Conserving Energy in Dense Sensor Networks via Distributed Scheduling

Bong-Jun Ko, Keith W. Ross, Dan Rubenstein

Conserving Energy in Dense Sensor Networks via Distributed Scheduling (Extended Abstract) *

Bong-Jun Ko
Columbia University

Keith Ross
Polytechnic University

Dan Rubenstein
Columbia University

1 Introduction

In wireless sensor networks, sensors are often battery-powered and cannot be recharged. Even the simple act of sensing its environment can consume significant amounts of these devices' energy. To reduce the likelihood that an insufficient number of sensors are deployed within a region, sensors will often be deployed at a greater density than what is expected to be necessary. The sensing devices can utilize this increased density to conserve energy by time-sharing the sensing load needed to cover their respective sensing region.

Ideally, to maximize sensor lifetimes, sensors that cover a shared region can coordinate their responsibility for covering that region with a schedule that has all but one of the sensors turned off at any given time within that region. What complicates this scheduling, however, is the fact that neighboring regions are covered by non-identical but overlapping sets of sensor devices. *In this paper, we perform an initial investigation into the problem of scheduling the on periods of sensor nodes in densely covered regions. Our goal is to reduce energy consumption per sensor while maintaining sufficient coverage of all regions that the sensors were deployed to cover.*

Few works have considered the problem of scheduling among multiple devices in dense regions. [4] and [3] are perhaps most closely related to our work here, both of which deal with energy-coverage problem in sensor networks. They both use probablistic probing schemes to determine when a sensor can safely go to sleep without losing coverage. Our work differs in that sensor coverage times are pre-determined, obviating the need for sensors to participate in a polling process that uses valuable energy to trasmit and receive polling messages.

We introduce our problem formulation in next section, describe our scheduling algorithm and prove some of its basic desirable properties in Section 3, and conclude in Section 4.

2 Problem Formulation

In this section, we describe our model of a sensor network and our scheduling objective within the context of this model. The sensor network consists of N sensor nodes numbered 1 through N deployed within a topological region where the distance between pairs of sensors is well-defined and the triangle inequality holds. Each node i is responsible for sensing some property within its surrounding region of radius r_i. When a node j lies within distance r_i from node i, we say that j is in i's *range*, and set $c_{ij} = 1$. Otherwise, j is not in i's range and $c_{ij} = 0$. Note that $c_{ii} = 1$ since i is always in its own range. The value of r_i is selected such that measurements collected by sensors in i's range are acceptable substitutes for measurements taken by node i itself. Thus, node i's region is "covered" when some j in i's range (including i itself) is actively sensing.

We assume that nodes in the network are time-synchronized into a cycle of K time slots, and that each node is capable of *turning on* or *turning off* its sensing device at the beginning of any slot. A sensor *schedule* determines a priori the slots for which each sensor is active (on).

If j is in i's range and if j is turned on at time slot $k \in \{1, \cdots, K\}$, then we say i is *covered by* j during time slot k. We let $x_{ik} = 1$ if the schedule has i turned on during time slot k. Otherwise $x_{ik} = 0$. We define $S_i = \{k : x_{ik} = 1\}$. The fraction of time for which i is active is proportional to $|S_i|$, so that reducing $|S_i|$ provides a means to conserve power and extend the sensor's lifetime.

We define $\rho_i \in [0, 1]$ to be the fraction of time that node i's region must be covered. In the context of our scheduling problem, we require for each i that

$$\sum_{k=1}^{K} \left(1 - \prod_{j=1}^{N} (1 - c_{ij} x_{jk}) \right) \geq \rho_i K, \qquad (1)$$

i.e., i is covered by the union of nodes in its range for at least $\rho_i K$ of the slots, and hence, its region is covered at least a fraction ρ_i of the time.

Our ultimate objective is to derive a schedule that determines, for each node i, the set of slots that are assigned to each node to maximize the network's operational lifetime with each sensor's area covered for the requisite fraction of time, ρ_i. There are several variants of this optimization problem. For instance, the criterion can be to maximize the time until some region can no longer be covered for the fraction of time, ρ_i, or to maximize the average lifetime for which regions are covered for the fraction of time, ρ_i. These problems are known to be hard to solve. For instance, if $\rho_i = 1$ for all i, the set covering problem, known to be NP-hard and approximable only to within $O(\log n)$, can be reduced to our problem.

Our approach is to break the problem into two stages. In the first stage, each node is assigned a number of slots, i.e., the value of each $|S_i|$ is fixed, at which point the specific metric to be optimized (e.g., the "lifetime" of the network) can be computed. Each $|S_i|$ can be chosen by solving a much simpler problem of finding an assignment

*This material was supported in part by the National Science Foundation under Grant No. ANI-0117738 and CAREER Award No. ANI-0133829. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

of number of slots to each node so that the "lifetime" of the network is acceptable (i.e., larger than some threshold, T). The second stage searches for feasible allocations of slots that compose each S_i such that the condition in (1) holds for each i.

3 Sensor Scheduling Algorithm

In this section, we describe the decentralized, distributed algorithm we use during the second stage to schedule each nodes' specific slots for which it is turned on. Note that this scheduling is performed after the number of slots, $|S_i|$, is assigned. Our algorithm assumes that each node can determine the geographic distance to other nodes in its range, e.g., by measuring SNR of radio signals. We represent this knowledge via a graph $G = (V, E)$, where V represents the set of sensor nodes in the sensor network and E is the set of edges, where, if node j is in node i's range or i is in j's range, i connects to j via a weighted edge whose weight is the same as the geographic distance between i and j .

We view the sensor node scheduling problem as a *node coloring problem*, where node i is to be assigned $|S_i|$ colors out of K possible colors, where each color represents a slot in the K-slot cycle. The coloring satisfies the constraint specified in (1) if the number of different colors assigned to nodes within distance r_i of i is no less than $\rho_i K$.

If one chooses the $|S_i|$ colors to assign to node i at random (without replacement), then the only way to guarantee that the condition in (1) is met is to ensure that some node j in i's range satisfies $|S_j| \geq \rho_i K$. We now present a distributed algorithm that lowers this bound. In particular, the bound is significantly lower in dense graphs where many nodes are within i's range. Furthermore, simulation results (not presented here) show that the likelihood that the schedule chosen by the algorithm is much more likely to meet the condition in (1) than the schedule chosen by a random coloring.

The coloring algorithm we apply is described in detail in our previous work [1]. There, we consider an arbitrary graph where each node is assigned a single color out of K possible colors. A node changes its color from a color c_1 to a color c_2 when the closest other node of color c_1 (along the shortest path in the graph to this other node) is closer than the closest node of color c_2. We prove that as long as one node changes its color at a time, the graph eventually reaches a "stable" coloring where no node can further increase this distance by changing its color. Intuitively, by increasing the distance between nodes of the same color, one increases the variety of colors within a given region. A distributed mechanism is clearly necessary in a sensor environment where centralized coordination and global (broadcast) communication is not feasible. In [2], we develop a decentralized algorithm that provably reaches a stable coloring by passing messages through neighboring nodes.

Mapping our scheduling problem where each node is assigned multiple slots into the framework of our coloring algorithm where each node is assigned a single color is straightforward. Sensor node i is mapped to $|S_i|$ nodes that are all connected to one another via edges of length 0. If an edge of length d connects i with j in the sensor graph, then an edge of length d connects all nodes that are mapped from i to all nodes that are mapped from j.

The following lemma presents a surprising sufficient condition:

Lemma 1 *Let R_i be the set of nodes that lie within distance $r_i/3$ of node i. If $\sum_{j \in R_i} |S_j| \geq \rho_i K$, then our distributed algorithm results in a coloring in which (1) holds for node i.*

Proof: The proof is by contradiction. Assume that $\sum_{j \in R_i} |S_j| \geq \rho_i K$ but that the number of colors reachable within distance r_i from i is less than $\rho_i K$ in a stable coloring. Obviously, fewer than $\rho_i K$ colors are reachable within distance $r_i/3$ from i. Thus two nodes, j_1 and j_2, within distance $r_i/3$ from i are assigned the same color, and hence these two nodes are distance at most $2r_i/3$ from one another. Consider a color that does not appear within distance r_i of i. Then j_1 and j_2 are at distances greater than $2r_i/3$ from any node of this other color. Hence, the current coloring is not stable, as either j_1 or j_2 would change one of their colors by the algorithm. ■

Hence, one can also ensure that i's region of radius r_i is covered for a sufficient fraction of time ρ_i by ensuring that the sum of slots provided by nodes within distance $r_i/3$ from i (including i itself) is no less than $\rho_i K$. In densely covered areas, the number of slots that nodes must be assigned to meet this condition is significantly less than the condition that must be met to ensure coverage in a random coloring.

4 Conclusion and Future Work

In this paper, we investigate a distributed algorithm that schedules sensor on-periods in an effort to conserve energy while ensuring the fraction of time regions are actively being covered is sufficient. We propose a distributed algorithm to implement this scheduling and show that, by ensuring that a sufficient number unscheduled slots are allocated in a smaller region, the scheduling assures a sufficient number of slots are covered in the larger region.

We plan on extending this work along two fronts. First, we wish to see how the schedule constructed by our distributed algorithm compares to a schedule where nodes randomly select slots in which they turn on. Second, we wish to further investigate methods for the first phase, i.e., developing algorithms to appropriately select the number of slots $|S_i|$ to assign to each node i.

References

[1] B.-J. Ko and D. Rubenstein. A greedy approach to replicated content placement using graph coloring. In *SPIE ITCom Conference on Scalability and Traffic Control in IP Networks II*, July 2002.

[2] B.-J. Ko and D. Rubenstein. Replica placement for emerging networks via an asynchronous, decentralized, graph-coloring algorithm. Technical report, Columbia University, November 2002.

[3] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the first ACM international workshop on Wireless sensor networks and applications 2002*, October 2002.

[4] F. Ye, G. Zhong, S. Lu, and L. Zhang. Energy effi cient robust sensing coverage in large sensor networks. Technical Report.