Scheduling Battery-Powered Sensor Networks for Minimizing Detection Delays

Aron Laszka, Waseem Abbas, and Xenofon Koutsoukos

***Abstract*— Sensor networks monitoring spatially distributed physical systems often comprise battery-powered sensor devices. To extend lifetime, battery power may be conserved using *sleep scheduling*: activating and deactivating some of the sensors from time to time. Scheduling sensors with the goal of maximizing average coverage, which is the average fraction of time for which each monitoring target is covered by some active sensor, have been studied extensively. However, many applications also require time-critical monitoring in the sense that one has to minimize the average delay until an unpredictable change or event at a monitoring target is detected. In this letter, we study the problem of sleep scheduling sensors to minimize the average delay in detecting such time-critical events in the context of monitoring physical systems that can be modeled using graphs, such as water-distribution networks. We provide a game-theoretic solution that computes schedules with near optimal average delays. We illustrate that schedules that optimize average coverage may result in large average detection delays, whereas schedules minimizing average detection delays using our proposed scheme also result in near optimal average coverage.**

***Index Terms*— Sensor network, sleep scheduling, battery power, potential game, system lifetime, water-distribution network.**

I. Introduction

The lifetime of battery-powered sensing devices can be significantly increased by scheduling their active times. While sleep scheduling can significantly increase lifetime, it may also decrease monitoring coverage since some parts of the system may not be monitored continuously by active sensors. A number of research efforts have studied schedules that maximize the average fraction of time for which each target is monitored by at least one sensor, often referred to as the average coverage (e.g., [1]–[4]). The average coverage formulation, though useful in many scenarios, ignores the average detection delay, that is the average time until an event is first observed by an active sensor. This is a significant concern in the detection of time-critical events as large detection delay could accumulate the potential losses. For example, a water leakage can be detected by sensors after a pipe burst, but the detection delay needs to be minimized to reduce physical damage.

In this letter, we study schedules that minimize the average delay until a random event at a monitoring target is first detected by some sensor. First, we show that the optimal scheduling problem to minimize average detection delay is NP-hard. Second, using game theory concepts, we formulate the problem as a *potential game*. Third, using the binary log-linear learning algorithm [5] to solve the potential game, we give a near optimal solution to the scheduling problem. We also analyze the special case of random scheduling in random geometric networks. Finally, we illustrate our approach on a benchmark water distribution network and random geometric networks. Our numerical evaluation exhibits that schedules that minimize average detection delays also achieve near optimal average coverage, whereas the ones that optimize average coverage can result in large average detection delays.

Next, we briefly review and compare our work with some of the most relevant works in this area. Guo *et al.* [6] study sleep scheduling for critical event monitoring in wireless sensor networks assuming that sensor nodes are equipped with passive event detection capabilities, which allow them to wake up in case of an event. In contrast, we do not assume that sensors have such capabilities. Yang and Vaidya [7] consider end-to-end communication delay in sensor networks. Here, we consider a different aspect, the delay until detection. Xiao *et al.* [8] study randomized scheduling algorithms and consider detection delay in the context of intrusions that last only for a fixed period of time. In addition to randomized schedules, we also study the problem of finding optimal schedules. Further, we consider events that have permanent effect instead of fixed-length intrusions. Bhatt and Datta [9] study the problem of deploying audio and video sensors in a two-tier architecture. In contrast, we study the problem of sleep-scheduling sensors in a general model. Premkumar and Kumar [10] assume that sensors are collocated, i.e., each sensor covers the same region. On the other hand, we assume that each sensor may detect an arbitrary subset of the events.

II. System Model

The physical network monitored by the sensor devices is represented by a graph $G(V, E)$. The sensors S are located at the nodes of the network, that is, $S \subseteq V$. The targets Y that have to be monitored are a subset of nodes and edges of the network, that is, $Y \subseteq (V \cup E)$. The distance $d(s, v)$ between two nodes s and v is defined as the number of hops on the shortest path between them (and infinite if there exists no path). The distance $d(s, e)$ between a node s and an edge $e = (u, v)$ is defined as $\max\{d(s, u), d(s, v)\}$. A sensor $s \in S$ can monitor any target $y \in Y$ that is at most ε distance from the sensor s, where ε is referred to as the range of the sensor. The set of targets within the range of sensor s defines the neighborhood of s denoted by $N(s)$.

We assume that time is divided into T time slots, and each sensor device can be active for at most B of these slots. A schedule for activating and deactivating sensors is represented as a vector of sets $\mathbf{S} = (S_1, \ldots, S_T)$, where set $S_t \subseteq S$ is the subset of sensors that are active during time slot

Manuscript received October 28, 2016; revised December 11, 2016; accepted December 27, 2016. Date of publication January 5, 2017; date of current version April 7, 2017. This work was supported in part by the National Science Foundation (CNS-1238959) and by the Air Force Research Laboratory (FA 8750-14-2-0180). The associate editor coordinating the review of this letter and approving it for publication was M. Erol-Kantarci.

The authors are with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37212 USA (e-mail: aron.laszka@vanderbilt.edu; waseem.abbas@vanderbilt.edu; xenofon.koutsoukos@vanderbilt.edu).

Digital Object Identifier 10.1109/LCOMM.2017.2648782

$t \in \{1, \ldots, T\}$. Based on the battery constraint B, we say that a schedule $\boldsymbol{S}$ is *feasible* if and only if

$$\forall s \in S : \quad \left| \left\{ t \in \{1, \cdots, T\} \,\middle|\, s \in S_t \right\} \right| \leq B. \tag{1}$$

Let $\Gamma_y(\boldsymbol{S}) \subseteq \{1, \cdots, T\}$ be the subset of time slots in which target y is monitored by at least one sensor in schedule $\boldsymbol{S}$. The number of time slots until an event (e.g., failure) at target y in time slot t is first detected by schedule $\boldsymbol{S}$, denoted by $\mathcal{D}(\boldsymbol{S}, y, t)$, is

$$\mathcal{D}(\boldsymbol{S}, y, t) = \min\left\{ j \,\middle|\, \left[\left(j \in \Gamma_y(\boldsymbol{S})\right) \vee (j = T+1) \right] \wedge (j \geq t) \right\} - t.$$

A. Problem Formulation

Using the above definition, we can express the *average detection delay* until an event at a random target in a random time slot is first detected by schedule $\boldsymbol{S}$, denoted by $\mathcal{D}(\boldsymbol{S})$, as

$$\mathcal{D}(\boldsymbol{S}) = \frac{1}{T} \sum_{i=1}^{T} \frac{1}{|Y|} \sum_{y \in Y} \mathcal{D}(\boldsymbol{S}, y, i). \tag{2}$$

The objective is to find a feasible schedule $\boldsymbol{S}$ (as defined in (1)) minimizing the average detection delay, that is,

$$\min_{\textit{feasible } \boldsymbol{S}} \mathcal{D}(\boldsymbol{S}). \tag{3}$$

We note here that optimizing schedules for average detection delay (3) is essentially different from finding schedules that maximize the average number of time slots for which a random target remains monitored by some sensor, referred to as the average coverage (e.g., see [1], [2], [4], [11]). For the case of average coverage, the objective is to maximize $\mathcal{C}(\boldsymbol{S})$ over all feasible schedules, where

$$\mathcal{C}(\boldsymbol{S}) = \frac{1}{|Y|} \sum_{y \in Y} \frac{|\Gamma_y(\boldsymbol{S})|}{T}. \tag{4}$$

III. Analysis

A. Computational Complexity

We begin our analysis by showing that finding an optimal schedule to minimize average detection delay is computationally hard. First, we formulate the problem of finding an optimal schedule as a decision problem.

Definition 1 (Minimum Average Detection Delay (MADD)): Given a graph $G = (V, E)$, a set of sensors $S \subseteq V$, a set of targets $Y \subseteq (V \cup E)$, a range ε, a network lifetime T, a battery supply B, and a threshold delay $\mathcal{D}^*$, determine if there exists a schedule $\boldsymbol{S}$ such that $\mathcal{D}(\boldsymbol{S}) \leq \mathcal{D}^*$.

Theorem 1: The MADD problem is NP-hard.

We prove the NP-hardness of finding an optimal schedule using a reduction from a known NP-hard problem, the 2-Disjoint Set Covers problem [11].

Definition 2 (2-Disjoint Set Covers): Given a set U and a collection $\mathcal{C}$ of subsets of U, determine whether the collection $\mathcal{C}$ can be partitioned into two disjoint set covers.

Proof Sketch: Given an instance $(U, \mathcal{C})$ of the 2-Disjoint Set Covers problem (2-DSC), we construct an instance $(G = (V, E), S, Y, \varepsilon, T, B, \mathcal{D}^*)$ of the MADD as follows:

Let the set of nodes be $V = U \cup \mathcal{C}$; the set of edges be $E = U \times \mathcal{C}$; the set of sensors be $S = \mathcal{C}$; and the set of targets be $Y = U$. Finally, let the network lifetime be $T = 2$; the battery supply be $B = 1$; the range be $\varepsilon = 1$; and the threshold delay be $\mathcal{D}^* = 0$.

Since we can clearly perform the above reduction in polynomial time, it remains to show that the 2-DSC problem has a solution if and only if the MADD problem does. First, assume that there exist two disjoint set covers $\mathcal{C}_1$ and $\mathcal{C}_2$. Then, $\boldsymbol{S} = (\mathcal{C}_1, \mathcal{C}_2)$ is a feasible schedule and its delay $\mathcal{D}(\boldsymbol{S})$ is zero since every target is covered by a sensor in both time slots. Second, assume that there exists a schedule $\boldsymbol{S}$ with delay $\mathcal{D}(\boldsymbol{S}) = 0$. Then, for similar reasons, S_1 and S_2 are disjoint set covers, which concludes our proof. ■

Next, we present our approach to solving the scheduling problem using a game-theoretic setup.

B. Scheduling as a Potential Game

Here, we present a general solution for the scheduling problem to minimize average detection delay using game-theoretic concepts. In this direction, a particular class of non-cooperative games, known as *potential games*, are especially useful [5], [12]. A basic theme in potential games is that a single global function, known as the *potential function*, can express the incentive of all players to change their strategy. Consider a set of players P, and let $\mathcal{A}_p$ be the action set of a player p, and $a_p, a'_p \in \mathcal{A}_p$. Moreover, let $\boldsymbol{a}_{-p}$ represent collectively the actions of players in $P \setminus \{p\}$; $u_p(a_p, \boldsymbol{a}_{-p})$ represent the utility of player p as a result of an action a_p; and $\phi(a_p, \boldsymbol{a}_{-p})$ be the value of potential function as a result of players' actions. Then, for every $p \in P$, $a_p, a'_p \in \mathcal{A}_p$, and every $\boldsymbol{a}_{-p}$, the following holds in potential games:

$$u_p(a_p, \boldsymbol{a}_{-p}) - u_p(a'_p, \boldsymbol{a}_{-p}) = \phi(a_p, \boldsymbol{a}_{-p}) - \phi(a'_p, \boldsymbol{a}_{-p}) \tag{5}$$

For potential games, algorithms such as log-linear-learning (LLL) [12] and binary log-linear-learning (BLLL) [5] converge to action profiles that maximize the potential function with arbitrary high probability, thus achieving the global objective. We utilize the framework of potential games to solve the scheduling problem. We first formulate the scheduling problem as a potential game by appropriately selecting the utility functions of players, which are sensors, and the potential function that represent the overall objective of minimizing the average detection delay as in (3).

1) Scheduling Game: For an instance of the scheduling problem, we define the Scheduling Game $(P, \mathcal{A}, \boldsymbol{u})$ as follows:

- The set of players P is the set of sensors S.
- For each player $s \in P$, the action set $\mathcal{A}_s$ is the set of all subsets of $\{1, \ldots, T\}$ with cardinality B, and $\mathcal{A} = (\mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_{|P|})$ is the players' joint action space. For each $\boldsymbol{a} \in \mathcal{A}$, we define the corresponding schedule $\boldsymbol{S}$ as the one where $S_t = \{s \in S \,|\, t \in a_s\}$ (i.e., in schedule $\boldsymbol{S}$, each sensor s is active in time slots a_s).
- For each player $s \in P$, we define the utility of an action $a_s \in \mathcal{A}_s$ as the overall reduction in delays until events occurring at the neighbors of s in all different time slots are first detected. This reduction in delays is possible only because of the action a_s. More precisely,

Algorithm 1 Binary Log-Linear Learning

```
Initialization: Pick a small $\tau \in \mathbb{R}_+$, an $\boldsymbol{a} \in \mathcal{A}$, and a total number of iterations.
While $i \leq$ number of iterations do
    Randomly pick a sensor $s \in S$ and action $a'_s \in \mathcal{A}_s$.
    Compute $V_\tau = \frac{e^{\frac{1}{\tau}u_s(a'_s,\boldsymbol{a}_{-s})}}{e^{\frac{1}{\tau}u_s(a'_s,\boldsymbol{a}_{-s})}+e^{\frac{1}{\tau}u_s(a_s,\boldsymbol{a}_{-s})}}$.
    Set $a_s \leftarrow a'_s$ with probability $V_\tau$.
    $i \leftarrow i + 1$
End While
```

if $\boldsymbol{S} = (S_1, \cdots, S_T)$ is the schedule given by $\boldsymbol{a}$, let $\tilde{S}_j = S_j \setminus \{s\}$ and $\tilde{\boldsymbol{S}} = (\tilde{S}_1, \cdots, \tilde{S}_T)$. Note that $\tilde{\boldsymbol{S}}$ is the same schedule as $\boldsymbol{S}$ with the only difference that sensor s is not active in any time slot. We can now formally define $u_s(a_s, \boldsymbol{a}_{-s})$ as follows:

$$u_s(a_s, \boldsymbol{a}_{-s}) = \frac{1}{T|Y|} \sum_{y \in N(s)} \sum_{i=1}^{T} \Big(\mathcal{D}(\tilde{\boldsymbol{S}}, y, i) - \mathcal{D}(\boldsymbol{S}, y, i) \Big) \tag{6}$$

Proposition 1: A Scheduling Game $(P, \mathcal{A}, \boldsymbol{u})$ is a potential game with the potential function $\phi(\boldsymbol{a}) = -\mathcal{D}(\boldsymbol{S})$.

Proof: For any $s \in S$ and a schedule $\boldsymbol{S}$ defined by $\boldsymbol{a} = (a_s, \boldsymbol{a}_{-s}) \in \mathcal{A}$, we have

$$\phi(\boldsymbol{a}) = \frac{-1}{T|Y|} \sum_{i=1}^{T} \left[\sum_{y \in N(s)} \mathcal{D}(\boldsymbol{S}, y, i) + \sum_{y \in Y \setminus N(s)} \mathcal{D}(\boldsymbol{S}, y, i) \right].$$

Note that for any other $\boldsymbol{S}'$ defined by $\boldsymbol{a}' = (a'_s, \boldsymbol{a}_{-s})$ and i, $\sum\limits_{y \in Y \setminus N(s)} \mathcal{D}(\boldsymbol{S}, y, i) = \sum\limits_{y \in Y \setminus N(s)} \mathcal{D}(\boldsymbol{S}', y, i)$, thus, we get

$$\phi(\boldsymbol{a}) - \phi(\boldsymbol{a}') = \frac{1}{T|Y|} \sum_{i=1}^{T} \sum_{y \in N(s)} \left(\mathcal{D}(\boldsymbol{S}', y, i) + \mathcal{D}(\boldsymbol{S}, y, i) \right). \tag{7}$$

Performing the same for the utilities gives

$$u_s(a_s, \boldsymbol{a}_{-s}) - u(a'_s, \boldsymbol{a}_{-s}) = \phi(\boldsymbol{a}) - \phi(\boldsymbol{a}').$$

■

2) Solution of the Scheduling Game: It is well known that in the case of exact potential games, if players select actions based on log-linear learning models, the dynamics converge to a specific set of Nash equilibria, namely the set of potential maximizers (e.g., see [5], [12], [13]). The basic idea in these models is to have noisy best-response dynamics that allow occasional selection of suboptimal actions by the players. Here, we utilize binary log-linear learning [5] to solve the scheduling problem.

Here, τ is the noise parameter that determines the selection of suboptimal actions by players. As $\tau \to 0$, players select the best response with arbitrary high probability. As $\tau \to \infty$, players select any action from the action space with equal probability. We also note here that using BLLL algorithm, schedules can be computed off-line in the design phase. Next, we analyze the special case of random scheduling in random geometric networks.

C. Random Schedules in Random Geometric Networks

In a random schedule, each sensor selects B time slots uniformly at random, in which it remains active. Here, we study the average detection delay of random schedules in random geometric graphs, which are widely used to model sensor networks with random deployment of sensors.

Proposition 2: Let $G(V, E)$ be a random geometric graph induced by a Poisson point process with constant density λ, in which two nodes are adjacent if the Euclidean distance between them is at most r. If $S = Y = V$, $\varepsilon = 1$, and every node is active in a set of B time slots chosen randomly at uniform from the set $\{1, \ldots, T\}$, then the average detection delay for a random node is

$$\sum_{k=1}^{T-B} \frac{k\mathcal{P}(k)}{e^{\lambda\pi r^2}} \left[e^{\lambda\pi r^2 \mathcal{P}(k)} - \left(\frac{T-k-B}{T-k} \right) e^{\lambda\pi r^2 \mathcal{P}(k+1)} \right], \tag{8}$$

where, $\mathcal{P}(x) = \prod_{i=0}^{x-1} \left(1 - \frac{B}{T-i}\right)$ for a given B and T.

Proof: The probability that a node is inactive in a time slot is $\left(1 - \frac{B}{T}\right)$. The probability that a node is inactive for at least $k \leq T - B$ consecutive time slots is $\prod_{i=0}^{k-1} \left(1 - \frac{B}{T-i}\right) = \mathcal{P}(k)$. Similarly, the probability that a node is inactive for $k \leq T - B$ consecutive time slots but is active in the following time slot is $\left(\frac{B}{T-k}\right) \mathcal{P}(k)$. Moreover, the probability that node u and all of its neighbors $N(u)$ are inactive for $k \leq T - B$ consecutive time slots but at least one of them is active in the following time slot is

$$\left[1 - \left(1 - \frac{B}{T-k} \right)^{|N(u)|+1} \right] \mathcal{P}(k)^{|N(u)|+1}. \tag{9}$$

Thus, the expected detection delay for node u given that it has $|N(u)|$ neighbors is

$$\sum_{k=1}^{T-B} k \left[1 - \left(1 - \frac{B}{T-k} \right)^{|N(u)|+1} \right] \mathcal{P}(k)^{|N(u)|+1}. \tag{10}$$

The probability that node u has j neighbors is $\Pr[|N(u)| = j] = \frac{(\lambda\pi r^2)^j e^{-\lambda\pi r^2}}{j!}$. Therefore, the average expected detection delay is

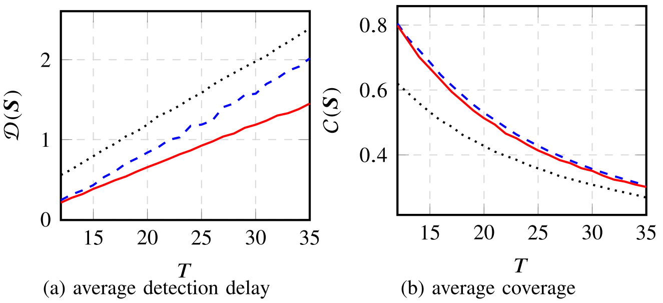
$$\sum_{j=0}^{\infty} \frac{\left(\lambda\pi r^2\right)^j}{j! e^{\lambda\pi r^2}} \sum_{k=1}^{T-B} k \left[1 - \left(1 - \frac{B}{T-k} \right)^{j+1} \right] \mathcal{P}(k)^{j+1}. \tag{11}$$

After simplifying and observing that $\left(1 - \frac{B}{T-k}\right) \mathcal{P}(k) = \mathcal{P}(k+1)$, we get the desired result. ■

IV. Numerical Results

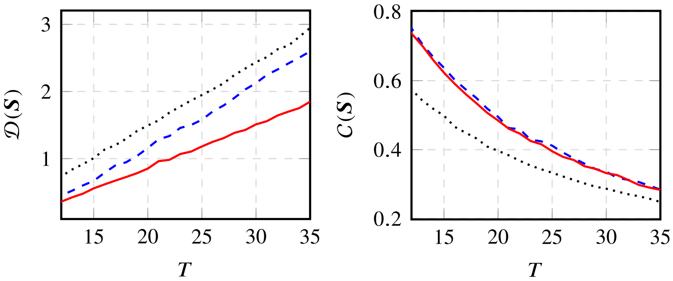
We evaluate our algorithms on two types of networks; a benchmark water-distribution network (see [14] for details), and random geometric networks. We compare our approach to two baselines: (i) schedules maximizing average coverage and (ii) random schedules (i.e., uncoordinated monitoring).

The water-distribution network considered here has been extensively used in research related to sensor placement in water-distribution networks. The network has 126 pipes, 168 links, one reservoir, two pumps, and two storage tanks. For our scheduling problem, we let $S = V$, $\varepsilon = 2$, and $Y = E$.



(a) average detection delay (b) average coverage

Fig. 1. (a) Average detection delay and (b) average coverage as functions of T in the water-distribution network with schedules computed using BLLL for minimizing delay (solid line), BLLL maximizing average coverage (dashed line), and chosen randomly (dotted line).



(a) average detection delay (b) average coverage

Fig. 2. (a) Average detection delay and (b) average coverage as functions of T in random geometric graphs.

Moreover, we assume $B = 2$ in all the simulations, which means that all three schemes have the same energy cost.

For random geometric networks, we select a network of 100 nodes deployed at random in a unit square. Each node can monitor a target that is at a Euclidean distance of at most r from the node, which is referred to as the sensing radius of the node. For evaluation, we consider both S and Y to be the set of nodes, and $r = 0.12$. We note that the range ε of the sensor nodes, which is measured in number of hops, is equal to one in the resulting graphs.

The results are plotted in Figures 1 and 2 for (a) average detection delay $\mathcal{D}(\boldsymbol{S})$ and (b) average coverage $\mathcal{C}(\boldsymbol{S})$ obtained from schedules computed using BLLL minimizing delay (solid line), BLLL maximizing coverage[1] (dashed line), and randomly (dotted line). For random schedules and random geometric graphs, each point on the plots is an average of fifty instances. The value of τ in BLLL for minimizing delay and maximizing coverage is chosen to be 10^{-4}. For each run of BLLL, 5000 iterations are performed.

The plots show that the $\mathcal{D}(\boldsymbol{S})$ of schedules computed using BLLL for minimizing delay are significantly lower than the ones obtained from random schedules as well as from the schedules for near optimal average coverage. For instance, as compared to random schedules, the average detection delays are reduced by 39% upto 62% in the water network, and by 37% upto 52% in random geometric networks (considering $B = 2, T = \{12, \cdots, 35\}$) due to the proposed scheme. Similarly, as compared to schedules that give near optimal average coverage using BLLL, the average detection delays in the proposed scheme are reduced by 11% upto 28% in the water network, and by 16% upto 30% in random geometric networks. At the same time, average coverage from schedules computed using BLLL for minimum delay is only slightly lower (i.e., by 0.7% upto 4% in the water network, and by only 0.1% upto 2.6% in random geometric networks) than the near optimal average coverage computed using BLLL for maximum $\mathcal{C}(\boldsymbol{S})$.

[1] The problem of finding schedules that maximize average coverage can also be formulated as a potential game, and thus, BLLL algorithm for maximizing coverage indeed gives schedules with near optimal average coverage.

V. Conclusion

In this letter, we studied the problem of sleep-scheduling battery-powered sensors to minimize detection delay. After formalizing the scheduling problem for physical systems that can modeled as graphs, we provided a solution for finding schedules based on a potential-game formulation of the problem. Finally, we evaluated the proposed solution numerically using random geometric graphs and a real-world water-distribution network. Our numerical results show that schedules that minimize average detection delays also achieve near optimal average coverage, whereas, the ones that optimize average coverage can result in large average detection delays.

References

[1] C. Wang, M. T. Thai, Y. Li, F. Wang, and W. Wu, “Optimization scheme for sensor coverage scheduling with bandwidth constraints,” *Optim. Lett.*, vol. 3, no. 1, pp. 63–75, Jan. 2009.

[2] A. Rossi, A. Singh, and M. Sevaux, “Column generation algorithm for sensor coverage scheduling under bandwidth constraints,” *Networks*, vol. 60, no. 3, pp. 141–154, Oct. 2012.

[3] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” *Ad Hoc Netw.*, vol. 7, no. 3, pp. 537–568, May 2009.

[4] W. Abbas, A. Laszka, Y. Vorobeychik, and X. Koutsoukos. (Aug. 2016). “Scheduling resource-bounded monitoring devices for event detection and isolation in networks.” [Online]. Available: https://arxiv.org/abs/1608.07329

[5] J. R. Marden and J. S. Shamma, “Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation,” *Games Econ. Behavior*, vol. 75, no. 2, pp. 788–808, Jul. 2012.

[6] P. Guo, T. Jiang, Q. Zhang, and K. Zhang, “Sleep scheduling for critical event monitoring in wireless sensor networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 2, pp. 345–352, Feb. 2012.

[7] X. Yang and N. H. Vaidya, “A wakeup scheme for sensor networks: Achieving balance between energy saving and end-to-end delay,” in *Proc. 10th IEEE Real-Time Embedded Technol. Appl. Symp. (RTAS)*, May 2004, pp. 19–26.

[8] Y. Xiao *et al.*, “Coverage and detection of a randomized scheduling algorithm in wireless sensor networks,” *IEEE Trans. Comput.*, vol. 59, no. 4, pp. 507–521, Apr. 2010.

[9] R. Bhatt and R. Datta, “A two-tier strategy for priority based critical event surveillance with wireless multimedia sensors,” *Wireless Netw.*, vol. 22, no. 1, pp. 267–284, Jan. 2016.

[10] K. Premkumar and A. Kumar, “Optimal sleep-wake scheduling for quickest intrusion detection using wireless sensor networks,” in *Proc. 27th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2008, pp. 1400–1408.

[11] M. Cardei and D.-Z. Du, “Improving wireless sensor network lifetime through power aware organization,” *Wireless Netw.*, vol. 11, no. 3, pp. 333–340, May 2005.

[12] L. E. Blume, “The statistical mechanics of strategic interaction,” *Games Econ. Behavior*, vol. 5, no. 3, pp. 387–424, Jul. 1993.

[13] A. Y. Yazicioglu, M. Egerstedt, and J. S. Shamma, “Communication-free distributed coverage for networked systems,” *IEEE Trans. Control Netw. Syst.*, to be published.

[14] A. Ostfeld *et al.*, “ The battle of the water sensor networks (BWSN): A design challenge for engineers and algorithms,” *J. Water Resour. Planning Manage.*, vol. 134, no. 6, pp. 556–568, 2008.