EAPC: Energy-Aware Path Construction for Data Collection Using Mobile Sink in Wireless Sensor Networks

Weimin Wen, Shenghui Zhao, Cuijuan Shang, and Chih-Yung Chang, *Member, IEEE*

***Abstract*— Data collection is one of the paramount concerns in wireless sensor networks. Many data collection algorithms have been proposed for collecting data in particular monitoring regions. However, the efficiency of the paths for such data collection can be improved. This paper proposes an energy-aware path construction (EAPC) algorithm, which selects an appropriate set of data collection points, constructs a data collection path, and collects data from the points burdened with data. EAPC is intended to prolong the network lifetime, it accounts for the path cost from its current data collection point to the next point and the forwarding load of each sensor node. Performance evaluation reveals that the proposed EAPC has more efficient performance than existing data collection mechanisms in terms of network lifetime, energy consumption, fairness index, and efficiency index.**

***Index Terms*— Network lifetime, energy consumption, efficient path, maximal benefit, collection point.**

I. INTRODUCTION

Wireless sensor networks (WSNs) have been widely applied for various applications, including environmental monitoring [1], forest fire detection [2], [3], coverage [4], [5], smart homes [6], and medical systems and health care [7], [8]. A WSN is composed of numerous sensor nodes randomly deployed in the monitoring region. Most studies have assumed that sensor nodes are battery-powered and hence have limited energy. Various studies that have investigated the data collection problem can be mainly classified into two categories, depending on the mobility of the sink node. In the class in which the sensor nodes and sink node are static, several energy-balanced media access control (MAC) and routing protocols have been proposed. A study proposed a distributed technique was proposed [9].

Each sensor had a particular slot for listening and another slot for sending. The slot was assigned in a distributed manner and was dynamically adapted. Yahya and Ben-Othman [10] presented an energy-efficient MAC mechanism with quality of service (hereafter, EQ-MAC). The proposed EQ-MAC consists of Classifier MAC (C-MAC) and Channel Access MAC (CA-MAC). Data messages were assigned scheduled slots with no contention, whereas short periodic control messages were assigned random access slots. These studies have improved network lifetimes, but the sensors that happen to be close to the sink nodes have greater data-forwarding workloads, leading to an energy imbalance problem.

To deal with energy imbalances, numerous studies have adopted mobile sink to collect data from sensor nodes. These studies can be classified into two major categories: no-data-forwarding and partial-data-forwarding. In the class of no-data-forwarding, some studies [11]–[14] have adopted mobile sink to visit all sensor nodes for data collection. Because data forwarding is not conducted in the no-data-forwarding category, the energy consumption levels of all sensor nodes can be balanced. However, the constructed paths are excessively long, leading to various problems of energy exhaustion of the mobile sink or buffer overflow of the static sensors.

The energy of mobile sink might be unable to support the complete traversal of a scheduled path because numerous sensor nodes might be deployed. To deal with this long path problem, some studies [15]–[18] in the partial-data-forwarding category have adopted mobile sink that visit only a subset of the sensor nodes. Those sensor nodes are called data collection points (CPs). All sensor nodes are organized as several subtrees, and each tree is rooted at one CP. In each subtree, each node transmits its reading to the root through the subtree. Under a certain path length constraint, the mobile sink constructs a path which passes through the roots of all subtrees. Along the constructed path, the mobile sink collects data from the root of each subtree. However, the data CPs selected in those studies might not be appropriate. This might reduce the lifetime of WSNs.

Given a mobile sink and a set of sensors, this paper proposes an Energy-Aware Path Construction algorithm, called *EAPC*, which evaluates the benefit of each sensor node and selects

---

Manuscript received September 21, 2017; accepted November 6, 2017. Date of publication November 13, 2017; date of current version December 21, 2017. This work was supported in part by the Anhui Science and Technology Project under Grant 1704e1002217, in part by the Natural Science Foundation of Educational Government of Anhui under Grant KJ2016B18 and Grant KJ2016B20, in part by the National Youth Talent Support Program under Grant gxyq2017086, in part by the Projects Leading Talents in Universities of Anhui under Grant gxfxZD2016250, in part by the Chuzhou University Project under Grant 2015qd03 and Grant 2015GH10, in part by the Chuzhou University of Science and Technology Innovation Team, and in part by the Ministry of Science and Technology of Taiwan under Grant MOST 103-2221-E-032-020-MY3 and Grant MOST 106-2221-E-032-015-MY3. The associate editor coordinating the review of this paper and approving it for publication was Prof. Okyay Kaynak. *(Corresponding author: Chih-Yung Chang).*

W. Wen, S. Zhao, and C. Shang are with School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China (e-mail: wenweim@chzu.edu.cn; zsh@chzu.edu.cn; shangcuijuan@chzu.edu.cn).

C.-Y. Chang is with Department of Computer Science and Information Engineering, Tamkang University, Taipei 25137, Taiwan (e-mail: cychang@mail.tku.edu.tw).

Digital Object Identifier 10.1109/JSEN.2017.2773119

a subset of sensor nodes with maximal benefits to play the role of data collection points. The following itemizes the contributions of this paper.

- *Constructing an efficient path.* We propose an algorithm that finds the better set of CPs through benefit calculations and constructs an efficient path.
- *Prolonging network lifetime.* The proposed EAPC considers the length cost between any two successive CPs. Hence, the constructed path allows the mobile sink to visit the greatest possible number of CPs. This facilitates distributing the workload of data forwarding to the greatest possible number of CPs, prolonging the network lifetime.
- *Collecting data while moving.* When the mobile sink moves along the constructed path, it might pass through some sensor nodes that are not data CPs. The data of these sensors also can be collected by the mobile sink when the mobile sink moves and enters the communication ranges of these sensors. Thus, the energy consumed by forwarding the data of these sensor nodes can be conserved.

The remaining parts of this paper are organized as follows. Section 2 reviews the related works. Section 3 presents the network environment and problem formulation of this paper. Section 4 details the operations of the proposed EAPC. The performance improvement of the proposed mechanism is investigated in Section 5. Finally, the conclusions are drawn in Section 6.

II. RELATED WORKS

Existing studies that use mobile sink for data collection are mainly classified into two categories: no-data-forwarding and partial-data-forwarding. The following data briefly reviews some relevant articles.

A. No-Data-Forwarding

In this category, most studies have adopted mobile sink to visit all sensor nodes for data collection. In a previous study [11], mobile sinks were mounted on animals or vehicles moving randomly to collect information from static sensors. This method only considered substantial power saving for every sensor node; however, it was time consuming for the mobile sink to visit each sensor; thus, the collected data were not fresh.

To reduce the energy consumption of the mobile sink, a study [12] proposed a travel planning algorithm for data collection. The algorithm shortened the tour path, the sink visited all sensor nodes and all energy consumption levels were balanced. However, the method may cause long delays in a large-scale sensor network.

In another study [13], a mobile sink was delivered to collect data from all sensor nodes. The main objective of this algorithm was to minimize data delivery latency by constructing a shortest path. Somasundara *et al.* [14] aimed to discover a path such that the data mule could collect all the data before the buffer of any sensor overflowed. They proved that the problem is NP-hard by a reduction from the Hamiltonian cycle problem and formulated it as an integer linear programming problem. However, for a high number of nodes, the approach was impractical because of its high computational complexity.

B. Partial-Data-Forwarding

The basic concept of this category is that visiting each sensor is impractical when numerous sensor nodes are deployed. To overcome this problem, some studies [15]–[18] in the partial-data-forwarding category have proposed that the mobile sink only visit a subset of the sensor nodes, and that all the other sensors transmit their own data to the visited sensors.

Zhao and Yang [15] presented two schemes for mobile sink collecting data, called the shortest-path-tree-based data-gathering algorithm (SPT-DGA) and the priority-based CP-selection algorithm (PB-PSA). The SPT-DGA is a centralized algorithm that iteratively discovers an optimal set of CPs among sensor nodes on a shortest-path tree. The selected CPs must connect as many sensor nodes as possible under a limited relay-hop bound. The PB-PSA is a distributed algorithm, in which each sensor node must count its neighbors and calculate the distance to the data sink, and then iteratively discover the optimal set of CPs according to those two parameters. However, the SPT-DGA does not consider the distance between two successive CPs when selecting the CPs. Hence, the efficiency of the constructed path still can be improved.

Xing *et al.* [16] proposed a rendezvous approach with a limited path length. The proposed mechanism first constructs a routing tree that connects all sensor nodes. Subsequently, each edge of the routing tree is assigned a weight that corresponds to the number of nodes that use that edge to forward the data to the sink node. However, the proposed mechanism is restricted to move only on the edges of the tree, suggesting that the mobile sink will visit the sensor nodes on the selected edges twice.

Almi'ani *et al.* [17] proposed a cluster-based algorithm that partitions the network into numerous clusters. In each cluster, a CP is selected. These selected CPs compose the shortest path with a length constraint. The mobile sink must visit the set of CPs along the path to collect data. However, the length of the constructed path still can be further reduced. Moreover, the CPs exhibit nonuniform energy depletion, which reduces the lifetime of the WSN.

A study [18] proposed weighted rendezvous planning (WRP). Each sensor is assigned a weight, and the sensor nodes with the highest weight are considered as CPs. Subsequently, a traveling tour that passes through all CPs is constructed for the mobile sink. In path construction, the WRP treats the sink as the first CP. Then the weight of each sensor node is calculated by multiplying the number of packets that it forwards by its hop distance to the closest CP on the tour. The highest weighted sensor node is treated as a CP, and a tour is determined for the mobile sink from the set of CPs. The tour length is limited, thus, the visited path length must be equal to or less than the maximum tour length. However, the path length from the current location to the next CP is not considered. This might lead to the problems of less data collected or shorter network lifetime.

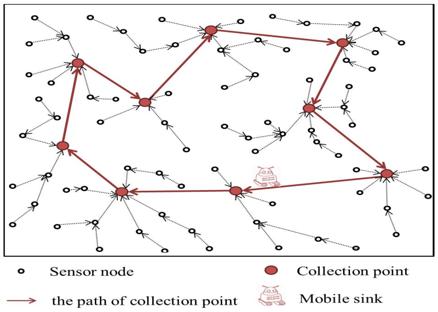


Fig. 1. Network environment. The mobile sink should construct the path to visit all CPs under predefined long constraint.

All aforementioned algorithms emphasize the improvement of energy imbalance or aim to cope with the data freshness problem. But most of them do not consider the path cost between each two CPs. Thus, the constructed paths can be shortened. This paper proposes EAPC, which selects an appropriate set of CPs, constructs a data collection path, and collects the sensor data from highly burdened CPs, thus helping CPs to maximize their network lifetimes.

III. NETWORK ENVIRONMENT AND PROBLEM FORMULATION

This section presents the network environment and assumptions of the modeled WSN. The problem formulation is subsequently presented.

A. Network Environment

This paper considers a certain mobile WSN, which comprises a set of n static sensors $S = \{s_1, s_2, s_3, \cdots s_n\}$ distributed over a given rectangular region R. A mobile sink moves in region R to collect data from all static sensors. The mobile sink is assumed to be equipped with energy harvesting modules, such as solar power systems. Assume the mobile sink knows its own location and the locations of all sensor nodes. A round is the process in which the mobile sink leaves its home location, moves, collects all data generated by all sensors, and then goes back to its home location. Each sensor node periodically generates one data packet in each round, and each packet must be delivered to the mobile sink. The mobile sink and all static sensor nodes are assumed to have an identical communication range. The simplest strategy for the mobile sink to collect data from all sensor nodes is to construct a path that passes through each sensor node. However, that would take so much time that some sensors' buffers might overflow before the mobile sink could visit them. To prevent this situation, the mobile sink should execute the path planning task, which comprises two major subtasks. The first subtask is to select a set of m CPs $P = \{p_1, p_2, p_3, \cdots p_m\}$ from the n static sensor nodes. The second subtask is to construct a path L_P that passes through all CPs. Along the constructed path L_P, the mobile sink can visit each CP and can receive data from each CP.

Fig. 1 presents an example of several sensor nodes and a mobile sink in a rectangular region. In Fig. 1, the empty circles represent the static sensor nodes whereas the red circles denote the selected CPs. The sensor nodes have been partitioned into nine groups, and each group organizes a tree topology rooted by a CP. In each tree, all static sensors send their data along the edges of the tree, and the CP stores the data until the mobile sink visits it. Furthermore, the mobile sink moves along the established path, which passes through all CPs to collect data such that the network lifetime of the WSN can be prolonged as much as possible. Table I depicts the notations that will be used throughout this paper.

TABLE I

NOTATION USED IN THIS PAPER

Notation	Description
E_R	Energy consumption for receiving one packet
E_T	Energy consumption when transmitting one packet
|L|	The length of path L
L_{max}	The valid path with maximal length
x	The data generated rate of each sensor node
b	The buffer size of each collection point
C	The battery capacity
$H(i, r_T)$	The hop distance from sensor s_i to the the root r_T of the tree T that contains s_i
$C(i)$	The number of sensor nodes in the subtree rooted by sensor node s_i
$\omega(i)$	The number of packets saved for transmission by selecting sensor node s_i as the CP
$ND(i)$	The number of packets collected by sensor node s_i in each round
A_i	The set of sensor node lying on the path from sensor node s_i to the root of the tree that contains sensor node s_i
B_i	The set of sensor nodes that are passed by the new path that contains s_i but do not belong the tree rooted by s_i
b_i	The benefit index obtained by selecting sensor node s_i as the data collection point

B. Problem Formulation

Energy consumption is the paramount factor determining the lifetime of a WSN. The lifetime of a WSN is measured by the time period starting from the time point when the network begins operation to the time point when the first sensor node runs out of energy, because a dead sensor is expected to be unable to relay or send data to the sink node. This paper presents a method to plan a tour for collecting data from highly burdened CPs. This helps the CPs minimize energy expenditure and hence maximize the network lifetime. The proposed strategy selects a set of CPs and plans the path for data collection to minimize the power consumption of the static sensor nodes.

The following presents the energy consumption model of this paper. The sensor nodes mainly consume energy by operations, such as data sending and receiving. Assume that the size of each packet is k bits. Let sender s_i and receiver s_j be a communication pair and consider that s_i sends k bits to s_j. Let d_{ij} be the distance between s_i and s_j. The energy consumption of each receiver node for receiving one packet (k bits) can be measured by Equ. (1):

$$E_R = k\beta \tag{1}$$

where β is the energy consumption per bit incurred by the receiving circuit. The energy consumption for sender s_i to transmit one packet (k bits) to its parent s_j is expressed as:

$$E_T = k\varepsilon_1 + kd_{i,j}^{\gamma}\varepsilon_2 \quad (2)$$

where ε_1 is a factor that represents the energy consumption per bit of the transmitting circuit, ε_2 is the energy consumption per bit of the amplifier circuit, and γ is the path-loss exponent (which typically ranges between 2 and 4, depending on the environment).

To reduce the length of the data collection path, some CPs are selected from the set of all static sensor nodes. Consider a set of CPs $\{P = p_1, p_2, \cdots p_m\}$. These m CPs are selected from the n sensors, and all sensors are partitioned into m disjoint trees $T_1, T_2, T_3, \cdots T_m$. Let tree T_i is rooted by CP p_i and T_i has totally $u_i + 1$ nodes. The sensor nodes will upload their data to the root along the tree topology. The mobile sink will construct a path that passes through all CPs for data collection from CPs. The static sensor nodes consume energy when they perform the operations including data sending and data receiving. Let E_i denote the energy consumption of CP p_i in each round. Equation (3) evaluates the value of E_i, which is the total energy required to receive u_i packets and transmit $u_i + 1$ packets in tree T_i.

$$E_i = u_i \times E_R + (u_i + 1) \times E_T \quad (3)$$

The network lifetime highly depends on the sensor node with the lowest quantity of remaining energy. Because each CP consumes more energy than any sensor in its tree, this study aims to minimize the energy consumption of the bottlenecked CP that consumes the most energy of all the CPs. Equation (4) reflects this goal.

Objective:

$$\underset{1\le i\le m}{Min\ Max}\ E_i \quad (4)$$

For objective (4) to be achieved, constraints (5)–(8) must be satisfied. Let $|L|$ denote the length of path L. Let L_{min} denote the shortest Hamiltonian path passing through every CP in set P and let L_{max} denote the valid path with maximal length. Let L_P denote the constructed path passing through all CPs and let $d_{i,i+1}$ represent the length of edge (p_i, p_{i+1}) in L_P. Expression(5) illustrates the constraint of the constructed path L_P.

1) Distance Constraint:

$$|L_{min}| < |L_P| = \sum_{i=1}^{m} d_{i,(i+1) mod\, m} \le |L_{max}| \quad (5)$$

The m CPs should receive the data packets periodically generated by the sensor nodes in its tree. Let x be the data generating rate of each sensor node. Let b be the buffer size of each CP. Let t_P denote the length of a cycle. The amount of data generated by each sensor node in time period t_p is $x \cdot t_p$. Because CP p_i is the root of tree T_i and has $u_i + 1$ nodes, the total amount of data stored in CP p_i in each round would be $(u_i + 1) \cdot x \cdot t_p$. To prevent the buffer overflow, t_p should satisfy the following constraint.

2) Buffer Constraint:

$$(u_{i+1}) \cdot x \cdot t_p \le b \quad \text{for all } p_i, \quad 1 \le i \le m \quad (6)$$

This also bounds the time for the mobile sink to collect data in each round. Let ζ denote the lifetime of a WSN. The number of rounds that the mobile sink can collect data from each CP is ζ/t_p, because the energy consumption of each p_i is $x \cdot t_p \cdot E_i$ in each round t_p. The total energy consumption of *CP* p_i is $x \cdot t_p \cdot E_i * \zeta/t_p$. Constraint (7) ensures that the total energy consumption must be less than or equal than the battery capacity.

3) Battery Constraint:

$$x \cdot t_p \cdot E_i \cdot \zeta/t_P \le C \quad 1 \le i \le m \quad (7)$$

where C denotes the battery capacity of each sensor.

Let S_j denote the set of nodes rooted by CP p_j. Let f_j^{out} denote the total amount of packets transmitted by CP p_j in each round and let f_i^{in} denote the number of packets generated from each sensor s_i in set S_j to CP p_j in each round. To ensure that all packets of S_j can be transmitted, the system should satisfy *Flow Constraint* (8) as follows:

4) Flow Constraint:

$$f_j^{out} = \left(\sum_{\forall s_i \in S_j} f_i^{in}\right) + 1 \quad (8)$$

IV. THE PROPOSED EAPC ALGORITHM

This section presents the proposed EAPC algorithm, which aims to identify the set of CPs and construct a path passing through all CPs such that the lifetime of the sensor network can be maximized, under constraints (5)–(8). Moving along the constructed path, the mobile sink can collect data from highly loaded CPs. The EAPC algorithm is composed of three phases: initialization, CP selection, and path construction phases. In the initialization phase, the EAPC constructs a minimum spanning tree, which is rooted at the base station. Then in the CP selection phase, the EAPC selects m data CPs from n sensor nodes and partitions the whole tree into m distinct subtrees. Finally, in the path construction phase, the EAPC constructs the shortest tour that passes through all CPs. As a result, the mobile sink can visit each CP along the constructed path and can collect all the data. The following presents the details of the three phases and the complexity analysis of EAPC algorithm.

A. Initialization Phase (I Phase)

The initialization phase aims to construct a minimum spanning tree rooted at the base station. Given a graph $G(V, \Psi)$ where V denotes the set of n sensor nodes and $\Psi = \{e_1, e_2, \cdots e_{n-1}\}$ denotes the set of edges between sensor nodes in V. In the *Initialization Phase*, the Prim's algorithm is applied to construct a minimum spanning tree (*MST*). Fig. 2 provides an example to explain the operations of this phase. Fig. 2(a) illustrates 20 sensor nodes, including one base station marked with red. After applying the ***I Phase***, a minimal spanning tree (which guarantees that the total length of the tree is minimal) has been constructed, as shown in Fig. 2(b).

After completing the executions of this phase, the mobile sink performs the Collection Point Selection Phase.

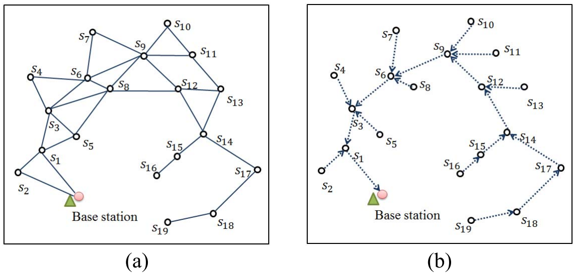


Fig. 2. Example of initialization phase. (a) Graph $G(V, \psi)$. (b) The *MST*.

B. Collection Point Selection Phase (CPS Phase)

In the constructed *MST*, the base station plays the role of the tree root, and each nonroot sensor is connected to a parent. It is assumed that the mobile sink is initially located at the location of the base station. Let P denote the set of data CPs, which has been selected by the base station. Let S denote all sensor nodes and S' denote the set of sensor nodes not selected by the base station. That is, $S' = S - P$. The base station is responsible for selecting one best sensor node s^{best} at a time from set S'. Then the selected s^{best} is included set P and is removed from S'. After that, the original tree is restructured. One major change of the tree is that the data collected by s^{best} is sent from s^{best} to the mobile sink, instead of the parent of s^{best}. That is, the link from s^{best} to its parent is broken. Another change occurs in those sensor nodes whose communication ranges are passed through by the path of the mobile sink. Data collected by each of those sensor nodes can be directly sent to the mobile sink, instead of the base station. In this manner, the energy consumption of some sensor nodes responsible for forwarding these data can be minimized.

The following presents the general algorithm for the base station to select the CPs. Assume that $P = \{p_1, p_2, \cdots p_g\}$. That is, the base station has selected g sensor nodes to play the role of *CP*s. The following illustrates how the base station selects the $(g+1)$-*th* sensor node from set S'. Let Boolean variable λ_i denote whether sensor node s_i is rooted by the base station. Define λ_i as follows:

$$\lambda_i = \begin{cases} 0, & sensor\ s_i\ is\ rooted\ by\ CP \\ 1, & sensor\ s_i\ is\ rooted\ by\ the\ base\ station \end{cases} \tag{9}$$

Let $H(i, r_T)$ denote the hop distance from sensor s_i to the root r_T of the tree T that containss_i. Let $C(i)$ denote the number of sensor nodes in the subtree rooted by sensor node s_i. Assume that each sensor generates one packet in each round. Let $ND(i)$ denote the number of packets collected by sensor node s_i in each round, including the packet generated by sensor node s_i. We have

$$ND(i) = C(i) + 1 \tag{10}$$

Let A_i denote the set of sensor nodes lying on the path from sensor node s_i to the root of the tree that contains sensor node s_i. In case that $\lambda_i = 1$ and sensor node s_i is selected as a CP, all sensor nodes $s_j \in A_i$ can save their energy because they are not required to forward data which are sent from s_i to the base station. Thus, the number of packets saved by selecting s_i as the *CP* is

$$\lambda_i \times H(i, base\ station) \times ND(i) \tag{11}$$

On the contrary, if $\lambda_i = 0$ and sensor node s_i is selected as a CP, all sensor nodes $s_k \in A_i$ can save their energy because they are not required to forward the data to the CP, say p_j. As a result, the number of packets saved by selecting s_i as the CP is

$$(1 - \lambda_i) \times H(i, p_j) \times ND(i) \tag{12}$$

Let $\omega(i)$ denote the number of packets saved for transmission by selecting sensor node s_i as the CP. From (11) and (12), the value of $\omega(i)$ can be measured as follows:

$$\omega(i) = \lambda_i \times H(i, base\ station) \times ND(i) + (1 - \lambda_i) \times H(i, p_j) \times ND(i) \tag{13}$$

Let B_i denote the set of sensor nodes that are passed by the new path that contains s_i but do not belong the tree rooted by s_i. Nodes in set B_i can also save their energy for forwarding the data to their roots. Therefore, the obtained benefits, by selecting s_i as a CP, can be measured by

$$\sum_{s_j \in B_i} (\omega(j))$$

Let $\text{dist}(p_j^{closest}, s_i)$ denote the minimal distance from sensor node s_i to $s_j \in P$. Let $\text{dist}(s_i, s_j)$ denote the distance between sensor nodes s_i and s_j. Let b_i denote the benefit index obtained by selecting sensor node s_i as the data CP. The value of benefit index b_i can be measured by the number of packets saved for transmission, divided by the cost of tour distance from some $s_j \in P$ to s_i. That is,

$$b_i = \frac{\omega(i) + \sum_{s_j \in B_i} (\omega(j))}{dist(p_j^{closest}, s_i)} \tag{14}$$

The base station calculates the b_i of each $s_i \in S'$ and selects the best node s^{best} to play the role of CP. That is, $s^{best} = arg \max_{s_i \epsilon s'} b_i$. After selecting the best sensor node as the CP, the base station will calculate whether the tour length of the mobile sink exceeds the maximal length L_{max}. Let L_P denote the length of the shortest tour that passes through all *CP*s in P including new CP s_i. Sensor node s^{best} will be included in P if the following condition is satisfied.

$$L_{max} \geq L_P \tag{15}$$

If it is the case, sensor s^{best} is added in P and is removed from S'.

$$P = P \cup \left\{s^{best}\right\} \tag{16}$$

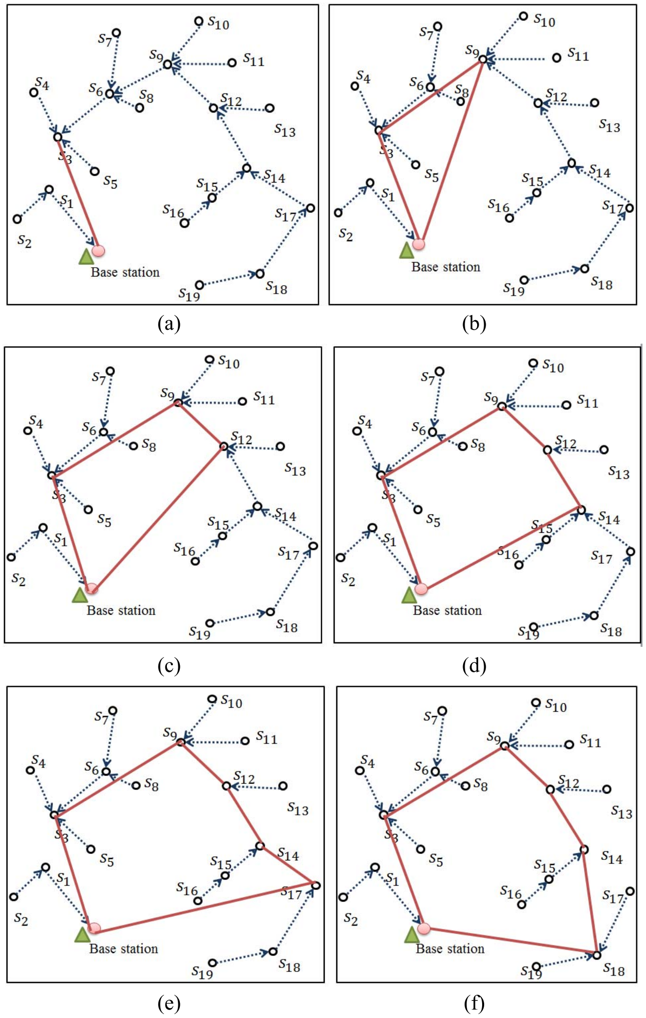
$$S' = S' - \left\{s^{best}\right\} \tag{17}$$

Otherwise, s^{best} is not included in P, and the execution of this phase is terminated. The aforementioned operations must be repeated until the tour length of the mobile sink is larger than L_{max}. The proposed CPS is summarized in Algorithm 1.

Algorithm 1: Collection Point Selection (CPS) Algorithm

```
Input: $L_{\max}$, $S = \{s_1, s_2, s_3, \ldots, s_n\}$, $S' = S - P$, $P = \emptyset$, $L_P = 0$;
Output: A set of ordered collection points $P = \{p_1, p_2, \ldots, p_m\}$;

P = {$p_0$ = base station};
While ($L_{max} \geq L_P$){
    Evaluate $ND(i)$ according to Equ. (10), for each $s_i \in S$;
    Evaluate $\omega(i)$ according to Equ. (13), for each $s_i \in S$;
    Evaluate $b_i$ according to Equ. (14);
    $s^{best} = \arg\max_{s_i \in s'} b_i$;
    If ($L_{max} \geq L_P$) {
        $P = P \cup \{s^{best}\}$;
        $S' = S' - \{s^{best}\}$;
        Reconstruct MST; } /* end if */
    else
      Exit};
      Return P;
```



(a) (b)

(c) (d)

(e) (f)

Fig. 3. Example of EAPC operating in a WSN. (a) Selecting sensor node s_3 as CP. (b) Selecting sensor node s_9 as CP. (c) Selecting sensor node s_{12} as CP. (d) Selecting sensor node s_{14} as CP. (e) Selecting sensor node s_{17} as CP. (f) Selecting sensor node s_{18} as CP.

The following gives an example of executing the proposed CPS algorithm.

To continue the example given in Fig. 2(b), assume the maximum length of traversal is $L_{max} = 150\ m$. As shown in Fig. 3(a), the proposed CPS phase calculates the benefit index value b_i of each sensor node in tree by applying Equ. 14. Then the *CPS-Phase* selects sensor node s_3, which has the maximal benefit as the candidate of CP. Because the length L_P of the tour set $P = \{base\ station, s_3\}$ is smaller than $L_{max} = 150\ m$, the CPS phase adds s_3 into the tour set P and removes s_3 and edge (s_3, s_1) from sets V and Ψ, respectively. The *MST* trees will be reconstructed by considering base station and s_3 as tree roots. The repetitive process of executing the CPS phase selects one sensor node with maximal benefit to play the role of CP at each iteration until the tour length is larger than the length bound L_{max}. In this example, the proposed CPS phase selects s_9, s_{12}, s_{14}, and s_{18} to serve as CPs. Figs. 3(b), 3(c), 3(d), and 3(e) present the results of the tree after selecting s_9, s_{12}, s_{14}, and s_{18}, respectively. Thus, the final tour path of $P = \{base\ station, s_3, s_9, s_{12}, s_{14}, s_{18}\}$.

C. Path Construction Phase

Let $P = \{p_i | 1 \leq i \leq m\}$ denote the set of m CPs. This phase aims to construct a path that passes through all p_i in P. Let $\hat{p}_i$ represent the i-th visited *CP* in the path $L_P = (\hat{p}_0, \hat{p}_1, \hat{p}_2, \ldots \hat{p}_m, \hat{p}_0)$, where $\hat{p}_0$ denotes the base station. The following presents how to construct the path L_P. Initially, the base station in P is selected as the first CP for constructing the path L_P.

The path construction phase includes three steps: the construction of the convex polygon, connection of the remaining CPs, and renumbering.

Step 1 (Construction of the Convex Polygon): Initially, let $\hat{p}_{turn} = p_0(base\ station)$ be the first turning point. Let l be a horizontal line passing through $\hat{p}_{turn}$ and let l have an infinite length. Then, we turn l in a counterclockwise direction until it touches any *CP*, which will be labeled with $\hat{p}_1$. Then the touched *CP* serves as $\hat{p}_{turn}$ and the aforementioned operations are repeatedly executed. In general, in the i-th repetition, assume that $\hat{p}_{turn} = \hat{p}_{i-1}$ which has been passed through by line l. We turn l in a counterclockwise direction. The CP that is firstly touched by l will be labeled with $\hat{p}_i$. This process will be repeatedly executed until the touched CP is base station. Finally, we can construct a path $P^{init} = \{\hat{p}_0, \hat{p}_1, \hat{p}_2, \ldots \hat{p}_k, \hat{p}_0\}$

Step 2 (Connection of the Remaining CPs): This step will be performed only if $k < m$. This indicates that there are $(m-k)$ *CP*s excluded in L_P. Because path L_P should pass through all CPs $p_i \in P$, this step joins the $(m-k)$ *CP*s in the path. Let $P^{internal}$ be the set of remaining $(m-k)$ *CP*s. Let $d(p_i, p_j)$ denote the distance between p_i and p_j. The first CP $p_{cl} \in P^{internal}$ that joins the path is the one which raises the least increment of path length as shown in Equ. (18).

$$p_{cl} = \arg \min_{p_j \epsilon P^{internal}, \hat{p}_i \epsilon P^{init}} \left[d\left(p_j, \hat{p}_i\right) + d\left(p_j, \hat{p}_{i+1}\right) - d\left(\hat{p}_i, \hat{p}_{i+1}\right) \right] \quad (18)$$

Then the point p_{cl} participates in polygon G to form a new polygon by connecting p_j to two CPs $\hat{p}_i$ and $\hat{p}_{i+1}$ and removing the edge $(\hat{p}_i, \hat{p}_{i+1})$. After that, the sets $P^{internal}$ and P^{init} will be updated by applying the following operations.

$$P^{init} = P^{init} \cup \{p_{cl}\} \tag{19}$$

$$P^{internal} = P^{internal} / p_{cl} \tag{20}$$

The aforementioned operations are applied repeatedly until the condition $P^{internal} = \phi$ is satisfied, which indicates that all *m-k* CPs have been include in P^{init}.

Step 3 (Renumbering): In this step, the path will be renumbered such that the base station is the first point in the constructing path. Therefore, the renumbered path is

$$L_P = \{\hat{p}_0, \hat{p}_1, \hat{p}_2, \ldots \hat{p}_m\} \tag{21}$$

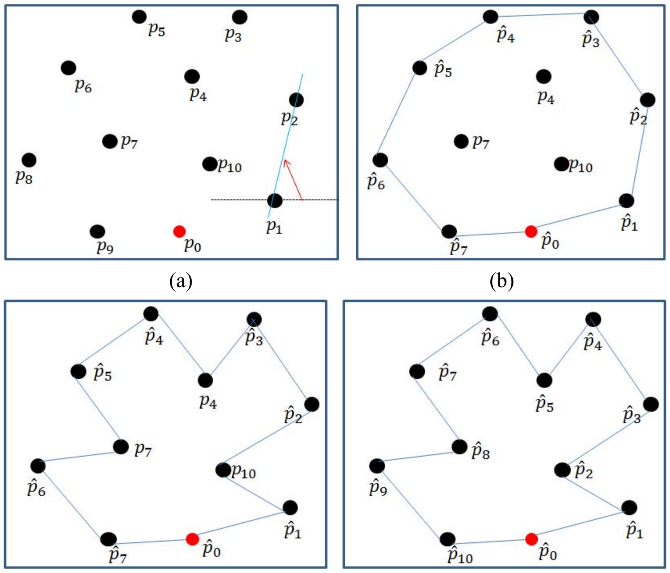
After constructing the path L_P, the third phase is finished. A formal example of the path construction algorithm is as follows:

Algorithm 2: Path Construction Algorithm

Input: A set of collection points $P = \{p_1, p_2, p_3, \cdots p_m\}$.
Output: The path L_P

```
1.  $P^{init} = \{\hat{p}_0 = p_0\}$;
2.  int $i = 1$; int $j$;
3.  $\hat{p}_{turn} = p_0$;
4.  Let $l$ be a horizontal line passing through $\hat{p}_{turn}$;
5.    do {
6.        Turn $l$ in a counterclockwise direction until it
          touches any CP $p_j$;
7.        labeled $p_j$ with $\hat{p}_i$;
8.        $P^{init} = P^{init} \cup \{\hat{p}_i\}$;
9.        $\hat{p}_{turn} = \hat{p}_i$;
10.       $i++$;
11.     } While ($\hat{p}_{turn} = p_0$)
12. $P^{internal} = P \setminus P^{init}$;
13. While ($P^{internal} \neq \phi$)
14.     { compute $p_{cl}$ according to Equ. (18);
15.       remove edge $(\hat{p}_i, \hat{p}_{i+1})$;
16.       connect point $p_{cl}$ to points $p_i$ and $p_{i+1}$;
17.       $P^{init} = P^{init} \cup \{p_{cl}\}$;
18.       $P^{internal} = P^{internal} \setminus p_{cl}$;
19.     }
20. $L_P \leftarrow renumbered\{\hat{p}_0, \hat{p}_1, \hat{p}_2, \ldots \hat{p}_m\}$;
21. Return $L_P$;
```

Fig. 4 depicts 11 CPs. Assume that the set of 11 CPs is $P = \{p_0, p_1, p_2, \ldots p_{10}\}$ where the base station is denoted as p_0. The following depicts how a convex polygon can be constructed. Let l be an infinitely long horizontal line passing through $\hat{p}_{turn} = p_0$. Then, turn l in a counterclockwise direction until it touches the first CP p_1, which is labeled with $\hat{p}_1$. Similarly, when the CP $\hat{p}_1$ is on the line l, it plays the role of $\hat{p}_{turn}$, and the CP $\hat{p}_2$ can be touched by l and will be included in P^{init}. Repeat the aforementioned operations, until $\hat{p}_0$ is finally touched by l . Then a path will be constructed as shown in Fig. 4(a). After that, points in set $P^{internal} = \{p_4, p_7, p_{10}\}$ should be further included in P^{init} by applying step 2.



(a) (b)
(c) (d)

Fig. 4. Example of the path construction phase. (a) Construction of convex polygon. (b) Connection of remaining CPs. (c) The remaining CPs connection. (d) Renumbering all the CPs.

Fig. 4(b) illustrates that remain three CPs. If all CPs have been included into P^{init}, the second step can be ignored. However, in the example shown in Fig. 4(b), the second step should be applied. According to Exps. (18)-(20), CPsp_4, p_7, p_{10} can be added to P^{init} as shown in Fig.4(c). Fig.4(d) shows the renumbering operation that all the CPs in P^{init} are renumbered, starting from the base station. Thus, the renumbered path $L_P = (\hat{p}_0, \hat{p}_1, \hat{p}_2, \ldots \hat{p}_{10})$ can be obtained.

D. Complexity Analysis

The proposed EAPC mainly consists of initialization phase, CP selection phase and path construction phase. In the initialization phase, the complexity is simply O(n).

In the CP selection phase, CPS algorithm is proposed. The following discusses the complexity of each statement in the algorithm. The complexity of step 1 is O(1). Similarly, the complexities of steps 3-6 and 10-11 are O(n) and the complexities of steps 7-9 are O(1). As a result, the complexity of the CP selection phase is O(n).

In the path construction phase, a path construction algorithm is proposed. The following discusses the complexity of each statement in the algorithm. The complexity of steps 1-4 is O(1). Similarly, the complexities of steps 5-10 and 20 are O(n) and the complexities of steps 11-19 are O(n^2). As a result, the complexity of the path construction phase is O(n^2).

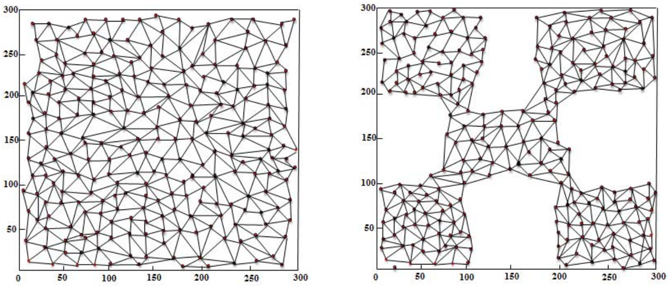
The complexity of EAPC algorithm is O(n) + O(n) + O(n^2) = O(n^2). The time complexities of WRP and CB are O(n^3). The time complexity of RP-CP is O($n^2 \log n$). In comparison, the proposed EAPC outperforms WRP, CB, and RP-CP in terms of time complexity.

V. PERFORMANCE EVALUATIONS

This section compares the performances of our proposed algorithm EAPC, and the existing algorithms RP-CP, CB, and

TABLE II
SIMULATION SETTING

Parameter	**Value**
Simulator	Matlab
Node deployment	Random
Given Region	300*300
The number of sensor node	10-300
Mobile sink speed	2 m/s
Sensor node transmission range	20m-40m
Consumed energy in transmitter circuit	0.18J
Consumed energy at the receiver circuit	0.1J



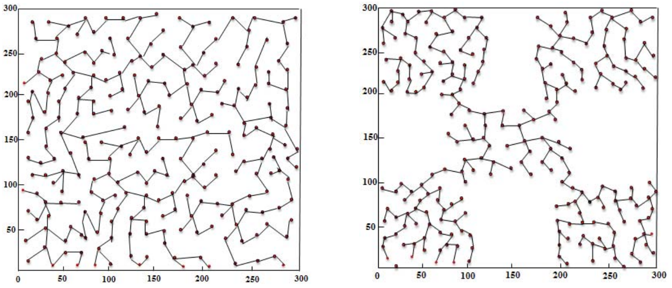
(a) (b)

Fig. 5. Two scenarios considered in the experiments. (a) BD Scenario. (b) UD Scenario.

WRP [16]–[18], in terms of the network lifetime, energy consumption, fairness index, and efficiency index. The MATLAB simulator is used as the simulation tool. Two scenarios are considered in the experiments.

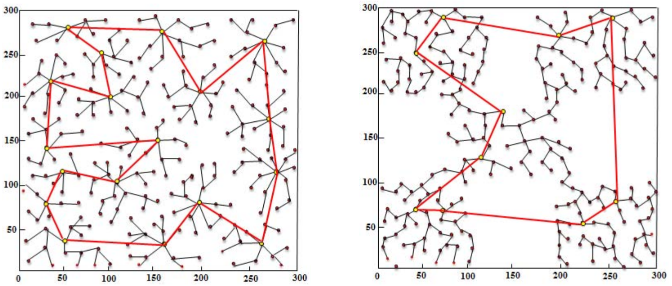
The following illustrates the parameters considered in the simulation environment. The sensor nodes are deployed randomly in the region. The initial energy of each sensor nodes is 100 J. The sensing range of each sensor node is set as 20 m, whereas the communication range varies from 20 to 40 m. The parameters are summarized in Table II. Each sensor node generates one data packet in each round, during each round, the sink starts from the base station, collects data at each CP and then goes back to the base station. Assume that every node is aware of the behavior of the mobile sink, including the movement trajectory and arrival time of the mobile sink.

To further investigate the performance of the proposed mechanisms, two scenarios are considered in the experiments (Fig. 5). In the first scenario, called the balanced deployment scenario (BD Scenario), sensors are randomly deployed over a region that measures 300 m × 300 m, all sensors are connected. Fig. 5(a) depicts a deployment snapshot of 200 sensor nodes in the BD Scenario. Any line connecting two sensor nodes represents that those two sensor nodes can communicate with each other. In the second scenario, called the unbalanced deployment scenario (UD Scenario), the given region is initially partitioned into nine subregions. Then, the sensor nodes are divided into five groups. The sensor nodes in the five groups are randomly deployed in subregions numbered with 1, 3, 5, 7, and 9, but all sensors are connected. Fig. 5(b) is an example of UD Scenario with 200 sensor nodes.



(a) (b)

Fig. 6. Construct a minimum spanning tree rooted at the sink. (a) BD Scenario. (b) UD Scenario.



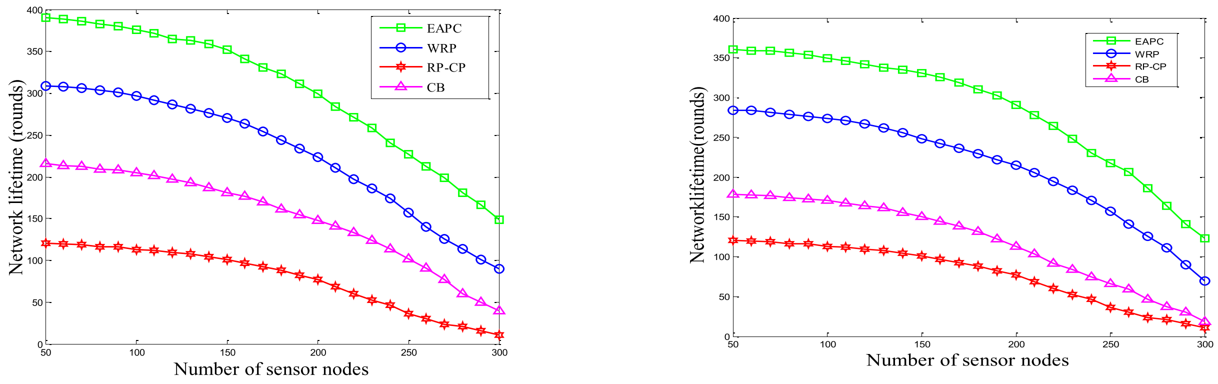
(a) (b)

Fig. 7. Results of EAPC operation in two experimental scenarios. (a) BD Scenario. (b) UD Scenario.

Recall that the initialization phase of the proposed EAPC algorithm aims at constructing an *MST*, denoted by T. Fig. 6 depicts the tree rooted at the base station, T can be found by applying Prim's algorithm. The *MST* $T(V, \Psi)$ connects all sensor nodes by using the shortest path and does not contain any cycles. Fig. 6(a) shows a snapshot of *MST* $T(V, \Psi)$ containing 200 sensor nodes in BD Scenario. Fig. 6(b) shows the snapshot of *MST* $T(V, \Psi)$ containing 200 sensor nodes in UD Scenario.

In the proposed EAPC algorithm, the goal of the CPS phases is to construct a path for data collection. Fig. 7 shows the constructed paths from the CPS phases for the two experimental scenarios. Each CPS phase starts to traverse the *MST* $T(V, \Psi)$ from the base station. And repeatedly selects the CP with the maximal benefit index from the set of sensor nodes. Finally, the proposed EAPC algorithm constructs a data collection path in a counterclockwise direction from the base station. Figs. 7(a) and 7(b) illustrate the constructed data collection paths for BD Scenario and UD Scenario, respectively.

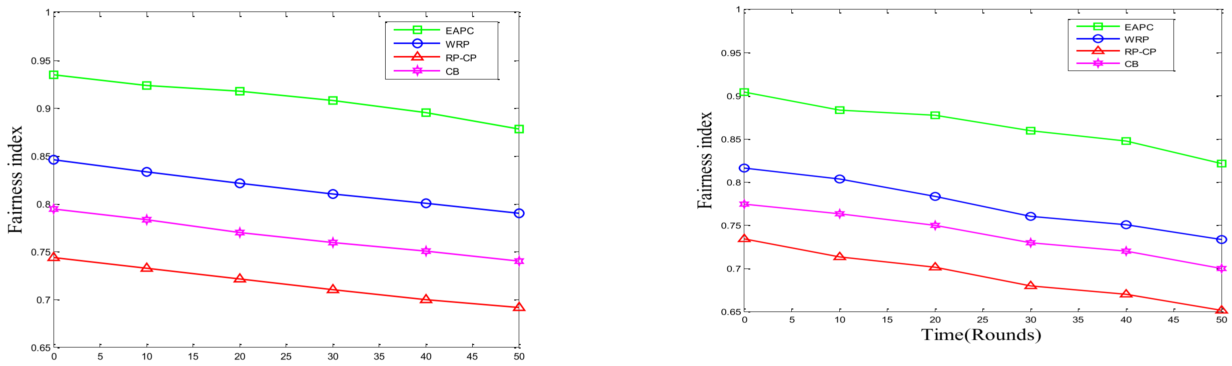
Fig. 8 compares the network lifetimes of the four algorithms with various numbers of sensor nodes, ranging from 50 to 300, for two scenarios. The network lifetime is measured by the time period from the start of network operations to the first time point at which any sensor runs out of energy. The proposed EAPC improves 120, 140 and 100 rounds longer than the lifetimes of CB, RP-CP, and WRP, respectively. The WRP selects CPs and constructs a tour that passes through all CPs within a certain delay bound. However, it does not account for the path length from the location of the current CP to the next CP. As a result, the proposed EAPC has



(a)

(b)

Fig. 8. 8 Network lifetimes for *EAPC*, *WRP*, *CP-RP*, and *CB*. (a) BD Scenario. (b) UD Scenario.



(a)

(b)

Fig. 9. Fairness Index. (a) BD Scenario. (b) UD Scenario.

better performance than WRP. The CB does not consider long data-forwarding paths from sensor nodes to the CPs and nonuniform energy depletion, which reduces the lifetime of the WSN. However, the EAPC measures the benefit of each sensor by considering the forwarding load of each sensor. Therefore, the CPs selected by EAPC can extend the network lifetime. As shown in Fig. 8, the proposed EAPC outperforms WRP, RP-CP, and CB in terms of network lifetime.

Fig. 9 investigates the fairness indices of the compared four algorithms in two scenarios. The lifetime of a WSN substantially depends on the residual energy. If the forwarding load of each sensor can be balanced, the lifetime can be efficiently improved. The following passage examines the *Fairness Index* $\zeta_{fairness}$ of energy consumption. Let n denote the number of sensor nodes. Let $q_i$denote the energy consumption of sensor node s_i in each round. The $\zeta_{fairness}$ is defined by (22) as follows:

$$\zeta_{fairness} = \frac{\left(\sum_{i=1}^{n} q_i\right)^2}{n \times \sum_{i=1}^{n} q_i^2} \tag{22}$$

Fig. 9 compares the fairness indices of four approaches. Because RP-CP creates long forwarding paths from each sensor node to the corresponding CP, it has low fairness. In WRP, the path length from the current location to the next CP is not considered. Therefore, more overhead is required for visiting two adjacent CPs. As a result, few CPs can be joined in the constructed path, leading to an unnecessarily large number of hops from each sensor to the CP. As a result, its fairness decreases with elapsed time. In the proposed EAPC, the mobile sink can visit numerous CPs and can distribute the workload of data forwarding to numerous CPs. Therefore, the fairness index of EAPC is close to 1. In all tested cases, the proposed EAPC outperformed WRP, RP-CP, and CB schemes in terms of the fairness index. Moreover, the BD Scenario is better than UD Scenario for data collection. This occurs because BD Scenario applies random deployment to deploy the sensors and hence the sensor density of the whole region is balanced. As a result, the distance between any two sensors will not be affected by the hole. However, the distribution of sensor nodes in the UD Scenario results in several big holes, which lead to a large average distance between any two sensors. As compared with the BD Scenario, each sensor in UD Scenario in average consumes more energy for transmitting data to its parent in the constructed tree.

Fig. 10 compares the energy consumption of the proposed mechanism with different policies. Two data collection policies were applied, including "collect while moving" and "stop and collect." The curves labeled with "BD" and "UD" represent the results obtained by considering BD Scenario and

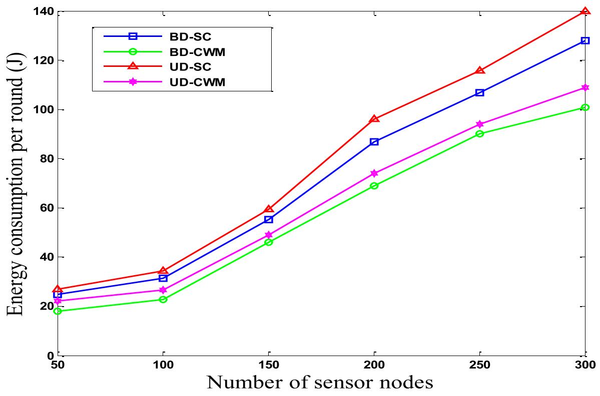


Fig. 10. Comparison of "collect while moving" and "stop and collect" in terms of the energy consumption per round.

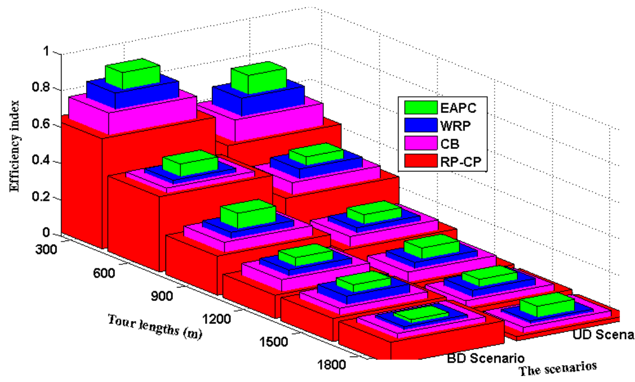


Fig. 11. Comparison of the proposed *EAPC*, *WRP*, *CB,* and *RP-CP* mechanisms in terms of efficiency index with various tour lengths in two scenarios.

UD Scenario, respectively. Furthermore, the curves labeled with "CWM" and "SC" denote the applied policies "collect while moving" or "stop and collect," respectively. The number of sensors is varied from 50 to 300. The curve labeled with BD-CWM consumes less energy than the other curves. More specifically, BD-CWM saves 14%, 31% and 40% energy consumptions as compared with UD-CWM, BD-SC, UD-SC, respectively. The BD-CWM mechanism displays the best performance. It outperforms the other three curves in terms of energy consumption. This occurs because the distribution of sensor nodes in the UD Scenario has big holes, which lead to a larger average distance between any two sensors and hence require more energy consumption for transmitting data from one sensor to another. The policy of "collect while moving" causes the mobile sink to collect data when it passes some sensors. This reduces the amount of data forwarding and hence yields more favorable performance.

Fig. 11 shows the efficiency index, which considers the two parameters, the length of the path and number of CPs. Two scenarios were considered in the simulation. Let $I_{efficiency}$ denote the efficiency index, which is equal to unused resources multiplied by the average length cost. A small value of unused resources and a small value of the path length cost for each CP can lead to a large value of efficiency index, as shown in (23).

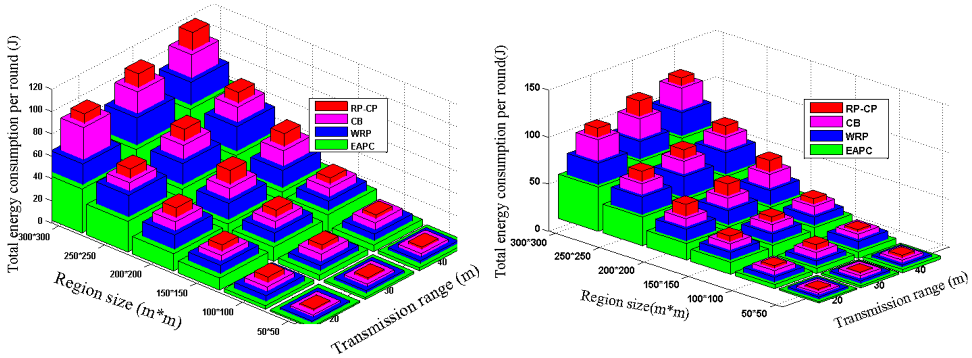
$$I_{effciency} = 1 - \left[\left(1 - \frac{L_{actual}}{L_{max}} \right) \times \left(\frac{L_{actual}}{\rho} \right) \right] \tag{23}$$

where L_{actual} denotes the actual length of the mobile sink tour paths in each round and ρ denotes the number of CPs. As shown in Fig. 11, in BD scenario, the efficiency indices of EAPC, WRP, CB and RP-CP algorithms are 0.897, 0.803, 0.721 and 0.605, respectively. The proposed mechanism EAPC, outperformed the other approaches, because applying EAPC results in higher resource utilization and smaller average path length for each CP, compared with the other three mechanisms. In UD scenario, the efficiency indices of EAPC, WRP, CB and RP-CP algorithms are 0.743, 0.642, 0.551 and 0.435, respectively. In general, the BD scenario is a better WSN environment for data collection than the UD scenario.

Fig. 12 compares the total energy consumption of sensors in each round for the four algorithms. The region size varied from 50 m × 50 m to 300 m × 300 m, whereas the transmission range varied from 20 m to 40 m. The four algorithms exhibited a similar trend, in that the total energy consumption increased with the region size. This occurred because a large region can lead to a large average distance between any two sensors, hence, more energy is consumed when transmitting data from one sensor to another. In average, EAPC consumes less energy than the other three algorithms. More specifically, it saves 13%, 25%, 39% energy consumptions, as compared with WRP, CB and RP-CP, respectively. The proposed EAPC yielded the best performance and outperformed the other three algorithms. The RP-CP had the highest energy consumption because it had the largest average number of hops from each sensor to each CP. The CB had higher energy consumption than the WRP and EAPC because the selected CPs had long data-forwarding paths from sensor nodes to CPs.

Fig. 13 shows the energy consumption of the four compared mechanisms in two scenarios. The number of sensor nodes varied from 50 to 300. As shown in Fig. 13, the four mechanisms exhibited a similar trend, in that the performance of BD Scenario was more favorable than that of UD Scenario and the energy consumption increased with the number of sensor nodes. In average, EAPC saves 44% energy consumption, as compared with RP-CP. Moreover, CB and WRP save 18% and 30% energy consumptions, respectively, as compared with RP-CP. The EAPC outperformed the other three mechanisms in terms of energy consumption. The main reason is that EAPC considers the path length as a vital parameter and thus can select more CPs than the other three mechanisms. Thus the average number of hops from each sensor to the CP is shorter. Another reason is that the proposed EAPC requires the mobile sink to collect data while moving.

Figure 14 shows the network lifetimes of the four compared mechanisms in two scenarios. In literature, the definitions of the network lifetime varies significantly. A lot of studies measured the network lifetime from the time point that the network starts working to the time point that the first node runs out its energy. Some other studies assumed that the network is dead when the few nodes or the average number of nodes have been dead. This experiment aims to observe the network lifetime by applying various definitions. Initially, there are 260 sensors deployed in the monitoring region. The x-axis depicts the number of working sensors while the y-axis depict the lifetime measured by the number of rounds. Figure 14 investigates



(a) (b)

Fig. 12. Comparison of energy consumptions of four algorithms. The region sizes ranged from 50 × 50 to 300 × 300. (a) BD Scenario. (b) UD Scenario.

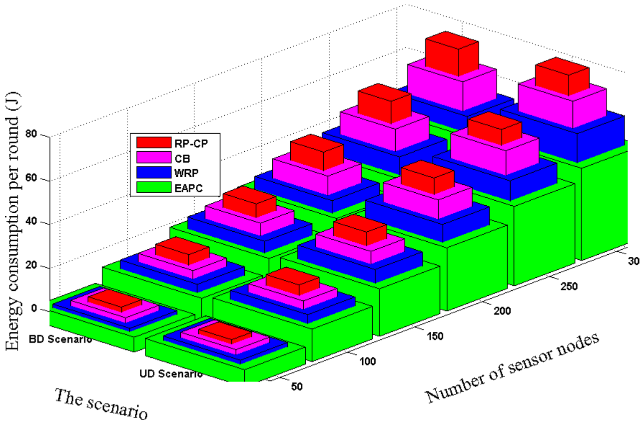


Fig. 13. Comparison of the four algorithms in terms of energy consumption in BD Scenario and UD Scenario.

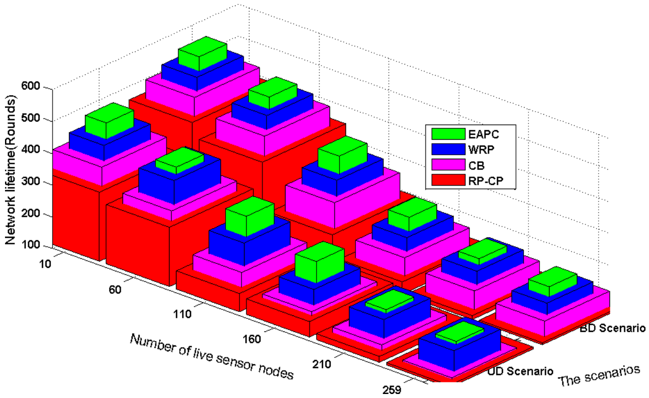


Fig. 14. Comparison of the four algorithms in terms of network lifetime in BD Scenario and UD Scenario.

the network lifetime which can be observed by applying various definitions, including that 1(first), 50, 100, 150(average number), 200 or 250(almost all) nodes have been dead. When the first sensor runs out its energy, the number of working sensors is 259. The whole network can be considered to be dead when we apply the lifetime definition that the first node runs out its energy. EAPC, WRP, CB and RP-CP algorithms have performed 168, 143, 121 and 102 rounds, respectively. This indicates that the proposed EAPC have longest lifetime, as compared with the other existing algorithms. When the number of working sensors is 110, almost a half number of sensors have been dead. EAPC, WRP, CB and RP-CP algorithms have performed 308, 281, 259 and 199 rounds, respectively. When the number of working sensors is 10, almost all sensors have been dead. EAPC, WRP, CB and RP-CP algorithms have performed 589, 541, 508 and 461 rounds, respectively. The results are similar that the proposed EAPC outperforms the other thee compared algorithms, in terms of network lifetime.

VI. CONCLUSION

Data collection is one of the most important concerns in WSNs. Mobile sink can help reduce the energy consumption of sensor nodes because mobile sink can visit some sensor nodes and collect data from them while moving along a specific path. This study has proposed a method for constructing a data collection path and selecting the appropriate sensor nodes to serve as CPs. The proposed EAPC comprises three phases: initialization, collection point selection, and path construction phases. In initialization phase, the EAPC constructs a minimum spanning tree that is rooted at the base station. In the collection point selection and path construction phases, the EAPC selects an appropriate set of data CPs, constructs a data collection path, and collects data from highly burdened CPs. Performance evaluation shows that the proposed EAPC outperforms existing schemes in terms of network lifetime, energy consumptions, fairness index, and efficiency index.

REFERENCES

[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. ACM Int. Workshop Wireless Sens. Netw. Appl.*, Atlanta, GA, USA, Sep. 2002, pp. 88–97.

[2] Y. Zeng, C. J. Sreenan, L. Sitanayah, N. Xiong, J. H. Park, and G. Zheng, "An emergency-adaptive routing scheme for wireless sensor networks for building fire hazard monitoring," *Sensors*, vol. 11, no. 3, pp. 2899–2919, 2011.

[3] J. Zhang, W. Li, Z. Yin, S. Liu, and X. Guo, "Forest fire detection system based on wireless sensor network," in *Proc. 4th IEEE Conf. Ind. Electron. Appl.*, Xi'an, China, May 2009, pp. 520–523.

[4] Y. Zhang, X. Sun, and B. Wang, "Efficient algorithm for k-barrier coverage based on integer linear programming," *China Commun.*, vol. 13, no. 7, pp. 16–23, Jul. 2016, doi: 10.1109/CC.2016.7559071.

[5] M. P. Kolba, W. R. Scott, and L. M. Collins, "A framework for information-based sensor management for the detection of static targets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 1, pp. 105–120, Jan. 2011.

[6] M.-S. Pan, L.-W. Yeh, Y.-A. Chen, Y.-H. Lin, and Y.-C. Tseng, "A WSN-based intelligent light control system considering user activities and profiles," *IEEE Sensors J.*, vol. 8, no. 10, pp. 1710–1721, Oct. 2008.

[7] N. K. Suryadevara and S. C. Mukhopadhyay, "Wireless sensor network based home monitoring system for wellness determination of elderly," *IEEE Sensors J.*, vol. 12, no. 6, pp. 1965–1972, Jun. 2012.

[8] N. Dessart, H. Fouchal, and P. Hune, "Distributed diagnosis over wireless sensors networks," *Concurrency Comput., Pract. Exper.*, vol. 22, no. 10, pp. 1240–1251 2010.

[9] T. Bernard and H. Fouchal, "Efficient communications over wireless sensor networks," in *Proc. Global Telecommun. Conf.*, Dec. 2010, pp. 1–5.

[10] B. Yahya and J. Ben-Othman, "Energy efficient and QoS aware medium access control for wireless sensor networks," *Concurrency Comput., Pract. Exper.*, vol. 22, no. 10, pp. 1252–1266, 2010.

[11] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Netw.*, vol. 1, nos. 2–3, pp. 215–233, Sep. 2003.

[12] M. Ma and Y. Yang, "Data gathering in wireless sensor networks with mobile collectors," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. (IPDPS)*, Apr. 2008, pp. 1–9.

[13] R. Sugihara and R. K. Gupta, "Optimal speed control of mobile node for data collection in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 1, pp. 127–139, Jan. 2010.

[14] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Trans. Mobile Comput.*, vol. 6, no. 4, pp. 395–410, Apr. 2007.

[15] M. Zhao and Y. Yang, "Bounded relay hop mobile data gathering in wireless sensor networks," *IEEE Trans. Comput.*, vol. 61, no. 2, pp. 265–277, Feb. 2012.

[16] G. Xing, T. Wang, Z. Xie, and W. Jia, "Rendezvous planning in wireless sensor networks with mobile elements," *IEEE Trans. Mobile Comput.*, vol. 7, no. 12, pp. 1430–1443, Dec. 2008.

[17] K. Almi'ani, A. Viglas, and L. Libman, "Energy-efficient data gathering with tour length-constrained mobile elements in wireless sensor networks," in *Proc. 35th IEEE Conf. LCN*, Denver, CO, USA, Oct. 2010, pp. 582–589.

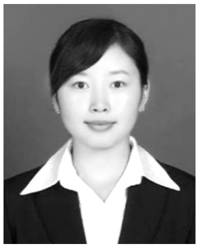
[18] H. Salarian, K.-W. Chin, and F. Naghdy, "An energy-efficient mobile-sink path selection strategy for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2407–2419, Jun. 2014.

Weimin Wen received the B.S. degree from Xidian University, China, in 2005, and the M.S. degree from the Xi'an University of Science and Technology in 2008.

She is currently an Engineer at the School of Computer and Information Engineering, Chuzhou University, Chuzhou, China. Her current research interests include wireless sensor networks and smart healthcare.

Shenghui Zhao received the M.S. degree from the Hefei University of Technology, China, in 2003, and the Ph.D. degree from Southeast University, China, in 2013.

She is currently a Professor with the School of Computer and Information Engineering, Chuzhou University, Chuzhou, China. Her current research interests include trusted computing, wireless networks, smart healthcare, and Internet of Things.



Cuijuan Shang received the B.S. degree from Zhengzhou University, China, in 2008, and the M.S. degree from the Shandong University of Technology in 2011.

She is currently an Assistant at the School of Computer and Information Engineering, Chuzhou University, Chuzhou, China. Her current research interests include wireless sensor networks, smart healthcare, and Internet of Things.

Chih-Yung Chang (M'05) received the Ph.D. degree in computer science and information engineering from National Central University, Zhongli, Taiwan, in 1995.

He is currently a Full Professor with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. His current research interests include Internet of Things, wireless sensor networks, *ad hoc* wireless networks, and long term evolution broadband technologies. He has served as an Associate Guest Editor for several SCI-indexed journals, including the *International Journal of Ad Hoc and Ubiquitous Computing* from 2011 to 2014, the *International Journal of Distributed Sensor Networks* from 2012 to 2014, *IET Communications* in 2011, *Telecommunication Systems* in 2010, the *Journal of Information Science and Engineering* in 2008, and the *Journal of Internet Technology* from 2004 to 2008.

Dr. Chang was an Area Chair of the IEEE International Conference on Advanced Information and Applications (AINA) 2005, the Taiwan Academic NETwork (TANET) Conference 2000, TANET 2010, the IEEE International Symposium on Wireless IP (WisCom) 2005, the IFIP International Conference on Embedded And Ubiquitous Computing 2005, the IEEE International Conference on Information Technology: Research and Education 2005, and the IEEE AINA 2008, and the Program Co-Chair of the IEEE International Workshop on Multimedia Network Systems and Applications 2005, the Mobile and Ubiquitous Technologies Enhanced Learning Conference 2006, the Workshop on Wireless, Ad Hoc, and Sensor Networks 2007, and the ACM International Workshop on Sensor, Ad Hoc, and Mesh Networks 2008.