Distributed Lifetime-Maximized Target Coverage Game

LI-HSING YEN and CHE-MING LIN, National University of Kaohsiung
VICTOR C.M. LEUNG, The University of British Columbia

Wireless sensor nodes are usually densely deployed to completely cover (monitor) a set of targets. Consequently, redundant sensor nodes that are not currently needed in the covering task can be powered off to conserve energy. These sensors can take over the covering task later to prolong network lifetime. The coverage problem concerns picking up a set of working sensors that collectively meet the coverage requirements. The problem is complicated by the possibility that targets may have different coverage requirements while sensor nodes may have different amounts of energy. This paper proposes a game-theoretic approach to the coverage problem where each sensor autonomously decides its state with a simple rule based on local information. We give rigorous proofs to show stability, correctness, and efficiency of the proposed game. Implementation variants of the game consider specific issues such as game convergence time and different amounts of sensor energy. Simulation results show significant improvement in network lifetime by the proposed approach when compared with representative alternatives.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Design, Algorithms, Theory

Additional Key Words and Phrases: Coverage, energy management, game theory, network protocols, wireless sensor networks

ACM Reference Format:
Yen, L., Lin, C., and Leung, V.C.M. 2011. Distributed Lifetime-Maximized Target Coverage Game. ACM Trans. Sensor Netw. V, N, Article A (January YYYY), 23 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Wireless sensor nodes are electronic devices capable of collecting, storing, and processing environmental information, and communicating with other sensor nodes through wireless communications. Hundreds or thousands of wireless sensor nodes deployed in a region of interest comprise a wireless sensor network (WSN), where all members cooperatively monitor the whole region, barriers, or targets (points) within the region [Cardei and Wu 2004]. In this paper, we are interested in complete coverage of a set of monitored targets at all time, a requirement known as *target coverage*. In this problem, a sensor node is said to *cover* some target if that target is under the surveillance of the sensor. Since sensor nodes are usually densely deployed, covering all targets demands only a small subset of sensor nodes. The rest can enter sleep mode to conserve precious energy. These sensors can be powered on later to take over the covering task and thus prolong network operation time. Determining whether individual sensors should be ac-

This work is supported by the National Science Council, Taiwan, under grant NSC 100-2628-E-390-002-MY2.

Author's addresses: L.-H. Yen and C.-M. Lin, Department of Computer Science and Information Engineering, National University of Kaohsiung, Taiwan; V.C.M. Leung, Department of Electrical and Computer Engineering, The University of British Columbia, Canada.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

tive or in sleep mode while retaining the desired target coverage is termed a *coverage problem*. All the active sensor nodes constitute a *set cover*. If set covers are formed in a periodic manner, the coverage problem becomes a work scheduling or sleep scheduling problem.

The coverage problem faces the situation that not all sensors in a WSN are of the same importance, as not every sensor covers the same set of targets and/or has the same amount of energy. Furthermore, in some applications, the number of sensors required to cover a target may differ from target to target [Gu et al. 2007; Chaudhary and Pujari 2009]. In the literature, the problem under discussion has been formulated in various ways. Most research consider uniform coverage requirement (1-coverage or k-coverage) and assume uniform sensor energy. A few studies take account of non-uniform coverage requirement (Q-coverage) or non-uniform sensor energy, but not both. Many variant forms of the problem (which demand a minimal number of active sensors) have been proven NP-hard, for which many heuristics have been proposed. However, little has been done on the coverage problem with the consideration of different amounts of sensor energy and coverage requirements.

This paper tackles the coverage problem with the goal to develop a feasible distributed mechanism for energy-constrained sensor devices. Sensor devices are typically numerous in a WSN and locations of sensors cannot be engineered due to practical constraints. Therefore, a centralized work scheduling algorithm that demands full knowledge of coverage relationship between sensors and covered regions/targets would not scale well. Distributed schemes based on special structure of sensor nodes such as grids or clusters incur additional cost in planning or managing such a structure. Furthermore, sensor devices generally have a limited computation capability due to the needs to reduce the hardware complexity and power consumption of these devices. Computation-intensive methods such as linear programming and genetic algorithms are therefore not favored.

The proposed distributed approach is based on game theory and does not demand any special organization of sensors into a specific topology such as grid or cluster. In this approach, sensor devices as autonomous participants seek their own interest and need only status information of neighboring sensors that have overlapping coverage with them to make their own decisions. Although neighboring sensors have conflicting interest and there is no centralized mechanism to coordinate actions among sensors, we are able to prove the stability, correctness, and efficiency of the proposed approach under the game theory framework. This game design demands only simple arithmetic computations and is therefore practical for implementation in hardware-constrained sensor devices. Besides the basic game design, utility-aware and power-aware realizations of the game in WSNs are proposed to shorten game convergence time and to balance sensor energy consumption, respectively. A hybrid implementation that considers both utility and power is then investigated. Simulation results indicate that the proposed approach provides a significant improvement in network lifetime when compared with existing greedy heuristics.

The remainder of this paper is organized as follows. Background knowledge and related work are presented in Section 2. Section 3 presents the proposed game-theoretic approach to the coverage problem with rigorous correctness proof. In Section 4, performance evaluations of the proposed approach are presented in comparisons with other alternatives. Section 5 concludes this paper.

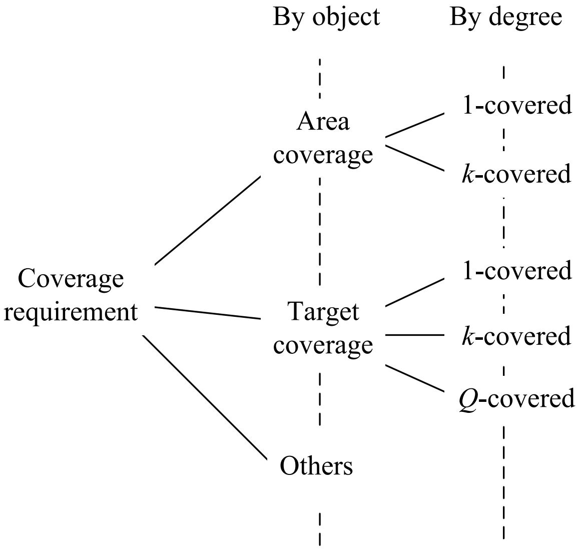

Fig. 1. Classifying coverage requirement definitions

2. BACKGROUNDS AND RELATED WORK

2.1. Backgrounds

It is required that any solution to the coverage problem be subject to some *coverage requirement*. Such requirement has a broad definition as the term *coverage* carries different meanings in the literature. The object of sensor coverage could be an area, a number of targets (points), or others. Different applications may demand different degrees of coverage from WSNs. The requested coverage degree is often expressed as "every object should be covered by at least one (1-covered) or $k > 1$ (k-covered) sensor nodes." When targets are of interest, requested coverage degrees may differ from one target to another, leading to the definition of *Q-coverage* [Gu et al. 2007]. Fig. 1 shows a brief classification of coverage requirement definitions.

Since more sensors than needed are usually deployed, it is not difficult to find a set cover that meets the coverage requirement. A greedy set-formation heuristic typically works by adding nodes into an initially-empty set cover on a node-by-node basis. In each round of the heuristic, the node with the highest remaining energy, the node that contributes the most coverage, or the node that covers the most critical object can be added into the set cover. However, finding the best node among all candidates a rank sorting based on global information, which is not easy to implement as a distributed protocol. A naive implementation may rely on one special sensor node to collect the necessary information from all others, form a set cover based on the collected information, and dispatch selected sensors. This creates a performance bottleneck as well as a single point of failure at that node.

Set covers can be *disjoint* or *non-disjoint*. When sensor nodes have uniform power, all nodes in a set cover exhaust their power at about the same time and all set covers operate for the same amount of time. In this case, finding the maximal number of disjoint set covers essentially maximizes network lifetime and corresponds to the optimal solution to the work scheduling problem. When sensors have different amounts of energy, not all set covers work for the same amount of time and other sensors may still have energy when the first node in the same set cover exhausts all its energy. The residual energy can be utilized if set covers are not disjoint. Consequently, a solution to

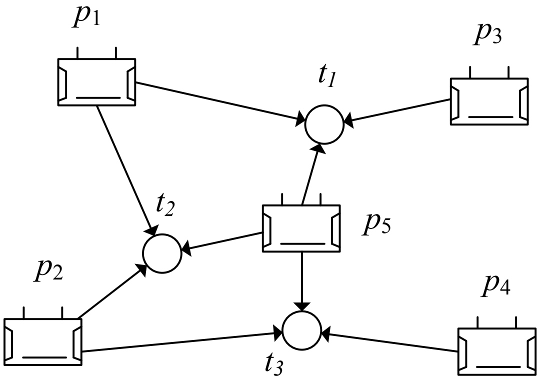

Fig. 2. An example where $\{p_1, p_2, p_3, p_4, p_5\}$ are sensors and $\{t_1, t_2, t_3\}$ are targets. An arrowed line from a sensor to a target represents the ability of the sensor to cover that target. All targets request 1-coverage.

the work scheduling problem that consists of disjoint set covers may not be optimal. In general, non-disjoint set covers are expected to produce longer network lifetime than disjoint set covers.

Figure 2 illustrates the difference between disjoint and non-disjoint set covers. If all sensors have uniformly one unit of energy, then the solution consisting of the maximal number of disjoint set covers (*i.e.*, $\{p_1, p_4\}$, $\{p_2, p_3\}$, and $\{p_5\}$) is the best solution in terms of network lifetime. The network lifetime is 3τ, where τ is the amount of time corresponding to dissipating one unit of energy. Suppose now both p_1 and p_2 have two units of energy while the others remain intact. The original solution yields the same network lifetime, while there exists a better solution that consists of four non-disjoint set covers (for example, $\{p_2, p_3\}$, $\{p_1, p_4\}$, $\{p_5\}$, and $\{p_1, p_2\}$).

Game theory provides a mathematical framework for the study of strategies in a competition where players have conflicting benefits or goals. A game consists of the following components: player set, strategy set, and utility functions. All competitors in a game comprise the *player set*. All feasible decisions of a player comprise the *strategy set* for the player. A *strategy profile* is a tuple of strategies, one from each player's strategy set. Each player can have a unique *utility function*, which returns the player's payoff (utility) with respect to a particular strategy profile. Players are usually selfish in the sense that the only goal of all the players is to maximize their own payoff.

In traditional sensor networks where all sensor nodes are deployed and controlled by one entity, a game-theoretic approach may simply represent a utility-based algorithm design. The utility itself need not have any physical meaning. In a participatory sensing framework where individual sensors may be controlled by different entities, the utilities should be connected to some type of payoffs in the physical world (such as payments or credits given to players/sensors) to motivate sensors to offer their coverage. In this case, how to prevent or identify fraud becomes an issue. The approach proposed in this paper is general in the sense that various meanings can be applied to the utilities in practice.

For the last decade, game theory has been used in resource/duty sharing problems in wireless networking environments. It has been used to model packets forwarding tasks among nodes in wireless ad hoc networks [Han et al. 2005; Félegyházi et al. 2006]. Another application of game theory is to analyze the competition for radio resources among terminals in a wireless data network [Yaïche et al. 2000; Han et al. 2005; 2007; Yen et al. 2011]. The problem of power control in Code Division Multiple Access (CDMA) wireless data systems has also been modeled as a non-cooperative power control game [Saraydar et al. 2002; Xiao et al. 2003; Rasti et al. 2009].

2.2. Related Work

Concerning area coverage, Chamam and Pierre [2007] proposed a greedy approach that finds set covers based on residual energy of sensors. The goal is to balance energy consumption among all set covers so as to prolong network lifetime. In [Simon et al. 2007], an approach to k-coverage was proposed where each sensor has its own sleep interval such that sleep intervals of sensors may vary in length. The length is set such that sensors with less energy or sensors that are in critical positions have longer sleep intervals than others. The idea is to conserve energy of these sensors until their participations become inevitable. Li and Gao [2008] studied work scheduling for k-coverage with a goal to maximize network lifetime. This problem is proven NP-hard by the authors, and two greedy heuristics were proposed for the formations of disjoint and non-disjoint set covers, respectively. The authors also considered adjusting sensing range to further conserve energy.

Cardei and Du [2005] studied target coverage under the constraint that every target should be 1-covered. Their approach partitions sensor nodes into disjoint set covers, each of which meets the coverage constraint. In this way, to maximize network lifetime is essentially to maximize the number of disjoint set covers. They showed that this problem is NP-complete, and proposed a heuristic based on an solution to the classic maximum-flow problem. This work was extended to consider non-disjoint set covers [Cardei et al. 2005], where the problem was proven NP-complete as well. For an efficient formation of non-disjoint set covers, the authors applied linear programming and greedy techniques.

Hefeeda and Bagheri's study [2007] aimed at area coverage, but they assumed dense sensor network such that covering all sensor locations approximates covering the whole area. The problem of selecting the minimum number of sensors to cover all sensor locations such that every location is k-covered was modeled by the authors as a problem of finding an optimal hitting set, and an approximation algorithm for the optimal hitting set problem was adapted for the coverage problem under consideration.

In [Fan et al. 2008], the authors discussed how to place a minimal number of sensors to make every target k-covered. The authors proposed an approach based on Computational Geometry and Combinatorics. This approach is unique in that the locations to place sensor nodes need not be exact.

Gu et al. [2007] are the first to define the Q-coverage problem and prove its NP-completeness. Although the problem can be converted into a linear programming problem, the difficulty lies in the nonexistence of a polynomial-time algorithm that is able to generate all possible set covers. The authors presented an approach that generates set covers randomly. A greedy heuristic proposed in [Chaudhary and Pujari 2009] prefers sensors with a higher energy level when forming a set cover for Q-coverage. When a set cover is found, a pruning technique is then applied as an attempt to further reduce the size of the set cover.

Comparatively little work has been done toward the scheduling problem under the framework of game theory. Zhu and Martinez [2009] defined a coverage game, which assumes visual sensors for which orientation and focal length of the camera can be adjusted. Consequently, the visual sensing area of a camera is directional with a finite angle of view and has a limited-range. By visually covering an area Δ, a sensor earns a profit proportional to the number of interesting events in Δ but inversely proportional to the number of all sensors that cover the same area. On the other hand, the cost of processing visual data (in terms of energy consumption) for a visual sensor is assumed proportional to the area of the region covered by it. The game has a goal to maximize overall payoff (profit minus cost). This game is a variation of the congestion games [Rosenthal 1973].

To the best of our knowledge, the closest related work to ours is [Ai et al. 2008], where the authors considered work schedule of sensor nodes under a game-theoretic framework. This work assumes that the lifetime of the whole network is known and given and consists of k duty periods. The problem is to arrange the work schedule of sensors to maximize average area coverage. In this model, every player (sensor) is required to select and join one of k duty periods. Thus all duty periods comprise a strategy set for a player. The coverage level in a duty period is defined by the size of the area collectively covered by all active sensors in that period. The payoff associated with a player's strategy corresponds to the additional coverage level resulting from the sensor's participation in the chosen duty period. The authors proved that the proposed game is a potential game [Monderer and Shapley 1996] as well as a congestion game.

The main issue associated with [Ai et al. 2008] comes from the assumption that k is known and given. The value of k in fact trades the lifetime of the whole network with average coverage level. If k is too large, the average coverage level in all duty periods will be lowered. On the other hand, a small k will shorten network lifetime and waste sensor energy. If duty periods are thought of as set covers, finding the maximal k such that full coverage in every set cover (duty period) is NP-complete [Cardei and Du 2005]. This is not a problem to [Ai et al. 2008] as that study did not demand a 100% coverage level in every duty period.

Our work differs from [Ai et al. 2008] in many ways. Coverage is a requirement in our work but a performance metric to maximize in [Ai et al. 2008]. On the other hand, network lifetime is assumed known in [Ai et al. 2008] but is a performance metric to maximize in this study. Finally, we consider non-uniform coverage requirement and sensor energy while the authors in [Ai et al. 2008] did not.

3. THE PROPOSED APPROACH

3.1. Problem Definition

We assume n sensors and m targets in a closed region. The ability of a sensor to cover some target is deterministic rather than probabilistic. This can be determined by, for example, applying the commonly-adopted notion of sensing range (all targets within the sensing range of some sensor are covered by that sensor.) We decompose the work schedule problem into independent coverage problems, each corresponding to the task of forming a set cover. We consider both disjoint and non-disjoint set covers.

We define *target coverage game* as follows. Let $P = (p_1, p_2, \ldots, p_n)$ denote the player set, which consists of all sensor nodes. For a tuple of all targets $T = (t_1, t_2, \ldots, t_m)$, let $Q = (q_1, q_2, \ldots, q_m)$ denote the coverage requirement of each target, where q_i is the number of sensors demanded by target t_i. A player's choice of being active or not is represented by 1 or 0, respectively, so each player p_i has a strategy set $S_i = \{0, 1\}$. A strategy profile is thus an n-tuple $C = (c_1, c_2, \ldots, c_n)$, where $c_i \in S_i$ represents p_i's choice. In this paper, strategy profile $(c_1, c_2, \ldots, c_n)$ is sometimes coded as a bit string $b_1 b_2 \cdots b_n$, where $b_i = c_i$, $1 \leq i \leq n$. For a specific p_i, we may express C as $C = (c_i, C_{-i})$, where $C_{-i} = (c_1, c_2, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n)$ denotes the set of all other player's choices other than p_i's. Function $u_i(C)$ gives p_i's utility (payoff) with respect to strategy profile C. We shall explore how to design $u_i(C)$ in the next subsection. The target coverage game $\Gamma = [P; \{S_i\}_{i=1}^{n}; \{u_i\}_{i=1}^{n}]$ can be formally defined by $\max_{c_i \in S_i} u_i(c_i, C_{-i})$ for all $i = 1, 2, \ldots, n$.

The game under consideration is a *noncooperative dynamic* game. In a noncooperative game, players do not cooperate with each other to seek the system's benefit. A game is dynamic if players take turns to make their decisions, knowing what

Table I. Partial list of notations

Notation	Meaning
Γ	The proposed target coverage game
n	Number of sensor nodes
m	Number of targets
P	The tuple of all sensor nodes; $P = (p_1, p_2, \ldots, p_n)$
T	The tuple of all targets; $T = (t_1, t_2, \ldots, t_m)$
Q	Coverage requirement; $Q(q_1, q_2, \ldots, q_m)$, where q_i is the number of sensors demanded by target t_i
T_i	The set of targets that p_i can cover
P_j	The set of sensors that can cover t_j
S_i	Strategy set of player p_i; $S_i = \{0, 1\}$ for all i
Y	Strategy space; $Y = S_1 \times S_2 \times \cdots \times S_n$
C	Strategy profile; $C = (c_1, c_2, \ldots, c_n) \in Y$
$u_i(C)$	p_i's utility with respect to strategy profile C

decisions have already been made. That definition corresponds to the *myopic* behavior that a sensor node will change its strategy (being active or not) whenever that change increases its utility. Formally, the *best response function* for sensor p_i is $r_i(C_{-i}) = \{c_i \in S_i | \forall c_i' \in S_i : u_i(c_i, C_{-i}) \geq u_i(c_i', C_{-i})\}$.

To compute u_i during the proposed game, p_i needs to know the present strategy profile (i.e., the current state of the game). If the game state is maintained by a particular entity or infrastructure, then all players can update and access the state information without interactions with each other. However, in a fully decentralized environment where the whole game state is collectively kept by all players in a distributed manner, players need to share their local game states (i.e., their own choices) with each other. Therefore, while the game is designed to be a noncooperative game, running the game may demand cooperative efforts from players. This type of cooperation causes no problem in a traditional sensor network since all sensors are controlled by one entity. In a participatory sensing framework where individual sensors may be controlled by different entities, sensors may have the incentive to cheat on or deny cooperation with their neighbors. Extra efforts are needed to deal with such misbehaviours. To simplify the algorithm design, this paper assumes a fully decentralized environment in a traditional sensor network. Implementation details about the cooperation under this environment are presented in Sec. 3.4.

We assume that the sensor density is sufficiently high such that Q-coverage can be guaranteed initially if all sensor nodes are active. The goal of the proposed target coverage game is to maximize the network lifetime while respecting the Q-coverage requirement by powering off redundant sensors. We shall first present a game design that respects the Q-coverage requirement while disregarding the sensor energy. We then show how to convert the game design into a practical distributed algorithm (protocol) that takes into account of sensor energy and game convergence time. For the ensuing discussions, most of the symbols used are summarized in Table I.

3.2. Game Design

This subsection presents our game-theoretic approach to the target coverage problem. Energy issue and other practical concerns will be addressed in Sec. 3.4. A critical mission of our work is to design a utility function for every player. This task centers on how much profit a sensor should gain when it covers some target. That profit should be high enough to motivate sensors to contribute their coverage so as to meet the coverage requirement. On the other hand, the profit should be sufficiently low to avoid possible activation of redundant sensors. How to distribute the profit among all contributors is also an issue. If all the credit goes to the last one that fills the gap between requested coverage and offered coverage, then no sensor would have the incentive to be the first

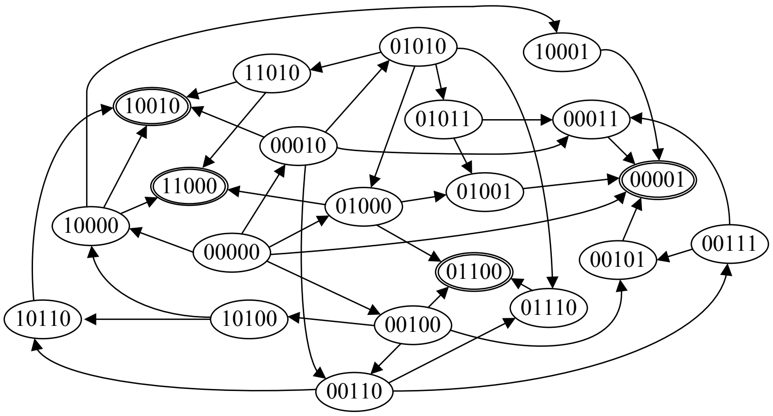

Fig. 3. Possible transitions of strategy profiles for the scenario shown in Fig. 2

few contributors. An even or uneven distribution of profit among all contributors may bring in some desired effects such as motivating the first few contributors, but it may also cause some other unexpected side effects. Generally speaking, the design should meet the following requirements:

Stability. The game should eventually enter a state where every player is satisfied with her/his payoff. Without this requirement, the game may not end up with a deterministic result.

Correctness. All possible final states of the game should meet the coverage requirement.

Efficiency. There should not be any active yet redundant sensors in any final state of the game in order to avoid unnecessary energy waste.

Feasibility. Any player's payoff should be defined as a function that only depends on information accessible to the player. Existence of any parameter in the utility function that involves global knowledge causes difficulties when turning the game into a feasible solution. The computation of the utility function should be as simple as possible so as to minimize the demand of computing resource on sensor devices.

We present first our design for the utility function and then the rationale behind the design. Let $g_j(C)$ be the profit that a player can gain with respect to a strategy profile C if it covers target t_j:

$$g_j(C) = \begin{cases} \alpha & \text{if } 0 < (\sum_{p_i \in P_j} c_i) \le q_j \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $\alpha > 0$ is a constant. The utility function of p_i is defined as

$$u_i(C) = \begin{cases} \left(\sum_{t_j \in T_i} g_j(C) \right) - \beta & \text{if } c_i = 1 \\ 0 & \text{if } c_i = 0, \end{cases} \tag{2}$$

where β is another constant such that $0 < \beta < \alpha$. Clearly, the payoff of a sensor's choice depends not only on the set of targets it covers, but also the choices of other sensors that have overlapping coverages with it.

In (1), the profit of any active sensor that covers t_j becomes zero when more sensors than needed are active. This is to prevent redundant sensors from being active. The condition $\beta < \alpha$ ensures that a sensor enters sleep mode only if it does not contribute any coverage.

For the scenario shown in Fig. 2, Fig. 3 shows all possible transitions of strategy profiles starting from 00000 with the defined utility function[1]. It is not difficult to verify that sensors choose to enter sleep mode if and only if they do not contribute any coverage. For example, strategy profile 01011 may change to 01001 or 00011 because both p_2 and p_4 contribute no coverage. Player p_5, on the other hand, will not revise its decision here because covering t_1 gives it a positive payoff $\alpha - \beta$. It is also not hard to see that all transitions converge to four possible final results: 10010, 11000, 01100, and 00001. The coverage requirement is satisfied in all these results.

3.3. Properties of the Game

We have already verified that the stability, the correctness, and the efficiency requirements are met in the particular scenario shown in Fig. 2. This subsection shall prove that these requirements are universally met with the designed utility function.

The stability requirement corresponds to Nash equilibria in our game. A Nash equilibrium is a strategy profile where no player can further increase its own utility by unilaterally changing its choice.

Definition 3.1 (*Nash equilibrium*). Given a target coverage game $\Gamma = [P; \{S_i\}_{i=1}^{n}; \{u_i\}_{i=1}^{n}]$, a strategy profile $C^* = (c_1^*, c_2^*, \ldots, c_n^*)$ is a Nash equilibrium if $\forall i \in \{1, 2, \ldots, n\} : \forall c_i \in S_i :: u_i(c_i^*, C_{-i}^*) \geq u_i(c_i, C_{-i}^*)$.

Recall that in our model, a player can change its choice if that change increases its payoff. The change may trigger another player's change and so on. If a Nash equilibrium does not exist in this game, activities of changing choices will persist and the game cannot enter a stable state.

To prove Nash equilibrium of the proposed game, we need to introduce some particular types of games. In a *potential game* [Monderer and Shapley 1996], we can find a potential function whose value increases whenever a player increases her payoff by changing strategy.

Definition 3.2 (*Potential game*). $\Gamma = [P; \{S_i\}_{i=1}^{n}; \{u_i\}_{i=1}^{n}]$ is a potential game if there exists a potential function $\pi(c_i, C_{-i})$ such that $\forall p_i \in P : \forall c_i, c_i^* \in S_i, c_i \neq c_i^* ::$ $\text{sgn}(u_i(c_i^*, C_{-i}) - u_i(c_i, C_{-i})) = \text{sgn}(\pi(c_i^*, C_{-i}) - \pi(c_i, C_{-i}))$, where

$$\text{sgn}(\rho) = \begin{cases} 1 & \text{if } \rho > 0 \\ 0 & \text{if } \rho = 0 \\ -1 & \text{if } \rho < 0. \end{cases}$$

Exact potential game [Monderer and Shapley 1996] is a particular type of potential game as defined below.

Definition 3.3 (*Exact potential game*). $\Gamma = [P; \{S_i\}_{i=1}^{n}; \{u_i\}_{i=1}^{n}]$ is an exact potential game if it admits a potential function $\pi(c_i, C_{-i})$ such that

$$\forall p_i \in P : \forall c_i, c_i^* \in S_i, c_i \neq c_i^* :: u_i(c_i^*, C_{-i}) - u_i(c_i, C_{-i}) = \pi(c_i^*, C_{-i}) - \pi(c_i, C_{-i}). \quad (3)$$

Potential functions that satisfy (3) are *exact potential functions*.

These definitions indicate that we need to find a potential function or an exact potential function to show some game being a potential or an exact potential game. This is the most difficult part of our analytic work. Note that a summation of all the player's

[1] The possible transitions of strategy profiles are sensitive to the starting strategy profile. For example, transitions of strategy profiles starting from 11111 (not shown here) are different from those shown in Fig. 3.

payoffs $f(C) = \sum_{i=1..n} u_i(C)$ in the proposed game is not a potential function: a transition from strategy profiles 01010 to 11010 results in a change of the payoff summation from $\alpha - 2\beta$ to $\alpha - 3\beta$, a negative increase.

THEOREM 3.4. *The proposed target coverage game $\Gamma = [P; \{S_i\}_{i=1}^n; \{u_i\}_{i=1}^n]$ is an exact potential game.*

PROOF. Consider the following function:

$$\pi(C) = \left(\sum_{j=1}^{m} \sum_{k=0}^{v(j)} \delta(j,k) \right) - \beta \sum_{i=1}^{n} c_i, \tag{4}$$

where

$$v(j) = \sum_{l \in P_j} c_l$$

and

$$\delta(j,k) = \begin{cases} \alpha & \text{if } 0 < k \le q_j \\ 0 & \text{otherwise.} \end{cases}$$

We shall prove that Γ is an exact potential game by showing that $\pi(C)$ is an exact potential function. Let $C = (c_i, C_{-i})$ and $\bar{C} = (\bar{c}_i, C_{-i})$ be two strategy profiles before and after some p_i changes its strategy from c_i to $\bar{c}_i$, respectively. The transition from C to $\bar{C}$ is possible only if $u_i(c_i, C_{-i}) < u_i(\bar{c}_i, C_{-i})$. There are two possibilities in such a transition.

Case 1: $c_i = 1$ and $\bar{c}_i = 0$. For all $t_j \in T$, let υ_j and $\bar{\upsilon}_j$ be the values of $v(j)$ in C and $\bar{C}$, respectively. Clearly,

$$\bar{\upsilon}_j = \begin{cases} \upsilon_j - 1 & \text{if } t_j \in T_i \\ \upsilon_j & \text{if } t_j \notin T_i. \end{cases} \tag{5}$$

Let $\sigma = \sum_{i=1}^n c_i$ in C. (4) can be rephrased as

$$\pi(c_i, C_{-i}) = \sum_{t_j \in T_i} \sum_{k=0}^{\upsilon_j} \delta(j,k) + \sum_{t_j \notin T_i} \sum_{k=0}^{\upsilon_j} \delta(j,k) - \beta\sigma, \tag{6}$$

where

$$\sum_{t_j \in T_i} \sum_{k=0}^{\upsilon_j} \delta(j,k) = \sum_{t_j \in T_i} \sum_{k=0}^{\upsilon_j - 1} \delta(j,k) + \sum_{t_j \in T_i} \delta(j, \upsilon_j). \tag{7}$$

If $\exists t_j \in T_i$ such that $g_j(C) \neq 0$, then $u_i(C) \geq \alpha - \beta > 0 = u_i(\bar{C})$ and the transition from C to $\bar{C}$ is not possible. Therefore, $\sum_{t_j \in T_i} g_i(C) = 0$. It follows that $u_i(\bar{c}_i, C_{-i}) - u_i(c_i, C_{-i}) = 0 - (0 - \beta) = \beta$. Another implication is that $\delta(j, \upsilon_j) = 0$ for all $t_j \in T_i$, and (7) reduces to

$$\sum_{t_j \in T_i} \sum_{k=0}^{\upsilon_j} \delta(j,k) = \sum_{t_j \in T_i} \sum_{k=0}^{\upsilon_j - 1} \delta(j,k). \tag{8}$$

By (8), (6) becomes

$$\pi(c_i, C_{-i}) = \sum_{t_j \in T_i} \sum_{k=0}^{\upsilon_j - 1} \delta(j,k) + \sum_{t_j \notin T_i} \sum_{k=0}^{\upsilon_j} \delta(j,k) - \beta\sigma. \tag{9}$$

On the other hand,

$$\begin{aligned}\pi(\bar{c}_i, C_{-i}) &= \sum_{t_j \in T_i} \sum_{k=0}^{\bar{\upsilon}_j} \delta(j,k) + \sum_{t_j \notin T_i} \sum_{k=0}^{\bar{\upsilon}_j} \delta(j,k) - \beta(\sigma - 1) \\ &= \sum_{t_j \in T_i} \sum_{k=0}^{\upsilon_j - 1} \delta(j,k) + \sum_{t_j \notin T_i} \sum_{k=0}^{\upsilon_j} \delta(j,k) - \beta(\sigma - 1) \qquad (10)\end{aligned}$$

by (5). Therefore,

$$\pi(\bar{c}_i, C_{-i}) - \pi(c_i, C_{-i}) = \beta, \qquad (11)$$

which is equal to the value of $u_i(\bar{c}_i, C_{-i}) - u_i(c_i, C_{-i})$.

Case 2: $c_i = 0$ and $\bar{c}_i = 1$. Since $u_i(c_i, C_{-i}) = 0$,

$$u_i(\bar{c}_i, C_{-i}) - u_i(c_i, C_{-i}) = \sum_{t_j \in T_j} g_j(\bar{C}) - \beta. \qquad (12)$$

Let υ_j, $\bar{\upsilon}_j$, σ be defined as in Case 1. We have

$$\bar{\upsilon}_j = \begin{cases} \upsilon_j + 1 & \text{if } t_j \in T_i \\ \upsilon_j & \text{if } t_j \notin T_i. \end{cases} \qquad (13)$$

By (13),

$$\begin{aligned}\pi(\bar{c}_i, C_{-i}) &= \sum_{j=1}^{m} \sum_{k=0}^{\bar{\upsilon}_j} \delta(j,k) - \beta(\sigma + 1) \\ &= \sum_{t_j \in T_i} \sum_{k=0}^{\upsilon_j + 1} \delta(j,k) + \sum_{t_j \notin T_i} \sum_{k=0}^{\upsilon_j} \delta(j,k) - \beta(\sigma + 1). \qquad (14)\end{aligned}$$

Since

$$\pi(c_i, C_{-i}) = \sum_{t_j \in T_i} \sum_{k=0}^{\upsilon_j} \delta(j,k) + \sum_{t_j \notin T_i} \sum_{k=0}^{\upsilon_j} \delta(j,k) - \beta\sigma, \qquad (15)$$

we have

$$\begin{aligned}\pi(\bar{c}_i, C_{-i}) - \pi(c_i, C_{-i}) &= \sum_{t_j \in T_i} \delta(j, \upsilon_j + 1) - \beta \\ &= \sum_{t_j \in T_i} \delta(j, \bar{\upsilon}_j) - \beta. \qquad (16)\end{aligned}$$

By (12) and (16), we have $u_i(\bar{c}_i, C_{-i}) - u_i(c_i, C_{-i}) = \pi(\bar{c}_i, C_{-i}) - \pi(c_i, C_{-i})$. □

Let $Y = S_1 \times S_2 \times \cdots \times S_n$ be the strategy space of Γ. Since Y is finite, Γ is a finite (exact) potential game. It has been proven [Monderer and Shapley 1996] that every finite potential game possesses a Nash equilibrium. In fact, it is also not difficult to see that *every* strategy profile in a finite exact potential game eventually leads to a Nash equilibrium through a series of *improvements*, where an improvement refers to a positive increase in the value of the exact potential function due to a transition of strategy profile caused by the myopic behavior of a single player.

COROLLARY 3.5. *Starting from any strategy profile, Γ eventually ends up with a Nash equilibrium.*

Corollary 3.5 confirms the stability of the proposed game. The following theorem proves the correctness of the proposed game.

THEOREM 3.6. *Q-coverage is ensured at every Nash equilibrium in Γ.*

PROOF. Since Q-coverage is guaranteed by turning on all sensors, we have $|P_j| \geq q_j$ for all $t_j \in T$. We shall prove the theorem by way of contradiction. Suppose that there exists some Nash equilibrium $C = (c_1, c_2, \ldots, c_n)$ in Γ where Q-coverage is not met. It follows that there must be some target $t_j \in T$ in C for which $\sum_{p_i \in P_j} c_i < q_j$ and $\exists p_i \in P_j : c_i = 0$. For any such p_i, changing c_i from 0 to 1 can cause a payoff change from 0 to at least $\alpha - \beta > 0$ (which comes from the new coverage on t_j). Therefore, C cannot be a Nash equilibrium, which contradicts with the assumption. □

It is well known that Nash equilibria are not necessarily desired results. In fact, global optima in games may not even exist. Nevertheless, we can seek *Pareto optimal* results. These results are desired when utilities defined in a game have direct physical significance such as payments or credits that someone is willing to give to individual players.

Definition 3.7 (*Pareto optimal*). A strategy profile $C = (c_1, c_2, \ldots, c_n)$ is Pareto optimal if and only if there exists no other strategy profile $C' = (c'_1, c'_2, \ldots, c'_n)$ such that $\forall i \in \{1, 2, \ldots, n\} : u_i(C') \geq u_i(C)$ and $\exists j \in \{1, 2, \ldots, n\} : u_j(C') > u_j(C)$.

THEOREM 3.8. *Every Nash equilibrium in Γ is also Pareto optimal.*

PROOF. First note that for all player p_i, its payoff is either 0 (iff $c_i = 0$) or $k\alpha - \beta > 0$ (iff $c_i = 1$), where $k = 1, 2, \ldots, |T_i|$. Suppose, by way of contradiction, that $C = (c_1, c_2, \ldots, c_n)$ is a Nash equilibrium but not Pareto optimal. This implies that there exists some strategy profile $C' = (c'_1, c'_2, \ldots, c'_n)$ such that $\forall i \in \{1, 2, \ldots, n\}$: $u_i(C') \geq u_i(C)$ and $\exists i \in \{1, 2, \ldots, n\} : u_i(C') > u_i(C)$. Consider every player p_i such that $u_i(C') > u_i(C)$. Since $u_i(C) \geq 0$, $u_i(C')$ must be greater than 0 and hence c'_i must be 1. The condition $u_i(C') > u_i(C)$ and $c'_i = 1$ implies that there must exist some target $t_j \in T_i$ for which $g_j(C) = 0$ and $g_j(C') > 0$. The condition $g_j(C) = 0$ implies either $\sum_{p_i \in P_j} c_i = 0$ or $\sum_{p_i \in P_j} c_i > q_j$. However, it is impossible that t_j is not covered at all in C (*i.e.*, $\sum_{p_i \in P_j} c_i = 0$) because C is a Nash equilibrium where Q-coverage is assured by Theorem 3.6. If t_j is over-covered in C (*i.e.*, $\sum_{p_i \in P_j} c_i > q_j$) but not in C' (because $g_j(C') > 0$), then there must exist some player $p_k \in P_j$ such that $c_k = 1$ and $c'_k = 0$. By definition, $u_k(C) > u_k(C')$, which contradicts with our assumption that $\forall i \in \{1, 2, \ldots, n\} : u_i(C') \geq u_i(C)$. □

Pareto optima are good and desired results in games, but they do not necessarily correspond to the best solution to our problem. For example, all the four Nash equilibria shown in Fig. 3 are Pareto optimal, but one of the equilibria, 11000, seems worse than the others because there is one target (t_2) that is over-covered in this equilibrium. Whether our game will end up with such a result is a probability problem. If we view each strategy profile as a state, transitions of strategy profiles can be modeled as a discrete-time Markov chain. In this way, Nash equilibria in our game correspond to absorbing states in the chain. Fig. 4 shows the state diagram of the Markov chain corresponding to Fig. 3 under the assumption of equal transition probability among all possible next states of any state. Observe that the chain is an absorbing Markov chain, which is already implied by Corollary 3.5. Starting from state 00000, the probability that the chain will enter absorbing states 00001, 10010, 01100, and 11000 are 0.5, 0.175, 0.1729, and 0.1521, respectively. We can see that the probability of reaching the equilibrium in question is only 0.1512.

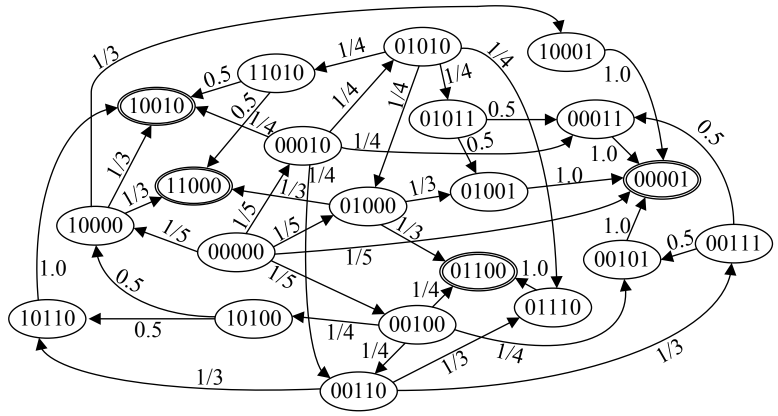

Fig. 4. Markov diagram when modeling the transitions in Fig. 3 as a Markov chain

The possibility of over-covered targets renders our method not optimal. Nevertheless, concerning the efficiency of the proposed game, we can at least ensure that there are no active redundant sensors in Nash equilibria. The reason is that if there is any such sensor, its payoff will be $-\beta$ as it is redundant. Changing its choice from 1 to 0 will give it a payoff of 0, a positive gain which contradicts with the definition that any player cannot further improve its payoff in a Nash equilibrium.

We summarize our theoretic work by the following corollary.

COROLLARY 3.9. *Starting from any strategy profile, the proposed target coverage game eventually ends up with a Nash equilibrium which guarantees Q-coverage and is Pareto optimal. Also no redundant sensors are active in the Nash equilibrium.*

3.4. Implementation Details

Several issues remain to be solved when applying the game design to WSNs. The utility function in the proposed game does not take into account of sensor energy. This may not be optimal when sensors have different amounts of energy. Although the game always ends up in a Nash equilibrium, the convergence time may be too long. The correctness of the proposed game model relies on the assumption that no two or more players make or change their decisions simultaneously, which may not be guaranteed when we turn the game design into a protocol running in the real world. All these issues will be addressed in this subsection.

To implement the utility function locally in every sensor node, every sensor p_i should know the set of targets it can cover (T_i) and the choices taken by all other sensors that have overlapping coverage with it ($\{c_j | p_j \in P, T_j \cap T_i \neq \phi\}$). T_i is essentially the set of targets that p_i is able to monitor or detect, so the knowledge of T_i is intrinsic to p_i. To obtain the latter information, p_i should be able to communicate with all p_j for which $T_i \cap T_j \neq \phi$. With the assumption of uniform sensing range, this requirement can be met by having a transmission range that is twice of the sensing range. More specifically, whenever p_i makes a new decision c_i, p_i broadcasts c_i together with T_i to all its neighboring nodes. The setting of transmission range ensures that all sensors p_j for which $T_i \cap T_j \neq \phi$ will receive T_i and update c_i. With that information, any p_j is able to react accordingly. The same setting has been used to ensure network connectivity while maintaining full area coverage [Xing et al. 2005; Zhang and Hou 2005].

The above-mentioned protocol can be refined in several ways. First, since T_i for all i is static, it needs to be broadcast only once in the very beginning. Second, since c_i

has only two possible values, p_i can transmit a special signal that uniquely identifies p_i itself in two different frequency channels, one for each value of c_i. Alternatively, p_i can use two orthogonal codes in a code-division multiple access system to represent the binary value of c_i.

The theoretic results presented in the previous subsection implicitly assume that no two or more sensors with overlapping coverage make or change their decisions simultaneously. This may not be the case if simultaneous decision makings are not prohibited by the protocol. Fortunately, the correctness of the proposed game model does not rely on any particular decision sequence made by players. This means that we can *arbitrarily serialize* concurrent decisions without breaking any desired properties, which can be achieved by a backoff scheme.

The backoff scheme operates as follows. Every player has a backoff timer. After making or updating a decision, a player p_i sets up her backoff timer with a randomly determined value. The decision will be announced by broadcasting after the backoff timer counts down to zero. If, before the backoff time expires, p_i receives a decision announcement from another player p_j for which $T_i \cap T_j \neq \phi$, p_i aborts the scheduled announcement and re-evaluates its payoff. If the re-evaluation still supports the to-be-announced decision, p_i reloads its backoff timer with a new randomly determined value, and restarts another backoff countdown. Note that all other players p_j for which $T_i \cap T_j = \phi$ are not affected by p_i's announcement. This means that the backoff-based serialization is local. Decision parallelism among sensors that do not have overlapping coverage is still allowed by the proposed scheme.

The introduction of backoff timers can also make our game design power-aware. Instead of setting up backoff timers with randomly determined values, we can give a shorter backoff time to sensors with more energy, which effectively gives priority to these sensors in decision makings. Such a design can hopefully prolong network lifetime as it preserves the largest possible number of living sensors. The same technique can be used to speedup the convergence time to Nash equilibrium. Here the idea is to give a shorter backoff time to players with higher utility gains. It works as higher utility gains lead to greater improvements on the value of the exact potential function and thus shorter convergence time. Consequently, several implementation variants of our game can be derived, which only differ in the setting of the backoff timer. Let e_i denote the power level of sensor p_i, g_i denote the utility gain due to p_i's current decision change, and $T_b(i)$ be the value of p_i's backoff timer for announcing p_i's decision. These variants include the following.

— Random, where $T_b(i)$ is randomly chosen from a fixed range,
— Utility-Based (UB), where $T_b(i) \propto 1/g_i$,
— Utility-Based Random (UBR), which is a hybrid of Random and UB in the sense that $T_b(i)$ is randomly determined but the range of the random value is proportional to $1/g_i$,
— Power-Based (PB), where $T_b(i) \propto 1/e_i$,
— Power-Based Random (PBR), where $T_b(i)$ is a random number with range proportional to $1/e_i$,
— Joint-Utility-Power (JUP), which is a hybrid of UB and PB as $T_b(i)$ is proportional to a weighted sum of g_i and $1/e_i$.

Since all these variants only alter the decision sequences, all the desired properties proven in the previous subsection are still preserved. We shall study the performance of these variants through simulations in the next section.

Finally, we comment on the computation resource requirement of the proposed game. Evaluation of the utility function defined in (1) and (2) involves only additions and com-

Table II. Parameter setting

Parameter	Range	Default
n: Number of sensors	$[150, 500]$	300
m: Number of targets	$[10, 60]$	25
q_i: Coverage requirement of target t_i	$[2, 6]$	3
r_s: Sensing range (m)	$[100, 300]$	150
r_c: Transmission range (m)		$2r_s$
e_i: Residual energy of sensor p_i (units)	$[10, 70]$	20
t: Length of duty period	$[4, 36]$	20
α in (1)	$[2, 18]$	10
β in (2)		1
ω used by JUP	[0,1]	0.2

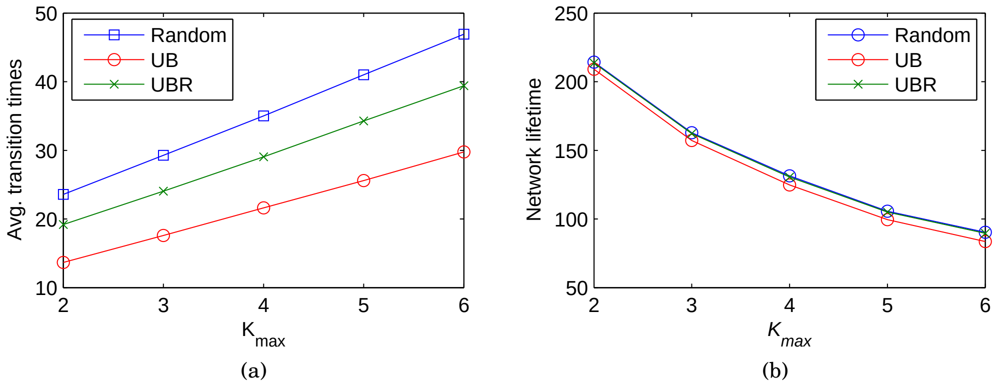

Fig. 5. Comparisons among Random, UB, and UBR in (a) average convergence time and (b) network lifetime with q_i randomly chosen from $[2, K_{\max}]$

parisons. The introduction of a backoff scheme demands additional one or two multiplication and/or division operations. Overall computation resource demand is trivial.

4. SIMULATION RESULTS

We studied the performance of the proposed game through simulations. We were interested in the performance gains realized by different settings of the backoff time in our game implementations and how well the proposed game-theoretic approach performs compared with prior work.

We assumed a 600×600 m^2 area, within which n sensor nodes and m target objects were randomly placed by a uniform distribution. Table II lists the parameter setting for the simulations. Unless otherwise specified, all parameters were set to their default values. Each result is an average over 500 runs. For the proposed game-theoretic approach, the initial choice of each player is set to sleep mode.

Two metrics were of interest in the experiments: network lifetime and game convergence time. Network lifetime measures the length of the time from the very beginning to the first time the coverage requirement is not met. To simplify the representation, we assumed that power dissipation in sensors is linear and running out one unit of power takes one unit of time. Game convergence time counts the number of strategy profile transitions in a game. It stands for the cost of the proposed approach. Set covers yielded are disjoint by default.

4.1. Comparisons Among Implementation Variants

The first experiment was to study whether utility-aware implementations (UB and UBR) improve convergence time of the game and whether such improvement, if any,

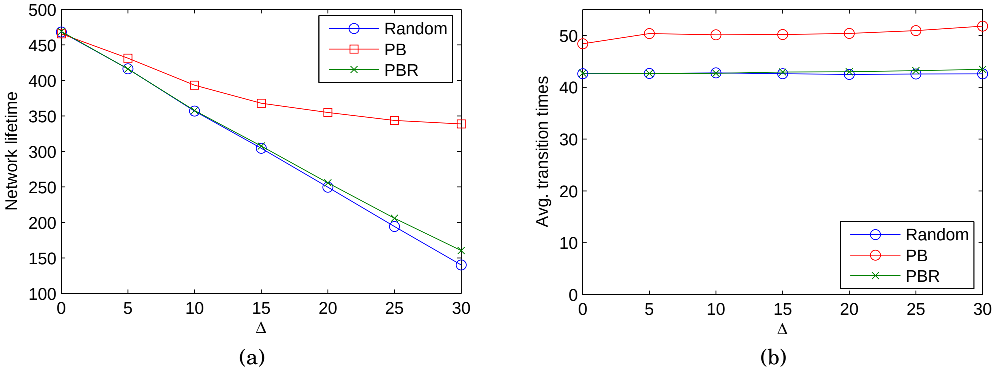

Fig. 6. Comparisons among Random, PB, and PBR in (a) network lifetime and (b) average convergence time with e_i randomly chosen from $[40 - \Delta, 40 + \Delta]$

comes at the cost of degraded network lifetime. The coverage requirement of each target t_i was varied by setting q_i to be a random integer within the range $[2, K_{\max}]$. Fig. 5 shows the experimental results with $n = 150$. We can see that UB significantly reduces the convergence time when compared with Random, which indicates that giving decision priority to players with high utility gains effectively shortens game convergence time. This also explains the intermediate performance of UBR. On the other hand, Random and UBR are hardly distinguishable concerning network lifetime, while UB is slightly worse than both. This result suggests that the order of decision announcements does not significantly alter network lifetime (when sensors have uniform power) and utility-aware designs are beneficial if game convergence time is of concern.

The next experiment investigated whether network lifetime benefits by power-aware designs (i.e., PB and PBR) when sensors have non-uniform energy levels. The energy level of each sensor p_i is varied by setting e_i to be a random integer within the range $[40 - \Delta, 40 + \Delta]$. The results shown in Fig. 6(a) confirm the superiority of PB over Random in terms of network lifetime when sensors have non-uniform energy levels. However, PB also incurs a longer convergence time than Random (Fig. 6(b)), where the gap seems independent of the difference of energy. This is reasonable as the utility function does not take energy level into consideration. This suggests that PB is preferable when network lifetime is of concern while sensors have different amounts of energy.

As the utility-aware designs reduce convergence time while the power-aware designs prolong network lifetime, we studied the joint effects of utility- and power-aware backoff time by testing JUP. Here $T_b(i)$ for each sensor p_i is set to proportional to $\omega \times 1/g_i + (1 - \omega) \times 1/e_i$, where $0 \leq \omega \leq 1$ is a weighting factor. Fig. 7 shows the results with respect to various settings of ω under the most different setting of coverage requirements and energy levels. Observe that an increased ω leads to a lower network lifetime but also a shorter convergence time. This is expected as a large ω gives priority to UB which benefits convergence time. On the other hand, a small ω gives priority to PB and thus results in a higher network lifetime. In the following experiments, JUP with $\omega = 0.2$ is used as our approach.

Parameter α in (1) was set to 10 in our experiments. We were also interested in whether this setting has a significant effect on our performance metrics. Fig. 8 shows the results of our investigations. We can see that when $\Delta = 0$, the setting of α seems irrelevant to the performance metrics. When $\Delta = 30$, network lifetime slightly in-

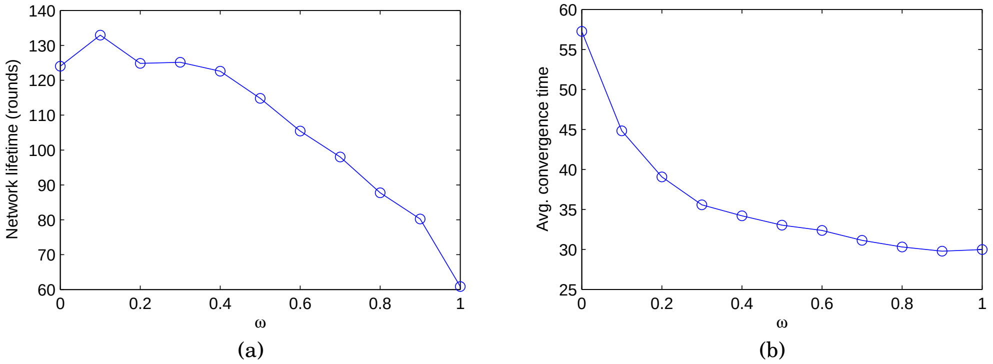

Fig. 7. (a) Network lifetime and (b) game convergence time in JUP with respect to the value of ω ($K_{\max} = 6$ and $\Delta = 30$)

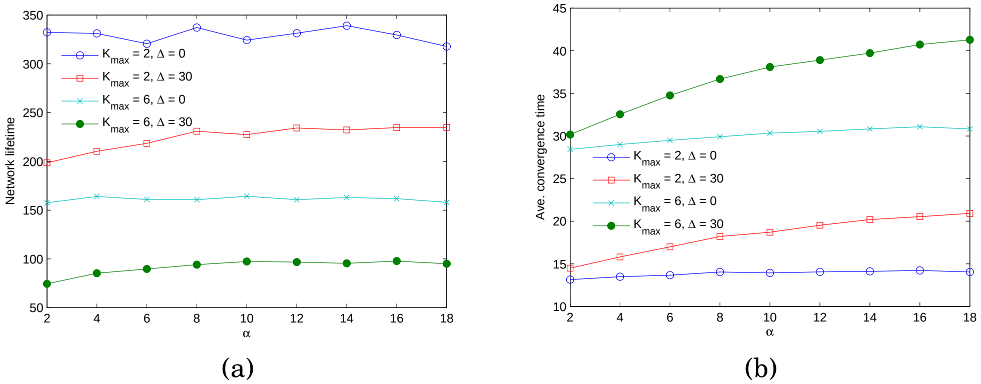

Fig. 8. (a) Network lifetime and (b) game convergence time in JUP with respect to the value of α.

creases and game convergence time significantly increases as α increases. In general, the setting of $\alpha = 10$ is not biased.

4.2. Comparisons With Centralized Heuristics: Using Disjoint Set Covers

We compared the proposed game-theoretic approach with three commonly-adopted greedy heuristics that all work in a centralized manner. These heuristics all construct a set cover by incrementally adding sensor nodes into the set cover:

- Maximal Coverage First (MCF), where sensors that cover the most targets are added into the set cover with priority.
- Most Power First (MPF), where sensors with the most power (energy) are included into the set cover with priority. This policy was adopted in [Chaudhary and Pujari 2009].
- Most Critical Sensor First (MCSF), where sensors that have the least overlapping coverage with other sensors, *i.e.*, those in $\arg\min_i\{\sum_{j\neq i} O_i(j)\}$, where

$$O_i(j) = \begin{cases} 0 & \text{if } T_i \cap T_j = \phi \\ 1 & \text{otherwise,} \end{cases}$$

  are added into the set cover with priority. A similar principle was adopted for area coverage in [Li and Gao 2008].

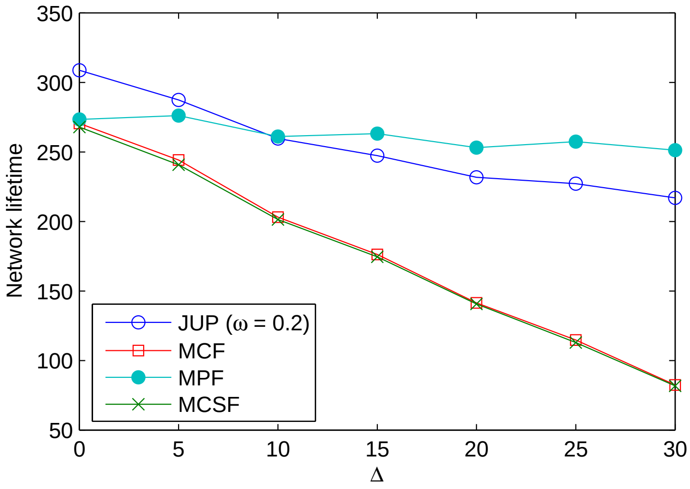

Fig. 9. Network lifetime with e_i randomly chosen from $[40 - \Delta, 40 + \Delta]$

When there were multiple sensors with the same priority, an arbitrary one was chosen. An additional *pruning* process was applied to all constructed set covers, where every sensor node in the set cover was rechecked to remove any redundant sensors from the final set cover [Chaudhary and Pujari 2009; Li and Gao 2008].

This set of experiments considered only disjoint set covers, where a set cover operates for the longest possible time (i.e., until the first sensor in the set cover exhausts its energy) before the next set cover takes over. No sensors in the current set cover can be included in any subsequent set cover even if these sensors still have energy when the current set cover retires.

Figure 9 shows how variances of energy level in sensor nodes affect network lifetime in all approaches under discussion. We can see that network lifetime generally decreases as the variance of energy level increases. MPF performs better than the other two heuristics when sensors do not have uniform energy and the performance gap increases as the variance of energy level increases, thanks to its power-aware design. JUP with $\omega = 0.2$ outperforms both MCF and MCSF. However, its superiority over MPF disappears when $\Delta > 10$.

When sensors have identical energy level but targets have different coverage requirements, all the counterparts perform the same (Fig. 10). This is justifiable since none of these methods considers non-uniform coverage requirements in set cover formations. Network lifetime generally decreases as K_{max} increases. This is because the mean of the requested coverage level, which is $(2 + K_{\text{max}})/2$, is proportional to K_{max}. Nevertheless, JUP with $\omega = 0.2$ outperforms its counterparts in all settings.

We also investigated the combined effects of non-uniform energy levels (with $\Delta = 30$) and non-uniform coverage requirements (with $K_{\text{max}} = 6$). Fig. 11 shows the resulting network lifetime versus the number of sensors. It is reasonable and desirable that adding more sensors in general leads to an increased network lifetime. Here MPF exhibits its superiority over the others, which is due to its power-aware design which works well when sensors have different energy levels. We argue that the setting of disjoint set covers also contributes to the advantage of MPF in lifetime. We shall explore more on this in the next subsection.

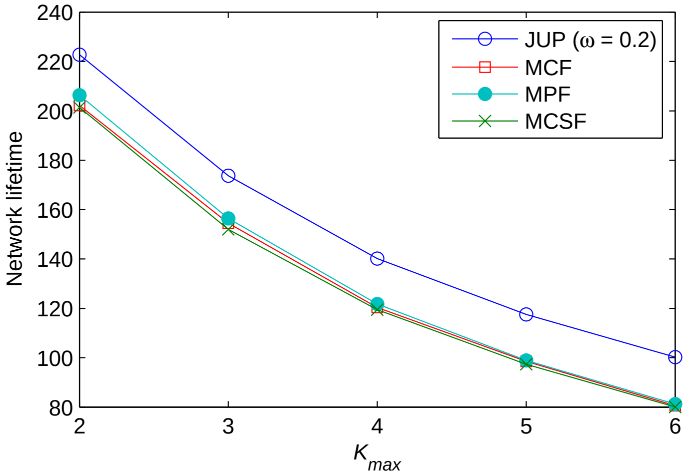

Fig. 10. Network lifetime with q_i being randomly chosen from $[2, K_{\max}]$

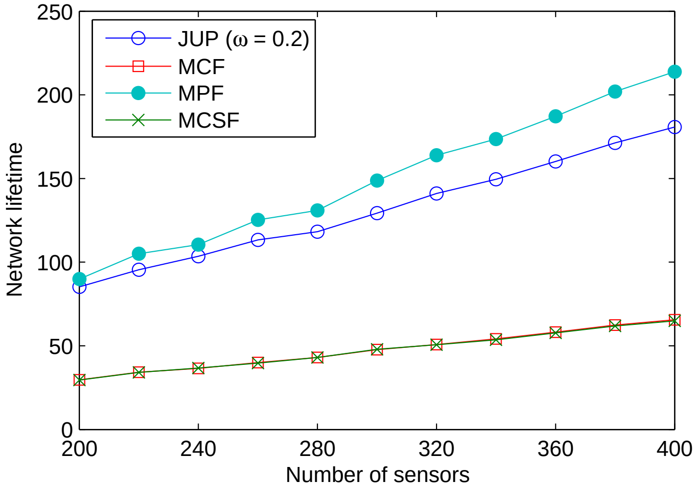

Fig. 11. Network lifetime with $\Delta = 30$ and $K_{\max} = 6$

4.3. Comparisons With Centralized Heuristics: Using Non-Disjoint Set Covers

When set covers can be non-disjoint, the results are expected to be different from those of disjoint set covers. To test non-disjoint set covers, we let a set cover operate for at most a fixed length of duty period t. After that time (or right before any sensor in the current set cover exhausts its energy, whichever is earlier), the set cover is off duty, and another set cover is formed to take over. Any sensors that have residual energy are eligible to be included in the subsequent set cover.

It was unknown whether the setting of t affects the resulting network lifetime. Therefore, we conducted an experiment to study the relationship between network lifetime and the length of duty periods for non-disjoint set covers. This experiment took the most extreme settings in energy level and coverage requirement ($\Delta = 30$ and

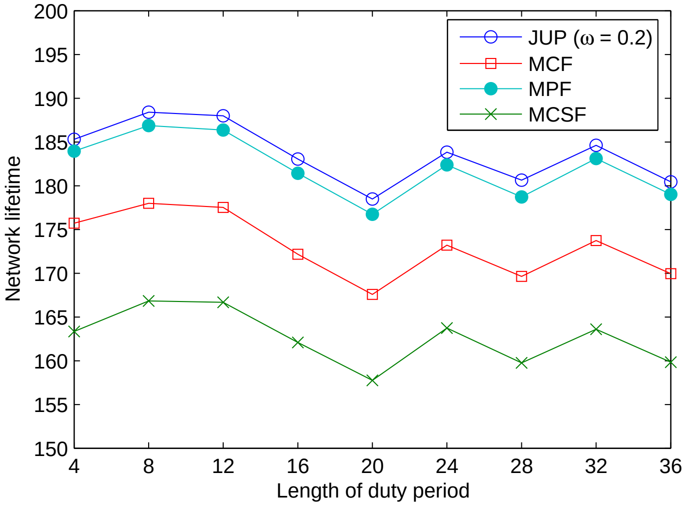

Fig. 12. Network lifetime versus the length of duty periods for non-disjoint set covers ($\Delta = 30$ and $K_{\max} = 6$)

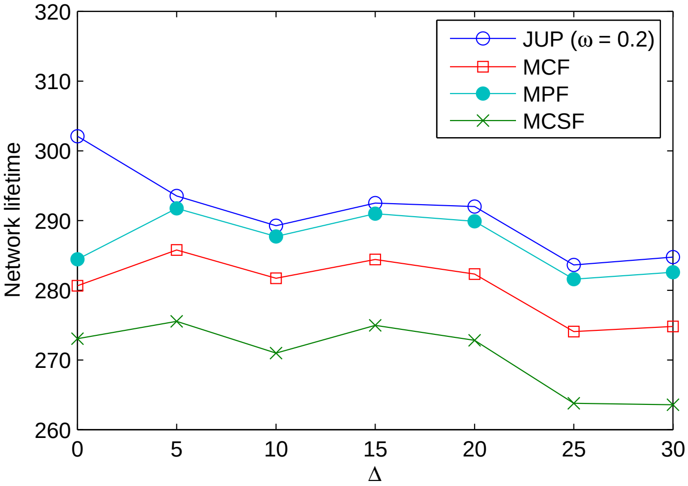

Fig. 13. Network lifetime for non-disjoint set covers with e_i being randomly chosen from $[40 - \Delta, 40 + \Delta]$

$K_{\max} = 6$). The results, as shown in Fig. 12, provide no evidence that network lifetime and the length of duty periods are closely related. We therefore took a default setting of 20-unit duty period and repeated all the experiments that had been done for disjoint set covers.

Figure 13 shows how network lifetime changes with the variance of sensor energy levels. Although the result still exhibits a decrease of network lifetime with Δ, the decreasing rate is much lower than the result of disjoint set cover (Fig. 9). The advantage of MPF over JUP in the case of disjoint set covers disappears in the case of non-disjoint set covers. In fact, the performance gap between the best (JUP) and the worst (MCSF) methods is shrunk in the case of non-disjoint set covers. The results indicate that for-

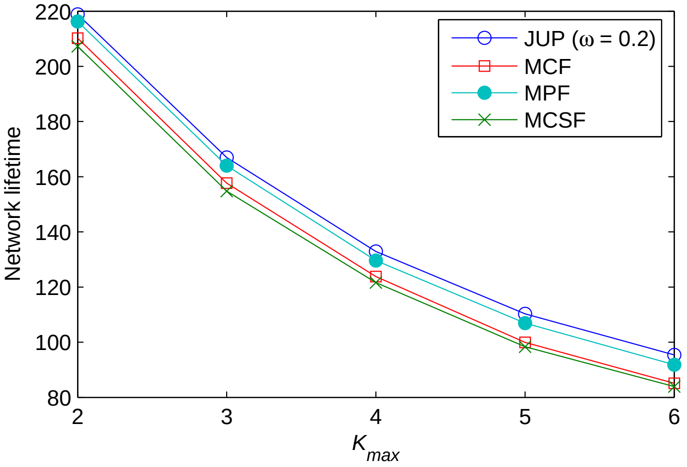

Fig. 14. Network lifetime for non-disjoint set covers with q_i being randomly chosen from $[2, K_{\max}]$

mation rules for non-disjoint set covers are less important to network lifetime than they are for disjoint set covers.

To study the relationship between network lifetime and the variance of coverage requirements, we fixed sensor's energy level to 20 units and varied $K_{\max}$. If the length of duty period had been uniformly set to 20 units, the result would have been identical to that shown in Fig. 10 because no sensor of a set cover can have residual energy when the set cover is off duty. Therefore, we let $t = 4$ and the result is shown in Fig. 14. Here the trend of decreasing network lifetime with increasing $K_{\max}$ remains. The performance of MPF is close to that of JUP, but JUP still outperforms all the counterparts.

Figure 15 shows the network lifetime for non-disjoint set covers with the most extreme settings in our experiments ($\Delta = 30$ and $K_{\max} = 6$). These results exhibit improved network lifetime (as expected) when compared with the results shown in Fig. 11. A distinct difference is that JUP now slightly outperforms MPF, in contrast to the superiority of MPF over JUP as indicated in Fig. 11. Generally speaking, JUP and MCF both score the highest, followed by MCF and then MCSF. However, the performance gap between the best and the worst methods is shrunk due to better utilization of energy by all methods when non-disjoint set covers instead of disjoint set covers are generated. This again confirms that formation rules for non-disjoint set covers are less important to network lifetime than they are for disjoint set covers.

5. CONCLUSIONS

We have proposed a target coverage game for work scheduling in WSNs. The utility function of each player (sensor) has been designed to guarantee that the game always ends up with a state of Nash equilibrium that meets the Q-coverage requirement and is Pareto optimal. Also, no redundant sensors are active in this state. Implementation issues such as acquisition of neighbor status, simultaneous decision making, non-uniform energy levels, and game convergence time have been addressed, which makes the game design more practical. When turning the game into a practical protocol, backoff timers are introduced to give decision priority to players with a high utility gain or high energy level. Such a design can respectively shorten the game convergence time or prolong the network lifetime. Extended simulations were done to investigate

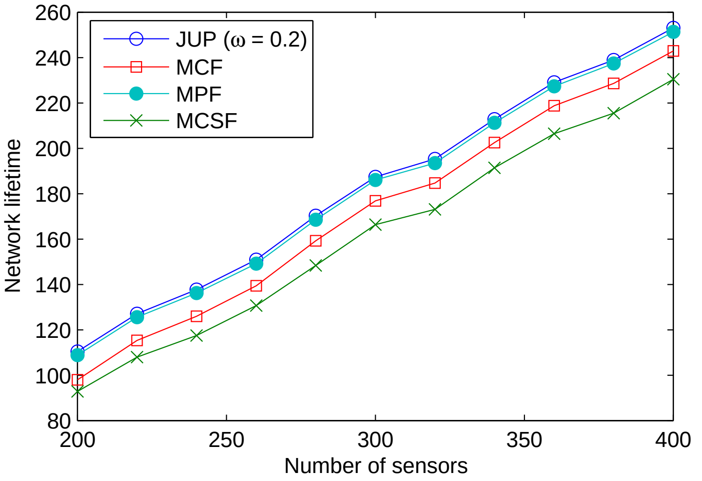

Fig. 15. Network lifetime for non-disjoint set covers with $\Delta = 30$ and $K_{\max} = 6$

the performance of various design alternatives and to compare the proposed game approach with existing greedy-based heuristics. For disjoint set covers, the proposed approach is inferior only to the power-based heuristic in terms of network lifetime. For non-disjoint set covers, the proposed approach results in the longest network lifetime among all alternatives considered in all settings. In short, we have demonstrated the feasibility and efficiency of applying game theory to the target coverage problem with non-uniform coverage requirements and sensor energy levels.

REFERENCES

AI, X., SRINIVASAN, V., AND THAM, C.-K. 2008. Optimality and complexity of pure Nash equilibria in the coverage game. *IEEE Journal on Selected Areas in Communications 26,* 7, 1170–1182.

CARDEI, M. AND DU, D.-Z. 2005. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks 11,* 3, 333–340.

CARDEI, M., THAI, M. T., LI, Y., AND WU, W. 2005. Energy-efficient target coverage in wireless sensor networks. In *Proc. IEEE INFOCOM*.

CARDEI, M. AND WU, J. 2004. Coverage in wireless sensor networks. In *Handbook of Sensor Networks*, M. Ilyas and I. Magboub, Eds. CRC Press.

CHAMAM, A. AND PIERRE, S. 2007. Dynamic sensor activation for maximizing network lifetime under coverage constraint. In *Proc. Int'l Symp. on Comm. and Information Technologies*. Sydney, 971–976.

CHAUDHARY, M. AND PUJARI, A. K. 2009. Q-coverage problem in wireless sensor networks. In *Lecture Notes in Computer Science 5408*. Springer-Verlag, 325–330.

FAN, G., LIANG, F., AND JIN, S. 2008. An efficient approach for point coverage problem of sensor network. In *Int'l Symp. on Electronic Commerce and Security*. 124–128.

FÉLEGYHÁZI, M., HUBAUX, J.-P., AND BUTTYÁN, L. 2006. Nash equilibria of packet forwarding strategies in wireless ad hoc networks. *IEEE Trans. on Mobile Computing 5,* 5.

GU, Y., LIU, H., AND ZHAO, B. 2007. Target coverage with QoS requirements in wireless sensor networks. In *The 2007 Int'l Conf. on Intelligent Pervasive Computing*. Jeju City, 35–38.

HAN, Z., JI, Z., AND LIU, K. J. R. 2005. Fair multiuser channel allocation for OFDMA networks using Nash bargaining and coalitions. *IEEE Trans. Commun. 53,* 8, 1366–1376.

HAN, Z., JI, Z., AND LIU, K. J. R. 2007. Non-cooperative resource competition game by virtual referee in multi-cell OFDMA networks. *IEEE Journal on Selected Areas in Communications 25,* 6, 1079–1090.

HAN, Z., PANDANA, C., AND LIU, K. J. R. 2005. A self-learning repeated game framework for optimizing packet forwarding. In *IEEE Wireless Communications and Networking Conference*. 2131–2136.

HEFEEDA, M. AND BAGHERI, M. 2007. Randomized k-coverage algorithms for dense sensor networks. In *Proc. IEEE INFOCOM*. Anchorage, USA, 2376–2380.

LI, Y. AND GAO, S. 2008. Designing k-coverage schedules in wireless sensor networks. *Journal of Combinatorial Optimization 15*, 127–146.

MONDERER, D. AND SHAPLEY, L. S. 1996. Potential games. *Games and Economic Behavior 14*, 124–143.

RASTI, M., SHARAFAT, A. R., AND SEYFE, B. 2009. Pareto-efficient and goal-driven power control in wireless networks: A game-theoretic approach with a novel pricing scheme. *IEEE/ACM Trans. Networking 17,* 2, 556–569.

ROSENTHAL, R. W. 1973. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory 2,* 1, 65–67.

SARAYDAR, C. U., MANDAYAM, N. B., AND GOODMAN, D. J. 2002. Efficient power control via pricing in wireless data networks. *IEEE Trans. Commun. 50,* 2, 291–303.

SIMON, G., MOLNÁR, M., GÖNCZY, L., AND COUSIN, B. 2007. Dependable k-coverage algorithms for sensor networks. In *Instrumentation and Measurement Technology Conf. Proc.*

XIAO, M., SHROFF, N. B., AND CHONG, E. K. P. 2003. A utility-based power-control scheme in wireless cellular systems. *IEEE/ACM Trans. Networking 11,* 2, 210–221.

XING, G., WANG, X., ZHANG, Y., LU, C., PLESS, R., AND GILL, C. 2005. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Trans. on Sensor Networks 1,* 1, 36–72.

YAÏCHE, H., MAZUMDAR, R. R., AND ROSENBERG, C. 2000. A game theoretical framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Trans. Networking 8,* 5, 667–678.

YEN, L.-H., LI, J.-J., AND LIN, C.-M. 2011. Stability and fairness of AP selection games in IEEE 802.11 access networks. *IEEE Trans. on Vehicular Technology 60,* 3, 1150–1160.

ZHANG, H. AND HOU, J. C. 2005. Maintaining sensing coverage and connectivity in large sensor networks. *Wireless Ad Hoc and Sensor Networks: An International Journal 1,* 1-2, 89–123.

ZHU, M. AND MARTINEZ, S. 2009. Distributed coverage games for mobile visual sensors (i): Reaching the set of Nash equilibria. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*. Shanghai, China, 169–174.