A Novel Distributed Algorithm for Complete Targets Coverage in Energy Harvesting Wireless Sensor Networks

Changlin Yang
School of Electrical, Computer and Telecommunications Engineering
University of Wollongong
Email: cy116@uowmail.edu.au

Kwan-Wu Chin
School of Electrical, Computer and Telecommunications Engineering
University of Wollongong
Email: kwanwu@uow.edu.au

***Abstract*—A fundamental problem in energy harvesting Wireless Sensor Networks (WSNs) is to maximize coverage, whereby the goal is to capture events of interest that occur in one or more target areas. To this end, this paper addresses the problem of maximizing network lifetime whilst ensuring all targets are monitored continuously by at least one sensor node. Specifically, we will address the Distributed Maximum Lifetime Coverage with Energy Harvesting (DMLC-EH) problem. The objective is to determine a distributed algorithm that allows sensor nodes to form a minimal set cover using local information whilst minimizing missed recharging opportunities. We propose an eligibility test that ensures the sensor nodes with higher energy volunteer to monitor targets. After that, we propose a Maximum Energy Protection (MEP) protocol that places an on-duty node with low energy to sleep while maintaining complete targets coverage. Our results show MEP increases network lifetime by 30% and has 10% less redundancy as compared to two similar algorithms developed for finite battery WSNs.**

***Index Terms*—Wireless Sensor Networks, Targets Coverage, Energy Harvesting, Distributed Algorithm**

I. Introduction

In a Wireless Sensor Network (WSN), each sensor node monitors its surrounding and routes sensed data to a gateway/sink via multi-hop communications wirelessly. In turn, the gateway/sink transmits the data to end users. A fundamental problem in conventional WSNs is their finite battery lifetime. In order to maximize the lifetime of a WSN and ensure all targets are monitored, one solution is to disperse many sensor nodes around a target. In such dense and randomly deployed WSNs, each target may be monitored by multiple sensor nodes; i.e., a sensor node may cover several targets. Moreover, these sensor nodes can either be in active or sleep state. Active sensor nodes are responsible for monitoring targets, and to transmit/forward data. Sensor nodes in sleep mode conserve energy and are called upon whenever requested. Recently, energy harvesting technologies have been applied in WSNs to convert ambient energy to electrical energy. The converted energy drives the load or recharges a node's battery. In theory, a WSN with energy harvesting nodes can operate perpetually, assuming energy neutral operation [1] and no other failures.

In this paper, we consider the Distributed Maximum Lifetime Coverage with Energy Harvesting node (DMLC-EH) problem [2]. The aim is to maximize a WSN's lifetime whilst ensuring all targets are monitored by at least one sensor node. Here, *lifetime* is defined as the time duration from when the WSN starts operation to the time when a target is not watched by any sensor nodes. Unlike the DMLC problem for finite WSNs that focus on reducing energy consumption, we aim to determine a recharging aware, distributed algorithm that utilizes energy efficiently. In particular, the key objective is to balance usage/demand and harvested energy.

To this end, we propose a Maximum Energy Protection (MEP) algorithm to address the DMLC-EH problem. The main idea is to replace on-duty sensor nodes with those currently in sleep state that have a higher energy level. Also, we consider the recharging capability of sensor nodes and ensure they do not lose any recharging opportunities. This occurs when a sensor node has a full battery, and therefore, is unable to store additional energy. We also propose an *eligibility test* to reduce redundancy, and thereby, minimize energy wastage. Finally, we show that MEP increases network lifetime by 30 % as compared to similar algorithms developed for finite battery WSNs. Moreover, it achieved said performance with 10% less redundancy.

This paper is structured as follows. Section II discusses related works. Section III introduces key definitions and notations. Then, we propose our solutions in Section IV. Section V presents our experiments and results. The paper concludes in Section VI.

II. Related Works

To date, past studies only consider the DMLC problem in conventional WSNs [2], in which, the operation time of sensor nodes is limited by their battery capacity. Therefore, the most common approach to minimize energy usage is to reduce redundancy by turning off overlapping sensor nodes [3][4][5][6][7][8] and [9]. In terms of operation, these algorithms can be classified into two groups: *global reshuffle* and *self-configuration*.

Global reshuffle algorithms operate in equal length rounds [3][6][9]. Each round consists of a configuration and a sensing phase. In particular, the length of the configuration phase is negligible as compared to the sensing phase. During the configuration phase, all sensor nodes exchange their location and targets information with their neighbors and decide their status in the sensing phase. Tian et al. [3] propose a Coverage-Preserving Node Schedule (CPNS) scheme that allows a sensor node to turn itself off without reducing the overall coverage if its targets are covered by other sensor nodes. Otherwise, it activates itself in the sensing phase. The authors of [9] propose a scheme whereby sensor nodes with a higher residual energy volunteers to cover a target. This is achieved by organizing sensor nodes with a lower residual energy to make decision earlier. Their decision scheme is similar to [3], whereby sensor nodes switch themselves off if all their targets are covered by other sensor nodes. Yan et al. [6] propose to schedule the activation time of sensor nodes during the configuration phase. In their algorithm, sensor nodes monitoring the same targets will build a schedule so that they are activated in turns and the total working time is equal to the duration of the sensing phase. In addition, sensor nodes covering multiple targets build a schedule for each target and their activation schedules are the sum of the individual schedule of all their covered targets. The main drawback of global reshuffle algorithms is their high signalling overheads at the beginning of each round. Moreover, if a sensor node unexpectedly fails, it can only be detected until the next reshuffle.

The works in [8][4] and [5] entail sensor nodes self-configuring their state as per local information. A sensor node can be in the probing state to exchange information with other active sensor nodes. The PEAS algorithm proposed in [5] allows probing nodes to enter the active state if there is no working node within their sensing range. Otherwise, they calculate their sleep time according to the working nodes' residual energy and the number of nodes in sleep state. Islam et al. [4] propose a scheme to maintain area coverage whereby probing sensor nodes will activate themselves if their distance to a working node is longer than a certain length. In general, a sleeping sensor node needs to activate itself periodically to enter the probing state [5]. This results in a coverage hole when a node exhausts its energy. That is, targets previously covered by the dead node remain uncovered until a probing sensor node detects them. To this end, one solution developed in [8] and [4] is to equip sensor nodes with a trigger circuit [10]. Once a node exhausts its energy, it wakes up all sensor nodes within its communication range to enter the probing state.

Unlike past works, we consider the recharging capability of sensor nodes. We want to ensure sensor nodes will not lose recharging opportunity. This is an important consideration as it ensures harvested energy is used to prolong coverage lifetime. Also, we do not use any trigger mechanism, which helps save cost. In addition, our solution does not require all nodes to have synchronous wake-up time, which is impractical in large scale WSNs.

III. Preliminaries

We consider energy harvesting WSNs comprising of sensor nodes placed on the Euclidean plane that can be in active (on-duty) or sleep (off-duty) state. We use s_i and z_j to index sensor nodes and targets, where $i = 1 \dots |S|$ and $j = 1 \dots |Z|$. Let E_i (Joules) be the current energy of sensor node s_i, which is bounded by its battery capacity B. Each sensor node s_i has a recharging rate of E_i^r (Joule/s) and an energy consumption rate of E_i^c (Joule/s). We define $Z(s_i)$ to be a function that returns the set of targets covered by sensor s_i. That is, sensor node s_i is able to monitor $Z(s_i)$ targets. Conversely, $S(z_j)$ is a function that returns the set of sensors covering target z_j. The following are two key definitions used throughout the paper:

Definition 1. *Coverage lifetime is the duration in which sensor nodes start monitoring targets until they fail to monitor these targets due to the lack of energy.*

Definition 2. *Complete target coverage is achieved when all targets are covered by at least one sensor node at all times.*

We assume all sensor nodes have the same sensing range and able to monitor all targets within their sensing range. The communication range of each sensor node is twice its sensing range. We define the *neighbour* of sensor node s_i, denoted as $N(s_i)$, to be the sensor nodes within its communication range. Therefore, we have the following proposition:

Proposition 1. *The neighbours of sensor node s_i are those that cover at least one common targets as s_i.*

Proof: Assume the sensing range of sensor node s_i is r and its communication range is $2r$. If a target z_j is within the sensing range of s_i, then let $d_{ij} \leq r$ be the distance between s_i and z_j. Assume another sensor node s_k that also covers target z_j and its distance is $d_{kj} \leq r$. Therefore, by the triangle inequality, the distance between sensor node s_i and s_k is $d_{ij} + d_{kj} \leq 2r$ which does not exceed the communication range of these two nodes. ■

Given the said proposition, the neighbours of a sensor node s_i can be expressed as

$$N(s_i) = S(Z(s_i)), \quad \forall s_i \in S \tag{1}$$

We also define a *reshuffle phase* to be the time interval from t to $t+\phi$ where all involved sensor nodes make decision as to whether to go to sleep. The length of ϕ is much less than a sensor node's battery lifetime. Here, decisions are made following an *off-duty* rule, which we use to remove overlapping sensor nodes [3][6][9]. That is,

Off-duty rule. *If all targets within the sensing range of sensor node s_i are covered by a sub-set of its neighbours $N(s_i)$, then s_i can turn itself off without reducing the overall targets coverage. Otherwise, s_i goes into the active state.*

At the beginning of each reshuffle phase, all sensor nodes involved first exchange their epoch and targets information; note, *epoch* is defined in Section IV which is the time that a sensor node starts its next reshuffle phase. Then, each sensor

node decides its status according to the off-duty rule after a delay T calculated based on its residual energy level as follows,

$$T_i = \frac{E_i}{B_i} \times \phi, \quad \forall i \in S \tag{2}$$

where B_i is the battery capacity of sensor node s_i, and E_i is its current energy level, and ϕ is the reshuffle phase duration. Equ. 2 implies that sensor nodes with a higher residual energy will volunteer to cover a target last. This allows sensor nodes with less residual energy to perform the eligibility test first, and thus, will often yield responsibility of monitoring targets to sensor nodes with ample energy.

A coverage hole may exist if more than one sensor node makes a decision simultaneously [11]. Consider two sensor nodes A and B that monitor the same targets. If A and B have the same current energy level during the reshuffle phase, they will both go to sleep by assuming another node is active. To this end, Equ. 2 can be rewritten as $T_i = \frac{E_i}{B_i} \times \phi + \tau$, where τ is a random value much less than ϕ. At the end of the reshuffle phase, only a subset of sensor nodes is activated to carry out the sensing task while other sensor nodes enter the sleep state to save energy and recharge.

IV. Solution

The main objective is to determine a distributed rule to be used by each sensor node such that all targets are continuously monitored for maximum time duration. First, define *epoch*, denoted as δ_i, to be the time that sensor node s_i starts its next reshuffle phase. It is worth pointing out that in conventional global reshuffle algorithms [3][6][9], the epoch of all sensor nodes are synchronized and have the same length. In our approach, however, the epoch of each sensor node is adjustable according to its status as well as that of its neighbours. This is important because global synchronization incurs high signaling overheads.

Recall that we consider the recharging capability of sensor nodes. In particular, we aim to enable sensor nodes when their battery reaches capacity At such time, a sensor node volunteers to monitor target(s). However, if a sensor node voluntary wakes up whilst all targets within its sensing range are already monitored by other sensor nodes, then the node will have wasted its energy. To this end, we propose an *eligibility test* to choose a subset of sleep sensor nodes that are able to cover all targets monitored by an on-duty node. That is, the test determines which sensor nodes in sleep state can be used to replace the current on-duty node without reducing network coverage.

Henceforth, we first define *eligibility test*. After that, we present a novel, distributed Maximum Energy Protection (MEP) algorithm that allows a subset of sensor nodes to enter each reshuffle phase. That is, only a subset of sensor nodes contend to form a minimal set cover to monitor all targets.

A. Eligibility Test

The goal of the eligibility test is to determine whether a sensor node in sleep state is *valuable* and hence it is to be activated. Here, valuable means a sensor node that is currently in sleep state is able to form a subset with other sensor nodes that cover all targets monitored by an on-duty sensor node. Assume sensor node s_i is active with an epoch of δ_i. Let $Z(N(s_i))$ denote the targets covered by its neighbours and $\delta_{N(s_i)}$ represents its neighbours' epoch. These information are stored in an array $\mathcal{N}_{s_i}$. Here, we define $subN(s_i)$ to be a subset of neighbouring nodes of s_i that are able to cover all targets monitored by s_i, i.e., $Z(s_i) \subset Z(subN(s_i)), subN(s_i) \subset N(s_i)$. Let s_j be a neighbour of s_i and also s_j is currently in the sleep state. When s_j reaches its epoch and enters a reshuffle phase, s_i will do the following two-stage test:

1) If $\delta_j > \delta_i$, then s_j passes, and the eligibility test ends. Otherwise, node s_j continues with the second stage test.
2) If s_j is in the subset $subN(s_i)$, meaning it covers all targets monitored by sensor node s_i, then s_j passes the eligibility test.

If s_j passes the aforementioned critera in the first stage, it updates its epoch to that of s_i, which is δ_i. If s_j passes at the second stage, this means s_i can be turned off by activating a subset of its neighbours $subN(s_i)$ without reducing coverage. Therefore, s_i updates its epoch to equal the maximum epoch in set $subN(s_i)$. Then, sensor node s_j also updates its epoch to equal the new δ_i. However, if s_j fails the eligibility test, both s_i and s_j retain their own epoch. When a sensor node passes the eligibility test for two or more on-duty neighbours, it will set its epoch to the earliest one.

According to the eligibility test, we have the following proposition:

Proposition 2. *Sensor node s_i and a subset of its neighbours $subN(s_i)$ will enter the reshuffle phase at or before the time instant δ_i.*

Proof: By applying test one, all neighbours of s_i have an epoch no longer than δ_i. After s_i updates its epoch to the maximum epoch in set $subN(s_i)$, any neighbours of s_i reach their epoch and enter the reshuffle phase before δ_i will then update their epoch. That is, sensor nodes in $subN(s_i)$ as well as s_i will enter the reshuffle phase before time δ_i. ■

As discussed above, a sensor node will only be activated if it is in a subset of sensor nodes that is able to cover all targets monitored by an on-duty node. In the next section, we introduce our novel distributed algorithm to solve the DMLC-EH problem.

B. Maximum Energy Protection Algorithm

We now describe MEP. At system initialization, all sensor nodes enter a global reshuffle phase and decide their status based on the off-duty rule to remove redundant sensor nodes. If a sensor node s_i decides to go into the sleep state, it sets its epoch to $\delta_i = \frac{B - E_i}{E_i^r}$ and broadcast a 'TURN OFF' message containing its epoch and targets. On the other hand, sensor nodes deciding to be active will set their epoch to their current battery lifetime and initialize an empty array $\mathcal{N}_{s_i}$. As will

become clear later, this array records the epoch and targets information of its neighbours.

When an on-duty sensor node receives a 'TURN OFF' message from its neighbour, it adds the neighbour node into the array $\mathcal{N}_{s_i}$ and applies the eligibility test to it. The on-duty node will then reply with a 'SET EPOCH' message if the neighbour passed the eligibility test. Otherwise, the on-duty node does not reply.

A sensor node receiving two or more 'SET EPOCH' messages will set its epoch to equal the one that occurs earliest in time, see Section IV-A. However, if it decides to be active at its epoch, there will be redundancy because it will be monitoring the same targets as the on-duty node with a later epoch. For example, if s_i receives two 'SET EPOCH' messages from its two on-duty neighbours A and B with epoch $\delta_A = 5$ and $\delta_B = 8$, sensor node s_i will set its epoch to 5 and enters the reshuffle phase at such time. If sensor node s_i decides to be active, there will be redundancy because node B still has three units of time before reaching its epoch. Therefore, in order to reduce redundancy, a sensor node that activates itself will broadcast a 'CHECK REDUNDANT' message to its on-duty neighbours. When an on-duty sensor node receives said message, it enters the reshuffle phase immediately to decide its status. Algorithm 1 shows the pseudo-code of MEP. In line 7, sensor node s_i will calculate the subset $subN(s_i)$ based on its observation of $\mathcal{N}_{s_i}$.

Algorithm 1: Pseudo-code of MEP

1 Input: $\delta_{i,t}$
2 Output: $\delta_{i,t+1}$
3 **if** *Any target* $t \in Z(s_i)$ *is not monitored by other sensor node* **then**
4 Change state to on-duty and set $\delta_{i,t+1} = \frac{E_i}{E_i^c}$
5 Set $\mathcal{N}_{s_i} = \emptyset$
6 Upon receiving a 'TURN OFF' message from s_j **if** $Z(s_i) \subset Z(s_j) \cup Z(N(s_i))$ **then**
7 Find $subN(s_i)$ such that $Z(s_i) \subset Z(subN(s_i)) \cup Z(s_j)$
8 Set $\delta_{i,t+1}$ to the maximum epoch in $subN(s_i)$
9 Send a 'SET EPOCH' message to s_j
10 **end**
11 **if** $\delta_{i,t+1} < \delta_{j,t+1}$ **then**
12 Send a 'SET EPOCH' message to s_j
13 **end**
14 Add s_j into $\mathcal{N}_{s_i}$
15 Upon receiving a 'CHECK REDUNDANT' message **if** *All targets* $Z(s_i)$ *are monitored by other sensor nodes* **then**
16 Jump to line 20
17 **end**
18 **end**
19 **if** *All traget* $Z(s_i)$ *are monitored by other sensor nodes* **then**
20 Set $\delta_{i,t+1} = \frac{B - E_i}{E_i^r}$
21 Broadcast a 'TURN OFF' message
22 **end**
23 **if** *Received a reply message from node* s_j **then**
24 $\delta_{i,t+1} = \delta_{j,t}$
25 Change state to off-duty
26 **end**

V. Evaluation

We verify MEP using the parameters of the WaspMote [12] platform, which consumes 60 mW when in the active state and 0.2 mW when sleeping. All sensor nodes are equipped with an Enocean ECS310 solar cell [13]. We assume it has a conversion rate of 10% and a recharging efficiency of 50%, which is conservative as compared to other technologies [14]. In addition, we use real solar irradiance data retrieved from Southwest Solar Research Park, Phoenix, Arizona, USA [15] on the 16-th of April 2013. All experiments are simulated using Matlab running on an Intel Core i7 CPU @ 3.5GHz with 8 G RAM computer. The parameter values used in our experiments can be found in Table I.

Parameters	Value
Battery size	1100 mA
Consumption rate	3.6 Joules/hour
Average recharge rate	0.96 Joules/hour
Voltage	4 V
Solar panel conversion rate	10 %
Recharging efficiency	50 %
Transmission cost	0.1 Joules

TABLE I
Simulation Parameters

We compare MEP to Coverage-Preserving Node Schedule (CPNS) [3] and Deterministic Energy-Efficient Protocol for Sensing (DEEPS) [8]. CPNS operates in equal length rounds. All sensor nodes decide their status at the beginning of each round. On the other hand, DEEPS allows each sensor node to decide its status according neighbour knowledge. However, a sensor node using DEEPS will operate until it exhausts all its energy; see Section II. We assume all sensor nodes are stationary and randomly located on a square area. Note that we do not consider the energy consumed due to sensing and forwarding data. However, they can be considered by scaling the available energy at each node to account for monitoring targets or by reducing the recharging rate of sensor nodes. We also assume each sensor is equipped with a timer to record its epoch. The said timer is able to trigger a sensor node that is in the sleep state to enter the reshuffle phase.

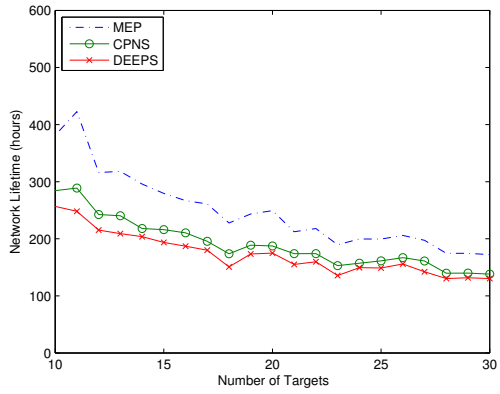
Our results are an average of 50 runs, each with a random generated topology, which is sufficient to present the difference in terms of network lifetime according to the previous research [16]. Each sensor node has a maximum 76.6 hours worth of energy. However, a network may operate perpetually if there are sufficient number of sensor nodes [1]. We define a network lifetime upper bound of 3000 hours, at which time we assume the network is operating perpetually. We observe the impact of the following parameters: *target density*, *node density* and

sensing range. In each experiment, we collected the following metrics:

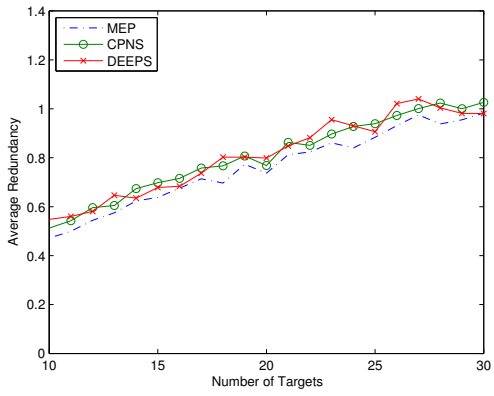
- *Network lifetime*. This is the time duration from which a network starts operation to when a target is not watched by any sensor nodes.
- *Average redundancy*. This is the average redundancy for each target. It represents the average number of overlapping sensor nodes for each target. Any on-duty sensor nodes monitoring a target covered by other on-duty nodes will increase average redundancy.

A. Results

1) Target Density: In the first experiment, we study the impact of target density on the lifetime and average redundancy of MEP, CPNS and DEEPS. We deploy 10 sensor nodes within a $1000 \times 1000 m^2$ field, each sensor node has a uniform sensing range of 500 meters. We then increase the number of deployed targets from 10 to 30.



(a)

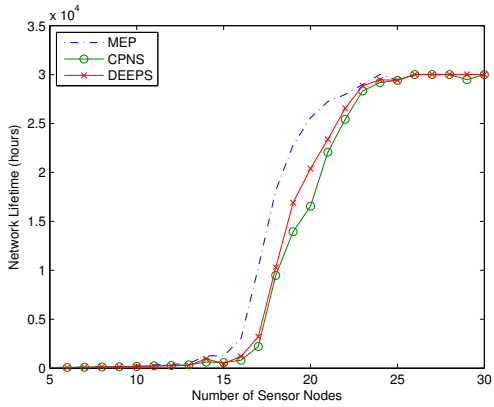


(b)

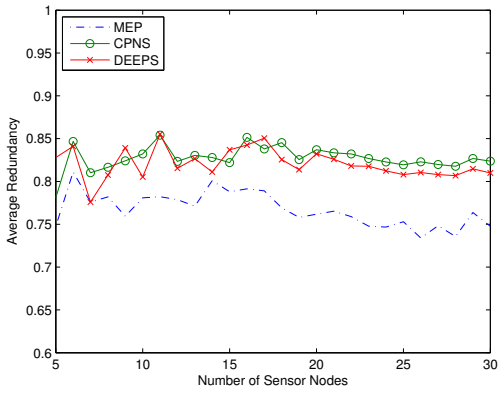
Fig. 1. Different target densities versus (a) network lifetime and (b) average redundancy

From Figure 1(a), we see that the network lifetime decreases when we add more targets. The reason is that as the number of targets increases from 10 to 30, each sensor node covers more targets within its sensing range. As a result, it has fewer opportunities to enter the sleep state to recharge itself, which reduces network lifetime. Additionally, in Figure 1(a), we can also see that the network lifetime of MEP is significantly higher than CPNS and DEEPS; i.e., more than 30%. The reason is that less energy is wasted when using MEP due to loss energy harvesting opportunity. Figure 1(b) shows the average redundancy increasing with the number of targets. This is because more targets are in the overlapping region of multiple sensor nodes. Moreover, MEP has 0.25 less average redundancy as compared to CPNS and DEEPS. This is due to MEP's eligibility test, which helps reduce redundancy. On the other hand, the redundancy of CPNS and DEEPS remains until next round or an overlapped sensor node exhaust its energy.

2) Node Density: In this experiment, we fix the number of targets to 20 and vary the number of sensor nodes from five to 30 - both sensor nodes and targets are dispersed within $1000 \times 1000 m^2$ field. All sensor nodes also have a uniform sensing range of 500 meters.



(a)



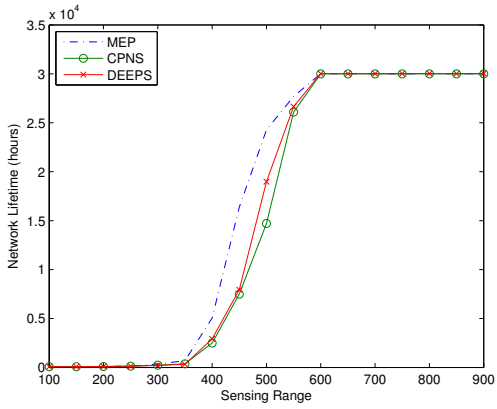
(b)

Fig. 2. Sensor node densities versus (a) network lifetime, and (b) average redundancy.

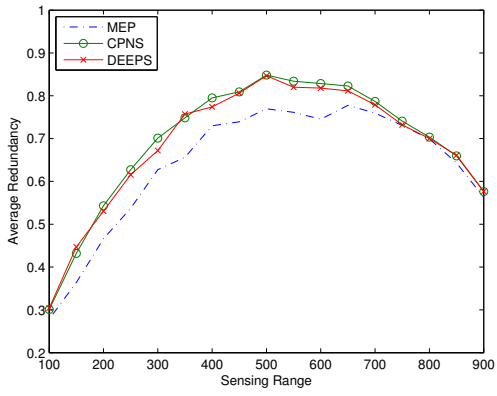
Figure 2(a) shows that when the number of sensor nodes increases from 15 to 25, the network lifetime of MEP, CPNS and DEEPS rapidly increases from less than 1000 hours to perpetual operation. The reason is that sensor nodes have more opportunities to be in the sleep state, which increases harvested energy. In this experiment, MEP outperforms CPNS

and DEEPS. In particular, it first achieves perpetual operation with one less sensor node. Figure 2(b) shows that the average redundancy of MEP is less than CPNS and DEEPS. Moreover, the average redundancy of MEP reduced from 0.8 to 0.75 as the number of sensor nodes increases while the redundancy of CPNS and DEEPS remain at 0.82. That is, MEP achieves 30% longer lifetime and 10% lower redundancy than CPNS and DEEPS.

3) Sensing Range: We now study the effect of sensing range on network lifetime. We disperse 20 sensor nodes and targets within a $1000m^2$ field. We then vary sensor nodes' sensing range from 100 to 900 meters. From Figure 3(a), we



(a)



(b)

Fig. 3. Varying sensing ranges versus (a) network lifetime, and (b) average redundancy.

see that the network lifetime of all these algorithms starts rising when the sensing range is 350 meters and achieve perpetual when the sensing range equals 600 meters. Networks using MEP are able to have same lifetime compared with CPNS and DEEPS but with sensor nodes have 50 meters less sensing range.. That is, it requires one less sensor node to achieve perpetual operation, which agrees with the results presented in Section V-A2. Figure 3(b) shows that the average redundancy peaked when the sensing range is 500 meters. The reason is that, when sensing range is small, each sensor node only covers a small number of targets and overlaps less with other nodes. When the sensing range is large, each sensor node will cover almost all targets in the sensing field. That is, it does not need to activate many sensor nodes to maintain coverage, and thereby, reduce redundancy.

VI. Conclusion

This paper is the first to propose a distributed solution to the maximum lifetime coverage problem in energy harvesting WSNs. The proposed solution allows only a subset of sensor nodes to enter the reshuffle phase to maintain complete targets coverage. Moreover, it considers the recharging rate of different sensor nodes and does not need global synchronization. Simulation results show that MEP increases network lifetime by at least 30% and reduces network redundancy by 10% as compared to other studied algorithms.

References

[1] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 6, no. 4, p. 32, 2007.

[2] B. Wang, *Coverage Control in Sensor Networks*. Springer, 2010, ch. Coverage Lifetime Maximization, pp. 65–95.

[3] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 271–290, 2003.

[4] M. N. Islam, Y. M. Jang, S. Choi, S. Park, and H. Park, "Redundancy reduction protocol with sensing coverage assurance in distributed wireless sensor networks," in *9th International Symposium on Communications and Information Technology*, Icheon, Sept 2009, pp. 631–636.

[5] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: A robust energy conserving protocol for long-lived sensor networks," in *23th International Conference on Distributed Computing System*, Providence, May 2003, pp. 28–37.

[6] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *First Intl. Conf. on Embedded Networked Sensor System*, Los Angeles, November 5-7 2003, pp. 51–62.

[7] S. K. Prasad and A. Dhawan, "Distributed algorithms for lifetime of wireless sensor networks based on dependencies among cover sets," in *IEEE HiPC*, 2007, pp. 381–392.

[8] D. Brinza and A. Zelikovsky, "Deeps: Deterministic energy-efficient protocol for sensor networks," in *7th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, Las Vegas, June 2006, pp. 261–266.

[9] H. Zhang, H. Wang, and H. Feng, "A distributed optimum algorithm for target coverage in wireless sensor networks," in *Asia-Pacific Conference on Information Processing, 2009.*, vol. 2. Shenzhen: IEEE, July 2009, pp. 144–147.

[10] L. Gu and J. A. Stankovic, "Radio-triggered wake-up capability for sensor networks," *Real-Time System*, vol. 29, no. 2-3, pp. 157–182, March 2005.

[11] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *IEEE ICC*, vol. 2, Helsinki, 2001, pp. 472–472.

[12] "Waspmote datasheet," http://www.libelium.com/downloads/documentation/waspmote_datasheet.pdf.

[13] "Enocean ecs 310," http://www.enocean.com/en/enocean_modules/ecs-310/.

[14] A. C. Valera, W.-S. Soh, and H.-P. Tan, "Energy-neutral scheduling and forwarding in environmentally-powered wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1202–1220, 2013.

[15] "Nrel: Southwest solar research park," http://www.nrel.gov/midc/ssrp/.

[16] C. Yang and K. Chin, "Novel algorithms for complete targets coverage in energy harvesting wireless sensor networks," *IEEE Communications Letters*, vol. 18, pp. 118–121, Jan 2014.