On Distributed Algorithms for Maximizing the Network Lifetime in Wireless Sensor Networks *

Akshaye Dhawan
Georgia State University
Atlanta, Ga 30303
akshaye@cs.gsu.edu

Abstract

A key challenge in Wireless Sensor Networks (WSNs) is that of extending the lifetime of these networks while maintaining certain coverage goals. Existing work has studied scheduling sensors into a sleep-sense cycle based on simple greedy criteria or by using centralized optimization techniques. We look beyond these greedy heuristics to study the problem structure that exists between different cover sets. In this position paper, we argue that improved distributed algorithms can be designed by paying attention to the inherent dependency that exists between different cover sets since they share sensors in common. In our work on this problem [5–7, 9], we propose a model for capturing the dependencies between different cover sets, examine localized heuristics based on this dependency model and present various improvements on the basic model. These heuristics represent a 20-30% increase in the network lifetime over the existing work [2, 3] which uses greedy criteria to make scheduling decisions. This work has opened up a new approach to designing distributed scheduling algorithms.

Keywords: Wireless Sensor Networks, Target Coverage, Distributed Algorithms, Maximum Lifetime Problem

1 Introduction

Wireless Sensor Networks (WSNs) are networks of low cost sensors equipped with a radio interface. These sensors are deployed in large numbers to monitor a geographical area of interest and transmit this data to gateway nodes. A key constraint of these networks is energy since individual sensors are equipped with a battery that cannot be replenished after deployment.

The lifetime of the network is defined as the amount of time that the network can satisfy its coverage objective, i.e., the amount of time that the network can cover its *area* or *targets* of interest. Having all the sensors remain "on" would ensure coverage but this would also significantly reduce the lifetime of the network as the nodes would discharge quickly. A standard approach taken to maximize the lifetime is to make use of the overlap in the sensing regions of individual sensors caused by the high density of deployment. Hence, only a subset of all sensors need to be in the "on" or "sense" state, while the other sensors can enter a low power "sleep" or "off" state. The members of this active set, also known as a *cover* set, are then periodically updated so as to keep the network alive for longer duration. In using such a scheduling scheme, there are two problems that need to be addressed. First, we need to determine how long to use a given cover set and then we need to decide which set to use next. This problem has been shown to be NP-complete [1, 4].

*This work was carried out under the supervision of Dr. Sushil K. Prasad, Professor, Georgia State University

Existing work on this problem has looked at both centralized and distributed algorithms to come up with such a schedule. The distributed algorithms typically operate in rounds. At the beginning of each round, a sensor exchanges information with its neighbors, and makes a decision to either switch on or go to sleep. In most algorithms, the sensor with some simple greedy criteria like the largest uncovered area [8], maximum uncovered targets [2], etc. is selected to be on. Due to space constraints, we do not provide a detailed discussion of related work.

In this extended abstract we highlight our contributions to the design of distributed and localized algorithms that maximize the lifetime of sensor networks as presented in [5–7, 9]. In designing greedy algorithms, the focus is on identifying sensors that will turn on for a given round, based on a greedy criteria. We take a different approach to this problem by trying to model the dependency between cover sets. Although globally there are an exponential number of possible cover sets making the problem intractable, the number of local cover sets, those minimal subsets of neighboring sensors covering nearby targets, is usually small. This opens up the problem to individual sensors distributively constructing the local covers and employing them as possible local configurations to systematically transition through them to arrive at a good neighborhood sense-sleep decision for each reshuffle round. What is more interesting, however, is how these cover sets influence each other. For example, if two cover sets share one or more sensors, their weakest common sensor is an upper bound on the lifetime of both covers collectively. This is because using either cover set reduces the battery of the common sensors. To model such interactions, we define the local lifetime dependency (LD) graph.

For the remainder of this extended abstract, we highlight key features common to the distributed algorithms that we proposed in [5–7, 9]. In Section 2 we discuss the LD graph model and the basic heuristic that uses this graph. In Section 3 we present some simulation results of the different heuristics and compare them to existing work. These heuristics represent a 10-20% improvement in lifetime over [2, 3]. This work has been extended into a framework to solve the area and k-coverage problems in [5, 7]. In [6] where we examine the idea of how an optimal schedule would use the LD graph and design heuristics based on provable properties of the optimal schedule. These heuristics show a lifetime improvement of 25-30% over existing work. Finally, we conclude in Section 4.

2 The Lifetime Dependency Graph model

Definitions: Let us start with a few definitions and notations to be employed throughout. Let the sensor network be represented using graph SN where, $S = \{s_1, s_2, \ldots, s_n\}$ is the set of sensors, and an edge between sensor s_i and s_j exists if the two sensors are in communication range of each other. Let the set of targets be $T = \{t_1, t_2, \ldots, t_m\}$. We consider the problem of covering a stationary set of targets. This can easily be translated into the area coverage problem by mapping the area to a set of points which need to be covered [10, 11]. In addition to this, we define the following notation:

• $b(s)$: The battery available at a sensor s.

• $T(s)$: The set of targets in the sensing range of sensor s.

• $N(s, k)$: The neighbors of sensor s at k or fewer communication hops (including s).

• Cover C: Cover $C \subseteq S$ to monitor targets in T is a minimal set of sensors such that each target $t \in T$ has a nearby sensor $s \in C$ which can sense t, i.e., $t \in T(s)$.

• $lt(C)$: Maximum lifetime of a cover C is $lt(C) = min_{s \in C} b(s)$.

• $lt(t_i)$: The lifetime of a target $t_i \in T$ is given by $lt(t_i) = \sum_{\{s | t_i \in T(s)\}} b(s)$.

• Bottleneck Sensor: Bottleneck sensor s of cover C is the sensor $s \in C$ with minimum battery, i.e., it is the sensor s that upper bounds $lt(C)$.

• Bottleneck Target (t_{bot}): The target with the smallest lifetime $lt(t_{bot})$.

• Lifetime of a schedule of covers: We can view the set of currently active sensors as a cover C_i that is used for some length of time l_i.Given a schedule of covers of the form, $(C_1, l_1), (C_2, l_2), ..., (C_r, l_r)$. The lifetime of this schedule is given by $\sum_{i=1}^{r} l_i$.

• *OPT*: The optimal schedule of covers that achieves the maximum lifetime. Note that this includes both the covers and their corresponding time periods.

The Lifetime Dependency (LD) Graph [9]: The Lifetime dependency graph $LD = (V, E)$ where V is the set of all possible covers to monitor targets in T and and two covers C and C' are joined by an edge in E if and only if $C \cap C' \neq \emptyset$.

The LD graph effectively captures the dependency between two cover sets by representing their intersection by the edge between them. Further, we define,

• $w(e)$: Weight of an edge e between covers C and C' is $w(e) = min_{s \in C \cap C'} b(s)$.

• $d(C)$: Degree of a node or cover C is $d(C) = \Sigma_{e\ incident\ to\ C}\ w(e)$.

The reasoning behind this definition of the edge weight comes from considering a simple two-node LD graph with two covers C_1 and C_2 sharing an edge e. The lifetime of the graph is upper bounded by $min(lt(C_1) + lt(C_2), w(e))$. Similarly, the reasoning behind the definition of the degree of a cover C is that by summing the weights of all the edges incident on the cover C, we are getting a measure of the impact it would have on all other covers with which it shares an edge.

Basic Algorithmic Framework: Our distributed algorithms consist of an initial setup phase followed by rounds of predetermined duration during which sensors negotiate with their neighbors to determine their sense/sleep status.

Setup: In the setup phase, each sensor s communicates with each of its neighbor $s' \in N(s, 1)$ exchanging battery levels $b(s)$ and $b(s')$, and the targets covered $T(s)$ and $T(s')$. Then it finds all the local covers using the sensors in $N(s, 1)$ for the target set being considered. The latter can be solely $T(s)$ or could also include $T(s')$ for all $s' \in N(s, 1)$. It then constructs the local LD graph $LD = (V, E)$ over those covers, and calculates the degree $d(C)$ of each cover $C \in V$ in the graph LD. Note that the maximum number of covers that each sensor constructs is a function of the number of neighbors and the number of local targets it has. Both of these are relatively small for most graphs (but theoretically is exponential in the number of targets).

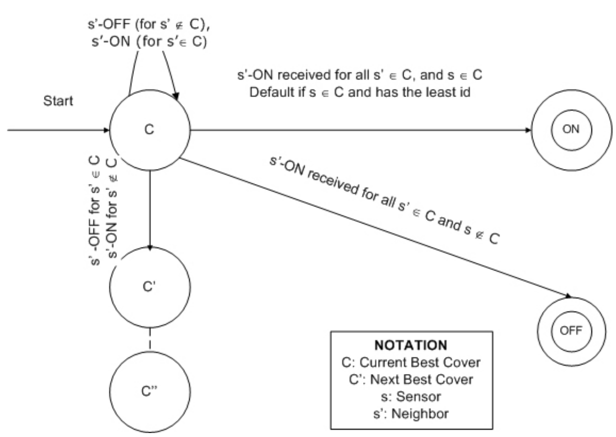

Figure 1: The state transitions to decide the sense/sleep status

Prioritize and Negotiate solutions: Once the LD graph has been constructed by each sensor, it needs to decide which cover to use. In order to do this, a *priority function* can be defined to prioritize the local covers. We base the priority of cover C on its degree *d(C)*. A lower degree is better since this corresponds to a smaller impact on other covers. Note that the priority function is computed at the beginning of every round by exchanging current battery levels among neighbors since the degrees may have changed from the previous round.

After calculating the priority function, the goal is to try and satisfy the highest priority cover. However, a cover comprises of multiple sensors and if one of these switches off, this cover cannot be satisfied. Hence, each sensor now uses the automaton in Fig. 1 to decide whether it can switch off or if it needs to remain on. The automaton starts with every sensor s in its highest priority cover C. The sensor s keeps trying to satisfy this cover C and eventually if the cover C is satisfied, then s switches on if $s \in C$ else s switches off. If a cover C cannot be satisfied, then the sensor s transitions to its next best priority cover C', C'' and so on, until a cover is satisfied.

The number of covers in the local LD graph is given by $O(\Delta^{\tau})$, where $\Delta = max_{s \in S}|N(s,1)|$ and $\tau = max_{s \in S}|T(s)|$ [9]. In practice, the number of covers in the local LD graph is small in practice, since Δ and τ are relatively small.

We simulated this Degree-Based heuristic along with a few of its variants over a range of sensor networks and compared the lifetime of their schedules with the current state-of-art algorithms, LBP [2] and DEEPS [3]. Our preliminary results showed an improvement of 10-20% in network lifetimes over others, while maintaining the same communication complexity.

In [6] we posed the question of what an imagined optimal schedule OPT might do with this LD graph. We were able to prove some properties that covers in the OPT sequence must exhibit. Based on these properties, we have designed algorithms which choose the covers that exhibit these OPT schedule like properties. We present three new heuristics - *Sparse-OPT* based on the sparseness of connectivity among covers in OPT, *Bottleneck-Target* based on selecting covers that optimize the use of sensors covering local bottleneck targets, and *Even-Target-Rate* based on trying to achieve an even burning rate for all targets. These heuristics are at a higher level and operate on top of degree-based heuristics to prioritize the local covers. Our experiments show an improvement in lifetime of 10-15% over the simple degree based heuristic and 25-30% over competing work in [2, 3] and 35% improvement for a two-hop version over the two-hop algorithm of [3]. The reader is referred to [6] for more details.

3 Results

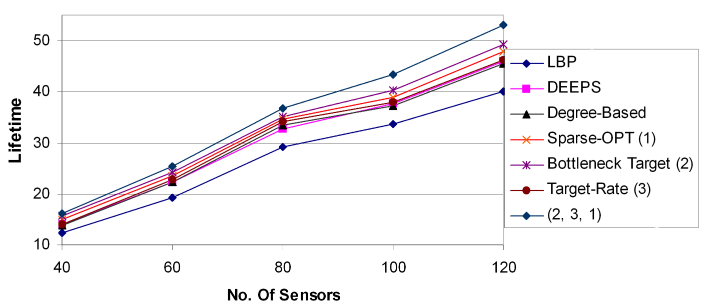

Figure 2: Lifetime with 25 Targets [6]

In order to compare the algorithm against LBP, DEEPS, and our previous work, we use the same experimental setup and parameters as employed in [2]. We carry out all the simulations using C++. For the simulation environment, a static wireless network of sensors and targets scattered randomly in $100m \times 100m$ area is considered. We conduct the simulation with 25 targets randomly deployed, and vary the number of sensors between 40 and 120 with an increment of 20 and each sensor with a fixed sensing range of $60m$.

For these simulations, we use the linear energy model wherein the power required to sense a target at distance d is proportional to d. We show here a snapshot of our results for the OPT-based heuristics as compared to LBP [2] and DEEPS [3]. More extensive simulation results and implementation details can be found in [6, 9]. Due to space constraints, we only show a representative sample in Fig. 2. As can be seen from the figure, among the three heuristics, the *Bottleneck-Target* heuristic performs the best giving about 10-15% improvement in lifetime over our previous Degree-Based heuristic and about 25-30% over LBP and DEEPS.

4 Conclusion

In this extended abstract, we provide a summary of our work on distributed algorithms for maximizing the lifetime of WSNs. As opposed to existing distributed solutions which are largely greedy in nature, we present a new way of approaching this problem. We introduce the lifetime dependency (LD) graph model for capturing the dependency that exists between different cover sets that share some sensors in common. We also provided an overview of different heuristics based on this graph along with some representative simulation results.

References

[1] Z. Abrams, A. Goel, and S. Plotkin. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. *Third International Symposium on Information Processing in Sensor Networks*, pages 424–432, 2004.

[2] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky. Efficient energy management in sensor networks. *In Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing*, 2005.

[3] Dumitru Brinza and Alexander Zelikovsky. Deeps: Deterministic energy-efficient protocol for sensor networks. *Proceedings of the International Workshop on Self-Assembling Wireless Networks (SAWN)*, pages 261–266, 2006.

[4] Mihaela Cardei and Ding-Zhu Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11:333–340(8), 2005.

[5] Akshaye Dhawan and Sushil K. Prasad. A distributed algorithmic framework for coverage problems in wireless sensor networks. *Procs. Intl. Parallel and Dist. Processing Symp. Workshops (IPDPS), Workshop on Advances in Parallel and Distributed Computational Models (APDCM)*, pages 1–8, 2008.

[6] Akshaye Dhawan and Sushil K. Prasad. Energy efficient distributed algorithms for sensor target coverage based on properties of an optimal schedule. In *To Appear, HiPC: 15th International Conference on High Performance Computing*, 2008.

[7] Akshaye Dhawan and Sushil K. Prasad. A distributed algorithmic framework for coverage problems in wireless sensor networks. *To Appear in International Journal of Parallel, Emergent and Distributed Systems(IJPEDS)*, 2009.

[8] Jun Lu and T. Suda. Coverage-aware self-scheduling in sensor networks. *18th Annual Workshop on Computer Communications (CCW)*, pages 117–123, 2003.

[9] Sushil K. Prasad and Akshaye Dhawan. Distributed algorithms for lifetime of wireless sensor networks based on dependencies among cover sets. In *HiPC: 14th International Conference on High Performance Computing, LNCS 4873*, pages 381–392, 2007.

[10] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. *IEEE International Conference on Communications (ICC)*, pages 472–476 vol.2, 2001.

[11] Di Tian and Nicolas D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *WSNA: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 32–41, New York, NY, USA, 2002. ACM.