point(t_{ref}) between 0 and T randomly and broadcasts it. After receiving the reference times of neighboring nodes, each node computes the smallest schedule for each sensing field within its sensing region. A sensing field is an area within a node's sensing region that is covered by a unique set of sensors. For example, if field f_a in the sensing region of s_i is covered by one other node s_j with reference time $t_{(j,ref)}$, then the field schedule for node s_i is evaluated as:

$$\begin{aligned} t_{(i,front)} &= \{T + t_{(i,ref)} - t_{(j,ref)}\}/2 \\ t_{(i,end)} &= \{t_{(j,ref)} - t_{(i,ref)})\}/2 \\ Schedule(i, f_a) &= [t_{(i,ref)} - t_{(i,front)}] \; to \; [t_{(i,ref)} + t_{(i,end)}] \end{aligned}$$

assuming $t_{j,ref} > t_{i,ref}$. Finally, the sensing schedule determined by overlapping all field schedules is broadcast. To optimize further, each node tries to shrink its schedule as far as possible, ordered by the length of node schedules to avoid conflicts. Although this algorithm is simple and efficient, the random selection of reference times forces sensors to remain active for longer periods than necessary. We propose an alternative random selection algorithm detailed in the next section which produces better results with the same communication overhead.

In density control schemes, OGDC[9] is designed for very dense networks and nodes are added into the cover by a sequential location-based activation scheme. While this scheme may produce a good initial layout of active sensors, it is unclear how the scheme will adapt to failures. Since the density of the network reduces with time due to node failures, the probability of finding a node at a desired location diminishes with tim and the scheme may yield suboptimal solutions. PEAS[8] is an extremely adaptive and robust protocol that works by nodes adjusting their periods of "sleeping" (inactivity). Sleeping nodes periodically wake up and probe their neighborhood to find out if any nearby sensor is active. If the querying node is not necessary to provide coverage, it dynamically adjusts its sleeping period based on the observed probing rate of nearby active nodes and enters the sleep mode. Otherwise, it joins the covering set of nodes. The probing range controls the coverage redundancy while the rate of probing determines the average latency in replacement of a failed node.

1.2.1 Intelligent Scheduling

While most of the coverage schemes discussed above increase the efficiency of coverage, none of them explicitly considers the resulting spatial distribution of coverage. The random deployment of nodes causes non-uniform placement of the nodes in the field and the number of nodes covering individual points in the field varies significantly. The sensing region of each node is thus divided into multiple *sensing regions* covered by different numbers of sensors.

We address the coverage problem by allowing the nodes to make simple decisions based on observable local topology. Each node acts greedily and attempts to minimize its schedule as far as possible. However, the extent to which a node can shrink can its active sensing time depends on its *weakest region*, defined as a sensing region within its field that is covered by the smallest number of sensors. Let f_b be the weakest region for node s_j covered by one other node s_k. To cover this region efficiently, nodes s_j and s_k each need to be active for duration $T/2$ in each sensing round of length T. The *weakness degree* of each node is the number of nodes covering its weakest region. Formally, let $F_{i,sense} = \{f_{i,1}, \dots f_{i,g}\}$ be the set of sensing regions, $f_{i,j}$ covered by the node s_i and $N_{i,cover} = \{n_{i,1}, \dots n_{i,g}\}$ the number of nodes, $n_{i,j}$ covering regions $f_{i,j}$. Define:

$$\begin{aligned} n_{i,weak} &= min(N_{i,cover}) \\ A_{i,weak} &= \sum_{j|n_{i,j}=ni,weak} A(f_{i,j}) \end{aligned}$$

where $A(f_{i,j})$ is the area of region $f_{i,j}$. Thus, $A_{i,weak}$ is the total area that is most weakly covered, and $n_{i,weak}$, termed the *weakness degree* of i, is the number of sensors covering it. Our intelligent scheduling approach is based on the idea that the sensor covering the most weakly covered regions have the greatest need to conserve energy, since they are the least dispensable nodes.

To provide a baseline, we consider a *random scheduling algorithm (RSA)*, where all nodes have the same duty cycle. Each node randomly selects m slots out of a total of k slots available in each sensing round.

We define two intelligent scheduling algorithms as follows:

Adaptive Scheduling Algorithm (ASA): This algorithm enhances the RSA by incorporating an initial *neighbor discovery phase*, where nodes announce their positions and record the locations of nearby nodes. Using this information, each node can construct the coverage landscape within its sensing region and evaluate its weakness degree. Nodes with lower degrees must stay on for longer durations because they have fewer nodes to share the burden in their weakest sensing regions. Each node locally estimates its duty cycle in order to ensure complete coverage of its sensing field with probability p_{cov}.

Collaborative Scheduling Algorithm (CSA): Due to nonuniform coverage distribution in the field, the weakness degrees of neighboring nodes can vary significantly. The CSA attempts to take this into account when nodes decide their schedules. For example, suppose that a node s_i has weakness degree 2, but the other node, s_j, covering its weakest region has a weakness degree of 1. This means that s_j will have to stay on throughout the sensing round to cover *its* weakest region, and the weakest region of s_i will be covered completely by s_j. This leaves s_i free to decide its schedule based on its next weakest region, resulting in a lighter schedule.

Under the collaborative scheduling scheme, each node checks the weakness degrees of the nodes covering its weakest fields and updates its weakness degree. Nodes with lower weakness degrees are allowed to update their degrees first to avoid conflict. This scheduling allows the more critical nodes to ensure they are on for the smallest possible duration. In case of a conflict, the nodes with the largest critical area declare first.

We optimize these schedules further in a similar fashion by allowing nodes with more critical values of $\{n_{i,weak}, A_{i,weak}\}$ optimize their schedules first.

1.3 Simulations and Discussion

For simulation, we use field dimensions of $50 \times 50m^2$, node sensing and communication radius as $4m$ and $8m$ respectively and uniformly distribute 500 nodes. Each sensing round is assumed to have 20 slots. We assume that the nodes are aware of their location coordinates.

ASA and CSA both provide nearly perfect coverage for p_{cov} set to 0.8. However, CSA's coverage is slightly lower due to the fact that as nodes revise their weakness degrees, their on times decrease and random scheduling may sometimes result in low coverage. Figure 1.2 displays the distribution of sensing times of nodes for the schemes over 10 simulations. The reduction in schedules due to collaboration can be observed as the distribution shifts towards lower

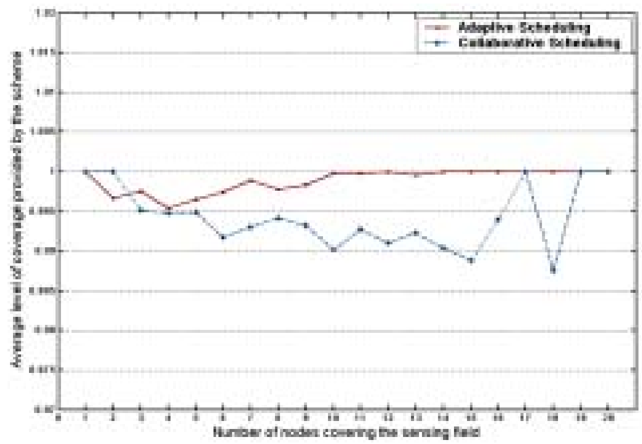

Figure 1.1: Coverage provided by adaptive and collaborative scheduling algorithms

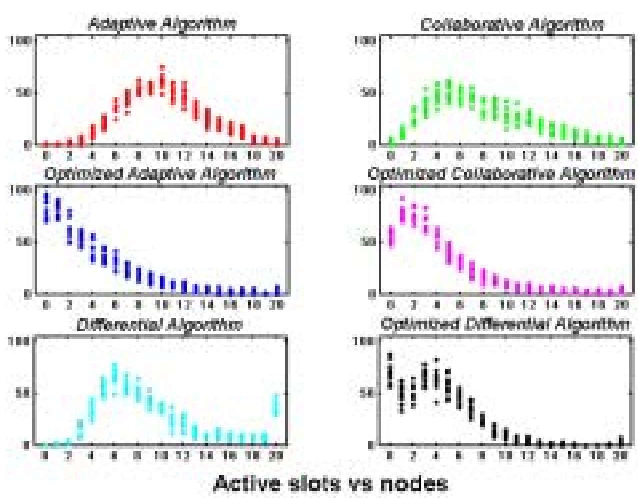

Figure 1.2: Number of active slots Vs. Number of nodes

active times. It can be seen that after optimization both adaptive and collaborative algorithms minimize node schedules to a greater extent compared to the optimized differential algorithm. This observation is further strengthened in Figure 1.3, where the differential algorithm shows a higher level of redundancy.

Both ASA and CSA show similar performances after optimization. It is interesting to note that as the coverage degree of the field increases, the average redundancy provided by all the schemes is higher since some increase in node schedules cannot be avoided due to excessive spatial overlap.

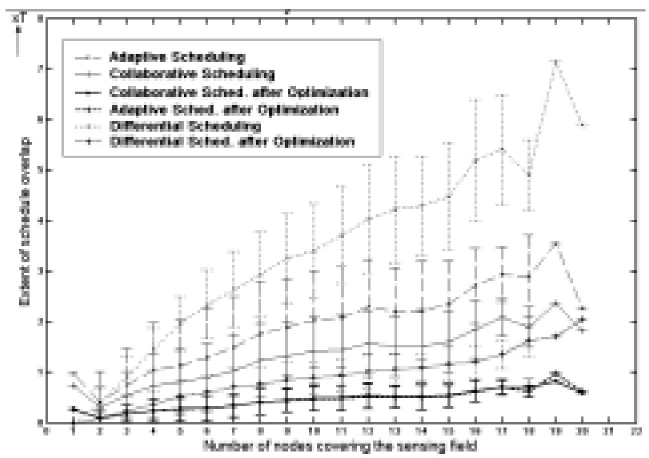

Figure 1.3: Number of nodes covering a field vs. average field redundancy

We also compare the optimized ASA with the PEAS algorithm [8] using 1000 nodes since PEAS is designed for higher densities. The probing range in PEAS is set to ensure near 100% coverage [8]. It can be observed that PEAS exhibits gradual degradation in performance and longer network lifetime although it activates a larger number of nodes at a time and provides a lower level of coverage than optimized ASA(Figure 1.4). On the other hand, optimized ASA provides nearly 100% coverage for a longer period and then undergoes abrupt degradation. As density decreases due to node failures, the coverage performance of PEAS deteriorates even further. This is because PEAS ensures that the distance between any two working nodes is at least the probing distance, but does not guarantee complete coverage. In sparse regions blind holes may appear and the algorithm is unaware of the existence of such spots. PEAS is more effective in prolonging network lifetime and shows graceful degradation with node failure whereas the adaptive scheme attempts to maximize coverage for as long as possible and is a best effort scheme.

For high density networks with fragile nodes, scheduling schemes are inappropriate since the communication overhead involved in maintaining and regularly updating neighborhood topology is high. However even at high densities, PEAS activates more nodes at a time to ensure near 100% coverage. The optimal

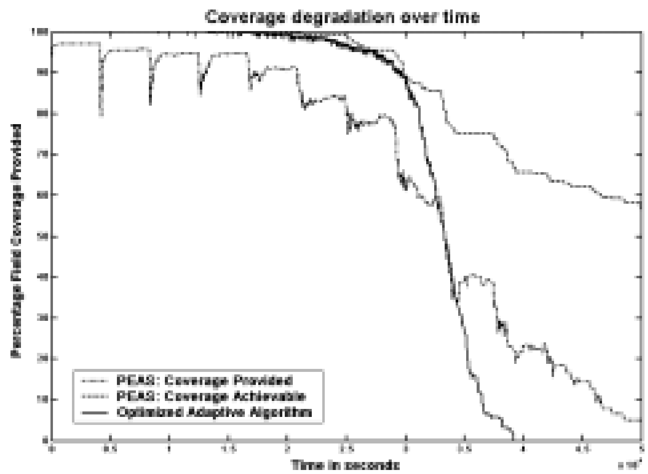

Figure 1.4: Percentage coverage provided over a period of time

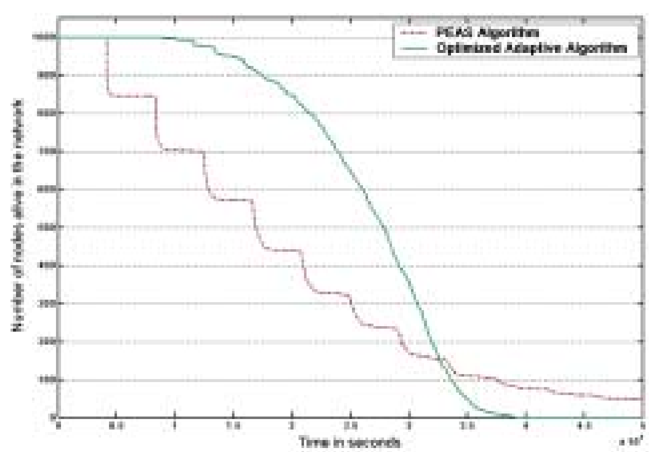

Figure 1.5: Number of active nodes remaining in the network

solution to the coverage problem might lie in a combination of scheduling and density control approaches.

1.4 Conclusion

We proposed and evaluated the performance of two randomized algorithms: Adaptive scheduling and Collaborative scheduling, supplemented by simple decision making by the nodes and mutual collaboration to negotiate a smarter schedule. The algorithms were shown to perform better than the differential coverage algorithm. We compared our scheme to a density control algorithm PEAS and observed that density control schemes fare better in extending network lifetime but at the cost of coverage quality, whereas the adaptive scheme achieves high levels of coverage for a longer duration but undergoes catastrophic degradation.

Bibliography

[1] HSIN, Chih-fan, and Mingyan LIU, "Network coverage using low duty-cycled sensors: Random and coordinated sleep algorithms", *Proceedings of the International Workshop on Information Processing in Sensor Networks (IPSN)*, (April 2004).

[2] HUANG, Chi-fu, and Yu-Chee TSENG, "The coverage problem in a wireless sensor network", *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, ACM Press (2003), 115–121.

[3] IGNJATOVIC, Zeljko, Mark PERILLO, Wendi HEINZELMAN, and Mark. BOCKO, "An energy conservation method for wireless sensor networks employing a blue noise spatial sampling technique", 2nd IEEE Upstate New York Workshop on Sensor Networks, (October 2002).

[4] KO, Bong-Jun, Keith ROSS, and Dan RUBENSTEIN, "Conserving energy in dense sensor networks via distributed scheduling, extended abstract", Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks(WiOpt '03), (March 2003).

[5] MEGUERDICHIAN, Seapahn, and Miodrag POTKONJAK, "Low power 0/1 coverage and scheduling techniques in sensor networks", *Tech. Rep. no. 030001*, UCLA, (January 2003).

[6] TIAN, Di, and Nicolas D. GEORGANAS, "A coverage-preserving node scheduling scheme for large wireless sensor networks", Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, (2002), 32–41.

[7] YAN, Ting, Tian HE, and John A. STANKOVIC, "Differentiated surveillance for sensor networks", Proceedings of Sensys, (August 2003).

[8] YE, Fan, Gary ZHONG, Jesse CHENG, Songwu LU, and Lixia ZHANG, "Peas: a robust energy conserving protocol for long-lived sensor networks", Proceedings. 23rd International Conference on Distributed Computing Systems (IEEE ICDCS), (2003), 28 – 37.

[9] ZHANG, Honghia, and Jennifer C. HOU, "Maintaining sensing coverage and connectivity in large sensor networks", NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, (February 2004), Invited paper.