January 2015

On complete targets coverage and connectivity in energy harvesting wireless sensor networks

Changlin Yang
cy116@uowmail.edu.au

Kwan-Wu Chin
University of Wollongong, kwanwu@uow.edu.au

Follow this and additional works at: https://ro.uow.edu.au/eispapers

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
On complete targets coverage and connectivity in energy harvesting wireless sensor networks

Abstract
In energy harvesting wireless sensor networks, ensuring complete targets coverage is a fundamental problem. In particular, targets are required to be monitored by at least one sensor node at all times. To date, past works have proposed to schedule sensor nodes alternately in the active/sleep state to maximize network lifetime whilst maintaining complete targets coverage, and affording sensor nodes recharging opportunities. However, they do not consider connectivity to the sink. We first propose a Linear Programming (LP) based solution to determine the activation time of sensor nodes. The design constraints include complete targets coverage, energy, and flow conservation to ensure data from sensor nodes monitoring targets are able to forward their data to the sink. We also propose an efficient heuristic algorithm as the LP solution requires an exhaustive collection of set covers. The heuristic iteratively selects sensor nodes to monitor targets and forward sensed data according to their residual energy. The simulation results show that the heuristic algorithm achieves 80% of the network lifetime computed by the LP solution at a fraction of the computation time.

Keywords
energy, sensor, harvesting, networks, connectivity, coverage, targets, wireless, complete

Publication Details

This conference paper is available at Research Online: https://ro.uow.edu.au/eispapers/4704
On Complete Targets Coverage and Connectivity 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—In energy harvesting wireless sensor networks, ensuring complete targets coverage is a fundamental problem. In particular, targets are required to be monitored by at least one sensor node at all times. To date, past works have proposed to schedule sensor nodes alternately in the active/sleep state to maximize network lifetime whilst maintaining complete targets coverage and affording sensor nodes recharging opportunities. However, they do not consider connectivity to the sink. We first propose a Linear Programming (LP) based solution to determine the activation time of sensor nodes. The design constraints include complete targets coverage, energy, and flow conservation to ensure data from sensor nodes monitoring targets are able to forward their data to the sink. We also propose an efficient heuristic algorithm as the LP solution requires an exhaustive collection of set covers. The heuristic iteratively selects sensor nodes to monitor targets and forward sensed data according to their residual energy. The simulation results show that the heuristic algorithm achieves 80% of the network lifetime computed by the LP solution at a fraction of the computation time.

Index Terms—Wireless Sensor Networks, Energy Harvesting, Targets Coverage, Network Connectivity

I. INTRODUCTION

Energy harvesting Wireless Sensor Networks (WSNs) are ideal for monitoring an environment and capturing events of interest that occur in one or more target areas. In this respect, there are two fundamental problems: complete targets coverage and network connectivity. In the complete targets coverage problem, each target must be covered by at least one sensor node at all times. This means any event that occurs in a given target area must be recorded. In practice, one sensor node is not sufficient to continuously monitor a target. This is because sensor nodes have a significantly higher energy consumption rate than its recharging rate [1]. To this end, one solution is to densely deploy sensor nodes around targets and schedule their sleep and wake schedule to allow them time to recharge as well as monitor targets. This is usually achieved by dividing sensor nodes into set covers, whereby sensor nodes that belong to a cover are activated for a given time interval to monitor all targets. Other sensor nodes are placed in sleep mode, thereby allowing them to conserve and recharge their battery [2].

As for network connectivity, all activated sensor nodes must have a path to forward any data due to target monitoring to the sink. In this respect, one approach is to ensure all sensor nodes belonging to a set cover form a connected network. As shown in Figure 1, eight sensor nodes are used to monitor two targets indicated by a triangle. The sensing and communication range of a sensor node is the same and indicated by a circle around each sensor node. We can first activate sensor nodes in the cover \( C_1 = \{s_1, s_2, s_3, s_4\} \), where sensor nodes \( s_1 \) and \( s_3 \) are used for targets coverage whilst \( s_2 \) and \( s_4 \) are responsible for forwarding sensed data to \( B \). When the energy level of sensor nodes in \( C_1 \) is low, another cover \( C_2 = \{s_5, s_6, s_7, s_8\} \) can be activated to ensure complete targets coverage and network connectivity. Thus, the sensor nodes in cover \( C_1 \) can enter sleep mode.

In this paper, we consider the Maximum Lifetime Coverage and Connectivity with Energy Harvesting node (MLCCEH) problem. In particular, sensor nodes, targets and a sink are randomly dispersed on a given sensing field. The goal is to maximize network lifetime whilst ensuring complete targets coverage and network connectivity. Here, the network lifetime is defined as the duration from when a WSN starts operating until a target fails to be monitored or a sensor node is disconnected from the sink. To date, the Maximum Lifetime Coverage and Connectivity (MLCC) problem has been studied in the context of non-rechargeable WSNs [3][4][5][6][7] and [8]. In particular, the authors of [3][4] and [5] divide sensor nodes into set covers and formulate the MLCC problem as...
an Integer Linear Programming (ILP) with the objective of maximizing network lifetime subject to energy and connectivity constraints. In [6][7] and [8], the authors relax the ILP and present a Linear Program (LP) solution for the MLCC problem. Critically, these solutions do not consider recharging opportunities. In the context of energy harvesting WSNs, only a handful of works, e.g., [2][9], have considered the complete targets coverage problem. This so called Maximum Lifetime Coverage with Energy Harvesting node (MLCEH) problem, however, does not consider network connectivity to the sink; i.e., scheduling the wake-up time of sensor nodes in order to forward sensed data back to the sink.

In this paper, our aim is to determine the subset of sensor nodes and their corresponding activation time such that they are afforded recharging opportunities and the network is connected at all times. We remark that this problem has not been considered in the context of energy harvesting WSNs.

To this end, this paper makes the following contributions:

- We are the first to consider the MLCC problem in energy harvesting WSNs. Our aim is different to works on non-rechargeable WSNs in that we aim to maximize recharging opportunities.
- We first use exhaustive search [6] to generate a collection of set covers that provide coverage and connectivity. Then using a LP solver, we derive the minimal time, along with the active time of the corresponding sensor nodes, for each set cover. However, the computation time of this approach increases exponentially with the number of sensor nodes.
- We then propose an efficient energy conservation heuristic to solve the MLCEH problem. It iteratively selects sensor nodes to monitor targets and ensure network connectivity based on their residual energy. This significantly reduces the computation time and achieves 80% network lifetime as compared to the result obtained by our LP solution.

The rest of this paper is structured as follows. Section II present of our network model. Section III outlines our solution followed by evaluation and results in Section IV. The paper concludes in Section V.

II. NETWORK MODEL

We model a WSN as two bipartite graphs $G(S \cup Z, E)$ and $G'(S, N \cup N')$. In the graph $G(S \cup Z, E)$, $S$ and $Z$ are the set of sensor nodes and targets, respectively. The set of edges that connects a sensor $s_i \in S$ to one or more targets in $Z$ is denoted by $E$. A node $B$ in the set $S$ is called the sink. All sensor nodes in $S$ have at least one path to forward data to $B$ when all sensor nodes are in the active state.

As for the graph $G'(S, N \cup N')$, the set $N$ represents the edges connecting a sensor node $s_i \in S$ to one or more sensor nodes within its communication range. The set $N'$ contains edges that connect sensor nodes whose distance is less than the sensing range. Let $N(s_i)$ and $N'(s_i)$ return the set of sensor nodes within the communication and sensing range of sensor node $s_i$, respectively.

We assume time is discrete and the length of a time interval is $\delta_t$, where $t = 1, \ldots, T$. Let $\delta_T$ be the last time interval when a sensor node has a route to the sink. Thus, the network lifetime is equal to $\sum_{t=1}^{T} \delta_t$. Define $C_k \subseteq S$ to be a subset of sensor nodes, where $k = 1 \ldots K$. Let $\phi(C_k, s_i)$ be a function that returns one if sensor node $s_i$ is in the subset $C_k$, otherwise it returns zero. We use $c_t^k$ to indicate the activation time of subset $C_k$ in time interval $t$. Thus, the total activation time for sensor node $s_i$ in time interval $t$ is $x_t^i = \sum_{k=1}^{K} \phi(C_k, s_i)c_t^k$. Without loss of generality, we assume all sensor nodes generate $R$ bit/s when it monitors a target. Let $f_{th} > 0$ be the data flow rate from sensor node $s_i$ to $B$ at time interval $t$. Let $E_t^i \leq E_{max}$ be the battery level of sensor node $s_i$ at the beginning of the time interval $t$. Specifically, it is equal to the battery level of sensor node $s_i$ minus its consumed energy plus harvested energy in time interval $t - 1$. Mathematically,

$$E_t^i = E_{i}^{(t-1)} - (E_{i}^{(t-1)} + E_{cr}^{(t-1)} + E_{ct}^{(t-1)} + f_{ih}^{(t-1)} + f_{th}^{(t-1)})$$

(1)

where $E_{ct}^{(t)}$ (Joule/s) is the energy harvesting rate of sensor node $s_i$. The term $E_{cr}^{(t)}$ (Joule/s) is the energy consumption rate for all sensor nodes when active. Other the other hand, $E_{cr}(Joule)$ and $E_{ct}(Joule)$ correspond to the energy incurred by a sensor node to receive or transmit one bit, respectively. We write $E_{f}^{(t)}$ (Joule/bit), where $E_{f} = E_{cr} + E_{ct}$, to be the energy consumption rate of a sensor node when forwarding sensed data. We assume all sensor nodes have a full battery, i.e., $E_{max}$, at $t = 0$.

III. SOLUTION

In this section, we first present an optimal solution for the MLCEH problem using a LP, called LP-MLCEH. We then present a heuristic, called EC-MLCEH in Section III-B, that selects sensor nodes based on their residual energy level to ensure network coverage and connectivity.

A. LP-MLCEH

The objective of LP-MLCEH is to maximize network lifetime $\sum_{t=1}^{T} \delta_t$; i.e., maximize the value of $T$. To this end, it aims to minimize the activation time of sensor nodes in each time interval whilst maintaining complete targets coverage and network connectivity. This ensures sensor nodes have ample time to conserve and recharge their battery.

Given the bipartite graphs $G(S \cup Z, E)$ and $G'(S, N \cup N')$, LP-MLCEH first uses an exhaustive search to determine all possible subset of sensor nodes $C_k$ to cover all targets in $Z$ and these sensor nodes must have a path to the sink. The resulting collection of subsets are recorded in $\Omega = \{C_k | k = 1 \ldots K\}$.

After that, LP-MLCEH aims to minimize $\sum_{i=1}^{S} x_t^i$ in each time interval $t$ by determining the activation time of a subset of covers $\Gamma' \subseteq \Omega$ such that their energy constraint is not violated.
Mathematically, the LP for the problem at hand is as follows,

$$\min \sum_{t=1}^{T} \sum_{i=1}^{S} x_{i}^{t}$$

Subject to:

$$\sum_{k=1}^{K} \alpha_{k} \geq 1, \quad \forall t \in \{1, \ldots, T\} \quad (3)$$

$$0 \leq x_{i}^{t} \leq 1, \quad \forall i \in S, \quad \forall t \in \{1, \ldots, T\} \quad (4)$$

$$\sum_{j \in Z(s_{i})} x_{i}^{t} + \sum_{h \in N(s_{i})} f_{i}^{h} = \sum_{h \in N(s_{i})} f_{i}^{h} + f_{i}^{B}, \quad \forall i \in S, \quad \forall t \in \{1, \ldots, T\} \quad (5)$$

$$\sum_{i \in S} f_{i}^{B} = R \sum_{i \in S} x_{i}^{t}, \quad \forall t \in \{1, \ldots, T\} \quad (6)$$

$$E_{\text{max}} = \sum_{t=1}^{T} \left[ E_{a} x_{i}^{t} + E_{c} f_{i}^{h} + E_{t} f_{i}^{h} - E_{r}^{t} \right] \geq 0, \quad \forall s \in S \quad (7)$$

Constraint (3) ensures all targets are continuously covered in any time interval \(t\). Constraints (5) and (6) ensure flow conservation. Note, the data flow \(f_{i}^{B}\) will be zero if node \(s_{i}\) is not directly connected to the sink. Constraint (7) ensures each sensor node does not spend more than its available energy.

LP-MLCEH uses binary search to determine \(T\) by repeatedly solving the presented LP containing \(T \times |\Omega| \times |S|^2\) decision variables and \(T \times (|S| + |S| \times \Omega + 2|S|)\) constraints. In practice, the exhaustive search for all set covers \(\Omega\) is computationally intensive [6]. Moreover, the LP solution incurs a high computational cost due to multiple calls to a LP solver for each \(T\) value. To this end, in the next section, we propose an efficient heuristic solution for the MLCCEH problem.

B. Energy Conservation Heuristic (EC-MLCCEH)

We start with a few key definitions. Recall that we are given the bipartite graph \(G'(S, N \cup \Omega')\). Let \(S' \subseteq S\) be a set that contains only sensor nodes that have a path to the sink. We then calculate multiple paths from each sensor node \(s_{i}\) to \(B\). This can be achieved using Yen’s K shortest paths algorithm [10]. We use \(P_{i}^{k}\) to indicate the \(k\)-th path from sensor node \(s_{i}\) to \(B\). All paths are stored in a collection \(\Psi = \{P_{i}^{k} | \forall i \in S', k = 1, \ldots, K\}\). Let \(S(P_{i}^{k})\) be a function that returns the set of sensor nodes on the path \(P_{i}^{k}\), and \(I(P_{i}^{k})\) returns a set containing all the index of sensor nodes returned by \(S(P_{i}^{k})\). The function \(S(P_{i}^{k}, s_{h})\) returns the set of sensor nodes in \(S(P_{i}^{k})\) that forward data through node \(s_{h}\), where \(s_{h} \in S(P_{i}^{k})\). As an example, see Figure 2, consider the path \(s_{3}-s_{2}-s_{1}-B\). In this case, \(S(P_{3}^{1}) = \{s_{2}, s_{1}\}\); \(I(P_{3}^{1}) = \{2, 1\}\); \(S(P_{3}^{1}, s_{1}) = \{s_{2}\}\).

Let \(E_{i}^{t}\) be the energy consumption rate of sensor node \(s_{i}\). The value of \(E_{i}^{t}\) is initialized to the energy consumption rate incurred when active plus transmission of sensed data; i.e., \(E_{i}^{t} = E_{a} + E_{c}^{t}(Z(s_{i}))\). Here, \(|Z(s_{i})|\) is zero if sensor node \(s_{i}\) does not monitor any targets. Note, a sensor node will spend more of its energy if others are forwarding data through it. We will use \(F_{i}^{t}(P_{k}^{t})\) and \(D_{i}^{t}(P_{k}^{t})\) to denote the increased data forwarding rate and energy consumption rate of node \(s_{i}\) when \(s_{i}\) forwards its data via path \(P_{k}^{t}\), respectively. For example, let sensor node \(s_{2}\) forwards its sensed data using path \(s_{2}-s_{1}-B\); see the solid line in Figure 2. In this case, \(E_{i}^{t}\) is increased by \(D_{1}^{t}(P_{2}^{t})\); i.e., \(E_{i}^{t} = E_{i}^{t} + D_{1}^{t}(P_{2}^{t})\). It then determines the path for \(s_{3}\). If \(s_{3}\) forwards data using path \(P_{3}^{1} = s_{3}-s_{2}-s_{1}-B\), then \(F_{2}(P_{3}^{1}) = F_{1}(P_{3}^{1}) + R\) and \(D_{1}^{t}(P_{3}^{1}) = D_{1}^{t}(P_{3}^{1}) + E_{i}^{t}\). Where \(R\) is the data generated from target \(t_{2}\). On the other hand, if \(s_{3}\) forwards data using path \(P_{3}^{2} = s_{3}-s_{4}-s_{1}-B\), the increased data forwarding rate \(F_{2}(P_{3}^{2})\) is \(R\) whilst \(F_{1}(P_{3}^{2}) = 2R\). In this case, the increased energy consumption rate of \(s_{4}\) and \(s_{1}\) is \(E_{i}^{t} + E_{i}^{t} + 2E_{i}^{t}\), respectively. This is because the increase in energy consumption rate at \(s_{4}\) is the sum of the energy consumption rate due to being active and forwarding of data from \(s_{3}\). Moreover, activating \(s_{4}\) results in additional data flowing through node \(s_{1}\); i.e., the sensed data from \(t_{1}\). Lastly, given the residual energy \(E_{i}^{t}\) of all sensor nodes \(s_{i} \in S\) in time interval \(t\), we define \(L(P_{k}^{t})\) to be the minimum operation time of the sensor nodes on the path \(P_{k}^{t}\) if \(s_{i}\) forwards its data via path \(P_{k}^{t}\); i.e., \(L(P_{k}^{t}) = \min \{E_{i}^{t} F_{i}^{t}(P_{k}^{t}) | \forall h \in I(P_{k}^{t})\}\).

We now describe EC-MLCCEH. Given the bipartite graph \(G(S' \cup \Omega, E)\) and \(K\) shortest paths from each sensor node to the sink, firstly determines the set of sensor nodes \(C_{i}\) to be activated in a time interval \(t\) to ensure complete targets coverage and connectivity, whilst allowing them time to recharge. Specifically, at the beginning of each time interval \(t\), EC-MLCCEH first selects sensor nodes from \(S'\) into a set \(C_{0}\) to cover all the targets. EC-MLCCEH then determines the paths \(P_{i}^{t}\) for all sensor nodes \(s_{i} \in C_{0}\) to transmit sensed data to the sink. Details of how EC-MLCCEH constructs \(C_{0}\) and the paths will be discussed later. After that, all sensor nodes in \(C_{0}\) and \(S(P_{i}^{t})\), where \(s_{i} \in C_{0}\), construct the set cover \(C_{t}\) to be activated in the time interval \(t\). The duration in which sensor nodes in \(C_{t}\) remain active, i.e., the value of \(\delta_{t}\), is defined as the minimum of the following value: (i) the minimal activation time of sensor nodes in \(C_{t}\), or (ii) the minimal recharging time of sensor nodes not in \(C_{t}\). That is, whenever a sensor node’s battery is at capacity, EC-MLCCEH generates a new subset of sensor nodes to be activated. This is
an important consideration because sensor nodes are not able to store harvested energy when their battery is at capacity, and thus, EC-MLCCEH minimizes energy wastage due to lost recharging opportunities.

We now describe how EC-MLCCEH selects a sensor node into the set \( C_g \); see Algorithm 1. Given the residual energy \( E_i^t \) of all sensor nodes \( s_i \in S' \), EC-MLCCEH will do the following at the beginning of each time interval \( t \). Let \( s_{i'} \) be a sensor node in \( S' \) with the minimal \( E_i^t \) value; i.e., \( E_i^t = \text{MIN} \{ E_i^t \ | \ s_i \in S' \} \). EC-MLCCEH will turn off \( s_{i'} \) if all targets covered by \( s_{i'} \) are monitored by other sensor nodes, see Line 5. This is because EC-MLCCEH ensures the sensor node with the lowest residual energy is turned off to conserve its remaining energy and provides it with opportunities to recharge. Critically, it checks that turning off \( s_{i'} \) does not uncovered targets. EC-MLCCEH then removes \( s_{i'} \) from \( S' \). If any target is only covered by \( s_{i'} \), sensor node \( s_{i'} \) is removed from \( S' \) and is added into the set \( C_g \). This is because \( s_{i'} \) needs to be activated to maintain complete targets coverage. EC-MLCCEH then repeatedly tries to turn the off the next sensor node with the minimal \( E_i^t \) until \( S' \) is empty; see Line 2-13. If any target is not monitored, EC-MLCCEH exits and no set cover is available. Note, the sink \( B \) is a sensor node that is active at all times.

**Algorithm 1:** Pseudocode for EC-MLCCEH (construction of \( C_g \))

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( C_g = { B } )</td>
</tr>
<tr>
<td>2</td>
<td>while ( S' \neq \emptyset ) do</td>
</tr>
<tr>
<td>3</td>
<td>( E_i^t = \text{MIN} { E_i^t \</td>
</tr>
<tr>
<td>4</td>
<td>Delete ( s_{i'} ) from ( S' )</td>
</tr>
<tr>
<td>5</td>
<td>if all ( z_j \in Z(s_{i'}) ) are covered then</td>
</tr>
<tr>
<td>6</td>
<td>Continue</td>
</tr>
<tr>
<td>7</td>
<td>else</td>
</tr>
<tr>
<td>8</td>
<td>Add ( s_{i'} ) into ( C_g )</td>
</tr>
<tr>
<td>9</td>
<td>if ( E_i^t = 0 ) then</td>
</tr>
<tr>
<td>10</td>
<td>Exit</td>
</tr>
<tr>
<td>11</td>
<td>end</td>
</tr>
<tr>
<td>12</td>
<td>end</td>
</tr>
<tr>
<td>13</td>
<td>if any ( z_j \in Z ) is not covered then</td>
</tr>
<tr>
<td>14</td>
<td>Exit</td>
</tr>
<tr>
<td>15</td>
<td>end</td>
</tr>
</tbody>
</table>

After EC-MLCCEH successfully generates the set \( C_g \), it then proceeds to construct the set cover \( C_t \); see Algorithm 2. EC-MLCCEH first adds all the sensor nodes in \( C_g \) into the set cover \( C_t \). It then determines the path for the sensor nodes in \( C_g \) to forward their sensed data to the sink. Note that all the sensor nodes on a selected path will be added into the set cover \( C_t \). For each sensor node \( s_i \) in the set \( C_g \), EC-MLCCEH calculates \( L(P_i^k) \) for all \( s_i \)'s shortest paths to the sink \( P_i^k \). To do this, it needs to first calculate the increased data flow rate \( F_h(P_i^k) \) and energy consumption rate \( D_h^t(P_i^k) \) for all \( s_i \in S(P_i^k) \). If a node \( s_h \) on the path \( P_i^k \) is not in the set \( C_t \), i.e., \( \delta(s_h, C_t) = 0 \), \( D_h^t(P_i^k) \) is the sum of the energy consumption rate when \( s_h \) is active plus forwarding of sensed data from other sensor nodes. Moreover, the energy consumption rate of all the sensor nodes on the path from \( s_i \) to the sink will be increased according to the sensed data generated by \( s_h \). Thus, the increased data flow and energy consumption rate of \( s_i \) is calculated as follows,

\[
F_h(P_i^k) = \sum_{s_j \in S(P_i^k), s_h} (1 - \delta(s_j, C_t)) R(Z(s_j)) + R(Z(s_h)) \tag{8}
\]

\[
D_h^t(P_i^k) = (1 - \delta(s_h, C_t)) E_h^t + E_f F_h(P_i^k) \tag{9}
\]

The summation part of Equ. (8) represents the increased data flow rate through \( s_i \) due to sensor nodes that are not in the set \( C_t \) and monitoring targets. Note, the expression \((1 - \delta(s_h, C_t))E_h^t\) is zero if \( s_h \) belongs to the set \( C_t \). Consider Figure 2. If \( s_3 \) forwards data through \( s_2 \), say path \( P_3^4 \), the increased energy consumption rate of \( s_1 \) is \( E_f R \); i.e., \( S(P_3^4), s_1 = \{ s_2 \} \) and \( \delta(s_2, C_t) = \delta(s_1, C_t) = 1 \) in Equ. (8) and (9). However, if \( s_3 \) forwards data through \( s_4 \), then this rate at \( s_1 \) is \( 2E_f R \). This is because \( s_4 \) is not in \( C_t \) and \( \delta(s_4, C_t) = 0 \) in Equ. (8).

**Algorithm 2:** Pseudocode for EC-MLCCEH (construction of \( C_t \))

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( C_g, \Psi, E_i^t )</td>
</tr>
<tr>
<td>2</td>
<td>( C_t ) and its activation duration ( \delta_t )</td>
</tr>
<tr>
<td>3</td>
<td>( C_t = C_g )</td>
</tr>
<tr>
<td>4</td>
<td>for all ( s_i \in S' ) do</td>
</tr>
<tr>
<td>5</td>
<td>( E_i^t = E^a + E^x R(Z(s_i)) )</td>
</tr>
<tr>
<td>6</td>
<td>end</td>
</tr>
<tr>
<td>7</td>
<td>for all ( P_i^k \in P_t ) do</td>
</tr>
<tr>
<td>8</td>
<td>for all ( s_h \in P_i^k ) do</td>
</tr>
<tr>
<td>9</td>
<td>Calculate ( D_h^t(P_i^k) ) as per Equ. (8) and (9)</td>
</tr>
<tr>
<td>10</td>
<td>end</td>
</tr>
<tr>
<td>11</td>
<td>( L(P_i^k) = \text{MIN} { E_i^t \</td>
</tr>
<tr>
<td>12</td>
<td>end</td>
</tr>
<tr>
<td>13</td>
<td>( L(P_i^k) = \text{MAX} { L(P_i^k) \</td>
</tr>
<tr>
<td>14</td>
<td>for all ( h \in I(P_i^k) ) do</td>
</tr>
<tr>
<td>15</td>
<td>( E_h = E_i^t + D_h^t(P_i^k) )</td>
</tr>
<tr>
<td>16</td>
<td>if ( s_h \notin C_t ) then</td>
</tr>
<tr>
<td>17</td>
<td>Add ( s_h ) into ( C_t )</td>
</tr>
<tr>
<td>18</td>
<td>end</td>
</tr>
<tr>
<td>19</td>
<td>end</td>
</tr>
<tr>
<td>20</td>
<td>( LO_t = \text{MIN} { E_i^t \</td>
</tr>
<tr>
<td>21</td>
<td>( LR_t = \text{MIN} { E^{\text{max}} - E_i^t \</td>
</tr>
<tr>
<td>22</td>
<td>Return: ( C_t, \delta_t = \text{MIN} { LO_t, LR_t } )</td>
</tr>
</tbody>
</table>

EC-MLCCEH selects a path \( L(P_i^k) \) that has the maximum...
\[ L(P'_k) \] value to transmit the sensed data from \( s_i \) to the sink; i.e., \( L(P'_k) = \text{MAX} \{ L(P'_k) \mid \forall k = 1, \ldots, K \} \). Thus, the sensor nodes with a low residual energy i.e., the sensor nodes on the path \( P'_k \) with a lower \( L(P'_k) \) value than \( L(P'_k) \), are not responsible for forwarding data and their energy is conserved. If any sensor node \( s_h \) in \( S(P'_k) \) is not in the set \( C_t \), EC-MLCCEH will add \( s_h \) into the set \( C_t \). After that, all sensor nodes in \( S(P'_k) \) increment their energy consumption rate \( E'_i \) by \( D'_h(P'_k) \). This procedure repeats for all sensor nodes in \( C_g \), see Line 5-19. Once we have \( C_t \), EC-MLCCEH then calculates the duration in which sensor nodes in \( C_t \) remain active. Let \( L_1 \) and \( L_2 \) be the minimal operation time of all sensor nodes in \( C_t \) and the minimal recharging time of all sensor nodes not in \( C_t \) respectively. Thus, the operation time \( \delta_t \) of \( C_t \) is equal to \( \text{MIN}(L_1, L_2) \).

We now describe the running time complexity of EC-MLCCEH at each time interval. In Algorithm 1, the number of elements in \( S' \) is reduced by one after each while iteration, see Line 4. Thus the total number of iterations is \( \frac{1+|S|}{|S|} \times |S| \). In the worst case, each iteration is carried out \( O(|Z|) \) times; i.e., search all targets. After that, Line 14-16 requires \( O(|Z|) \) steps for each time interval. Therefore, the running time complexity of Algorithm 1 is \( O\left(\frac{1+|S|}{|S|} |Z| + |Z| \right) \). In the worst case, the number of sensor nodes in \( C_g \) is \( |S| \), each has \( K \) paths to the sink and the number of sensor nodes on each path is \( |S| \). Moreover, the procedure for determining the activation time of a calculated set cover, i.e., Line 20 and 21 in Algorithm 2, requires \( O(|S|) \) steps for each time interval. Thus, the running time complexity of Algorithm 2 is \( O(|S|(K|S| + |S| + 2)) \). Therefore the total time complexity is \( O\left(\frac{1+|S|}{|S|} |Z| + |Z| + |S|(K|S| + |S| + 2)\right) \) or \( O(|S|^2(\frac{1}{2}|Z| + K + 1) + |S|(\frac{3}{2}|Z| + 2) + |Z|) \).

\section{IV. Evaluation}

Our experiments are conducted in Matlab running on a computer with an Intel Core i7 CPU@3.5GHz with 8GB RAM. The sensor node parameters correspond to the WaspMote [11] platform. A sensor node consumes 60 mW when in the active state and 0.2 mW when sleeping. Moreover, all sensor nodes are equipped with an Ensocan ECS310 solar cell [12] to harvest solar energy. We assume a conversion rate of 10\% and a recharging efficiency of 50\%, which is conservative as compared to other technologies [13]. In addition, we use real solar irradiance data retrieved from a sunny day at Southwest Solar Research Park, Phoenix, Arizona, USA [14] on the 16th of April 2013. The average recharging rate is derived from the data collected in 24 hours. Table I shows the parameter values used in our experiments.

In the experiments, we compare EC-MLCCEH with LP-MLCCEH. In the EC-MLCCEH, we calculates three shortest paths from each sensor node to the sink, i.e., \( K = 3 \). We also compare EC-MLCCEH and LP-MLCCEH to a heuristic proposed in [5] called Random Activation for MLCCEH (RA-MLCCEH). For the RA-MLCCEH, all sensor nodes have an identical probability to be activated at the beginning of each time interval \( t \) to construct a subset \( C'_t \). If \( C'_t \) does not guarantee complete targets coverage and network connectivity, then it discards this subset and repeats the process until one is found. RA-MLCCEH will terminate if no subsets are found when a predetermined iteration limit is reached. For a \( C_t \) that guarantees coverage and connectivity, we define its operation time to be \( \text{MIN}(L_1, L_2) \). This is because in energy harvesting WSNs, RA-MLCCEH needs to consider recharging opportunities of sensor nodes.

In the experiments, we investigate the network lifetime and running time by varying the following parameters: sensor node density, target density and sensing range. Here, network lifetime is the time duration from which a network starts operation to when any target is not watched by any sensor nodes or a sensor node is not connected to the sink. We disperse sensor nodes and targets within a \( 100 \times 100 \) \( m^2 \) field. All sensor nodes have the same sensing range and their communication is twice the sensing range. Each experiment is conducted with one change to the network configuration while others are fixed. Details of the configurations will be made specific in Section IV-A. Our results are an average of 100 runs, each with a different topology.

\begin{table}[h]
\centering
\begin{tabular}{|l|c|}
\hline
\textbf{Parameters} & \textbf{Value} \\
\hline
Battery size & 1100 mA \\
Consumption rate & 3.6 Joules/minute \\
Data generation rate & 3.8 Kbytes/minute \\
Average recharge rate & 0.96 Joules/minute \\
Voltage & 4 V \\
Solar panel conversion rate & 10\% \\
Recharging efficiency & 50\% \\
Transmission cost & 0.1 Joules \\
\hline
\end{tabular}
\caption{Simulation Parameters}
\end{table}

\section{A. Network Lifetime}

In the following sections, we study the coverage lifetime when the number of sensor nodes, targets and sensing range changes.

1) Sensor Node Density: We first fix the number of targets to five and vary the number of sensor nodes from two to ten with an interval of one. All sensor nodes have an uniform sensing radius of 30 meters. From Figure 3(a), we see that the network lifetime of LP-MLCCEH and EC-MLCCEH increases rapidly when we add more sensor nodes. This is because each sensor node has more opportunities to enter sleep state to recharge as the number of sensor nodes increases. In the RA-MLCCEH algorithm, randomized scheduling of sensor nodes lead to unnecessary activation, e.g., a target is monitored by multiple sensor nodes, and hence result in energy waste. Figure 3(a) also shows that EC-MLCCEH achieves 80\% performance in terms of network lifetime as compared to LP-MLCCEH. This is because EC-MLCCEH minimizes the number of sensor nodes to be activated in each time interval, see Algorithm 1. This helps prolongs network lifetime.
2) Target Density: We then fix the number of sensor nodes to eight and vary the number of targets from two to ten with an interval of one. The sensing range of all sensor nodes is 30 meters. Figure 3(b) shows the network lifetime of LP-MLCCEH and EC-MLCCEH rapidly reduces from 1800 and 1100 hours to less than 200 hours with increasing number of targets from two to five. The reason is because each sensor node needs to have more activation time to monitor the newly added targets. Thus they have fewer opportunities to enter the sleep state to recharge themselves, which reduces network lifetime.

3) Sensing Range: We now fix both the number of sensor nodes and targets to eight and five respectively. The sensing radius is increased from five to 50 meters with an interval of five. From Figure 3(c), we see that the network lifetime of LP-MLCCEH, EC-MLCCEH and RA-MLCCEH increases with the increasing of sensing range. This is because each sensor node is able to cover more targets, and thus, the total number of sensor nodes to be activated is reduced. As a result, sensor nodes have more opportunities to be in the sleep state.

In comparison to the three experiments, we see that the performance of EC-MLCCEH is close to LP-MLCCEH in terms of network lifetime. On the other hand, the network lifetime of RA-MLCCEH is less than 200 hours in all three experiments, which is significantly lower than that of EC-MLCCEH.

B. Running Time

Figure 4 shows the running time of LP-MLCCEH, EC-MLCCEH and RA-MLCCEH when we vary the number of sensor nodes, targets and sensing range. From Figure 4(a) and 4(b), we see that the running time of LP-MLCCEH increases exponentially with increasing number of sensor nodes and sensing range. This is because the number of set covers, i.e., \( \Omega \), increases with more sensor nodes or a larger sensing range. Thus, the number of decision variables to be decided by the LP solver increases. Moreover, with increasing of network lifetime \( T \), there will be more calls to the LP solver. On the other hand, EC-MLCCEH iteratively selects sensor nodes at each time interval, which incurs significantly less computational cost. RA-MLCCEH has the lowest operation time when increasing the number of sensor nodes and targets, see Figure 4(a) and 4(b). This is because sensor nodes waste energy due to improper activation schedule, which result in low network lifetimes and incurs fewer iterations to generate set covers. However, Figure 4(c) shows the running time of RA-MLCCEH is higher than LP-MLCCEH when sensing range is less than 25 meters. The reason is because RA-MLCCEH needs a longer time to construct the set covers that ensure coverage and connectivity when sensor nodes have a small sensing range.

V. Conclusion

This paper is the first to address the network connectivity problem along with the complete targets coverage problem...
in energy harvesting WSNs. Our solutions ensure the set of sensor nodes that monitor targets are connected to the sink. Critically, our solution ensures sensor nodes are afforded recharging opportunities. We outline two solutions. The first assumes an exhaustive collection of set covers and uses a LP solver. The second is a heuristic that selects sensor nodes into a set cover based on their energy level. Simulation results show that the heuristic achieves 80% of the network lifetime attained by the LP based solution.

REFERENCES