A data gathering algorithm for a mobile sink in large-scale sensor networks

E.M. Saad, M.H. Awadalla, M.A. Saleh, H. Keshk, and R.R. Darwish

Department of Electronics and Communication, Helwan University, Cairo, Egypt

Abstract — Sink mobility is one of the most comprehensive trends for information gathering in sensor networks. This way of information gathering has a prominent role in balancing the energy consumption among sensor networks, and culling the hotspots problem of sensor networks. In this paper, a well planned adaptive moving strategy for a mobile sink in large-scale, hierarchical sensor networks is presented. The mobile sink traverses the entire network uploading the sensed data from cluster heads in time driven scenarios. The mobile sink trajectory is planned such that all heads require no multi-hop relays to reach the mobile sink. The proposed system aims at extending the lifetime of the sensor network by achieving a high level of energy efficiency and fair balancing of energy consumption across all network heads. Furthermore, reducing the loss that data incur due to buffer overflow. Extensive simulations are conducted in order to validate the proposed strategy. The adopted data gathering scheme outperforms the static sink scheme and periphery scheme in terms of life time elongation, and scalability.

Keywords— Sensor Networks, Sink mobility, Network lifetime, Bees Algorithm

I. INTRODUCTION

Wireless sensor networks constitute an emerging technology that has received recently a significant attention both from industry and academia. On the one hand, there is an ever-widening range of attractive applications (e.g., disaster and environmental monitoring, wildlife habitat monitoring, intrusion detection, security surveillance) sensor networks can be used for. On the other hand, sensor networks are self-organizing ad-hoc systems where optimized energy consumption is of paramount importance [1].

Various types of energy efficient data gathering mechanisms have been investigated for large scale networks. Usually the readings taken by the sensor nodes are relayed to a static base station for processing using a multi-hop network formed by the sensor nodes[3-6]; therefore, the sensors that are close to the sink would become hotspots and die earlier than other sensors because they have to transmit huge amounts of data for other sensors. However recent studies [7-9] have proposed sink mobility as an efficient solution for data gathering problem. In such networks the sink changes its position from time to time, traverses the network, and collects sensed data from nearby nodes while moving. Therefore, employing a mobile device to collect data can reduce the effects of the hotspots problem, balance energy consumption among sensor nodes, and thereby prolong the network lifetime to a great extent [10, 11].

Various types of moving strategies have been considered in literature for the mobile sink. These strategies can be broadly classified as random, autonomous, and planned moving strategies. In this paper, the problem of planning an arbitrary moving trajectory for a mobile sink in hierarchical structure sensor networks is considered. The main aim of the proposed strategy is to achieve even distribution for the traffic load among the network cluster heads, in order to achieve considerable network life time elongation compared with sensor networks with a static sink.

The rest of the paper is organized as follows: Section II discusses the most related work. Section III introduces the proposed moving strategy. Section IV shows the performance evaluation results. Finally section VI concludes the paper.

II. BACKGROUND

Various types of mobility strategies have been studied extensively in the literature. The authors in [12, 13], used radio-tagged zebras and whales as mobile nodes to collect sensed data in a wild environment. These animal based nodes wander randomly in the sensing field and exchange sensed data only when they move close to each other. Thus, sensor nodes in such a network are not necessarily connected all the time. Moreover, the mobility of randomly moving animals is hard to predict and control; thus, the maximum packet delay cannot be guaranteed.

In [9, 10], the authors have exploited sink controlled movement in order to improve data delivery performance. In
their work, the path planning for a mobile sink was formulated as the mobile element scheduling (MES) problem based on the assumption that a mobile element visits each sensor node to collect data. Although the strategies in which a mobile device visits each sensor node or awake one-hop neighbor nodes to collect sensed data can save the most energy, but the main disadvantage of these approaches in [9, 10] is the increased data collecting latency because the mobile sink has to traverse the transmission range of each sensor one by one to collect data.

In [11] it has been theoretically proved that, under the conditions of a short path routing and a round network region, moving along network periphery is the optimum strategy for a mobile sink. Their analysis was based on an ideal load-balance short path routing protocol and the simulation were performed without consideration of Medium Access Control protocols (MAC) effects.

An attempt to determine specific sink movement for energy optimization is presented in [14]. The authors propose to employ multiple sink nodes that periodically change their locations, and present an ILP (Integer Linear Programming) model to obtain the optimal positions of these sinks. A linear programming solution to determine the movement of the sink and its sojourn time in different points of the network is given in [15] as well. Both the sensors and the sink are placed on a bi-dimensional grid. The sink moves along the grid, sojourns times in the specific grid points being calculated so as to maximize the network lifetime.

Recently, several researches have investigated the autonomous moving strategies for mobile sinks. In [7] the authors argue that multi-hop communication results in fast depletion of the sensors neighboring the sink. Therefore, they propose an autonomous moving strategy for the mobile sink, in this approach, a mobile sink approaches the nodes with high residual energy to force them to forward data for other nodes and tries to avoid passing by the nodes with low energy.

### III. The PROPOSED MOVING STRATEGY

Our model problem focuses on gathering data in large scale networks. Thereby as an attempt of minimizing the whole tour length, and elevating the overall network energy savings as well, the network is organized into several clusters, where cluster head node manages the communication with the mobile sink on behalf of their members. Further details about cluster head selection and cluster formation algorithms have been formulated by the author at [16].

In this work, the problem of planning the moving path for a mobile sink is investigated. It is assumed that a mobile sink equipped with a powerful transceiver and battery gathers the sensed data periodically from network’s head. Each mobile sink tour begins from a fixed starting point then the mobile sink moves along a well planned moving path that traverses the whole network, gathers the sensed data from heads directly while moving, and finally returns to the starting point.

The key idea of the adapted moving strategy is to search within a space of possible configurations for a solution path, with the objective of balancing the trade off among, energy efficiency, total path length, and buffer overflow deadlines. Such path is accomplished by performing a search for a shortest path on an implicit set of pre-determined path-points in state space. Energy efficiency can be realized by searching for a path that traverses all network head nodes’ transmission range. Moving along such path will balance the traffic load among the heads, so the network lifetime can be prolonged. The adopted moving strategy can be roughly broken into two tasks. The first task concerning with path-points identification, while the second task deals with path optimization. A flow chart that depicts the proposed moving strategy is shown in Fig. 1.

```
A. Path-points identification model.

In order to accomplish the adopted moving strategy goal of producing a well planned path, it is first necessary to develop a
```
way-point identification model. Such model is created to indicate a series of interested points, through which the mobile sink is required to pass. The resultant path-points which some how present discretised version of the full terrain, must account for the overall goal of minimizing the maximum traffic load at each head.

Analyzing the full terrain searching for the most effective path-points is impractical also infeasible process, specially in a large-scale terrain. So this work presented a simple non-expensive path-point identification model. First, an agglomerative clustering approach is used to organize all network heads into distinct groups. The agglomerative approach, starts with each head forming a separate group. It successively merges the groups close to one another in terms of network hops, maximum two hops away, until the size of all groups are maximized. The size of any group is maximized if no more heads can be added to it. After cluster formation phase, the path-points necessary to form the path can be simply set to the centroid points of these clusters.

Therefore the proposed path-points identification model guarantees that all network heads are one hop away from the mobile sink path, which enables each head to send data directly to the mobile sink without relaying on other heads. Figure 2 illustrates the proposed path-points identification model. Initially all nodes are put into set $H$ ( $H$ is the set of all heads), where $n$ is the total number of heads in the network. In order to construct $cm$ which represents the set of heads in the $m$th group, one can start from an empty set $cm$. A node $t_j \in H$ is added into $cm$ and removed from $H$. After that, all the neighbors (maximum two hops away) of $t_j$ are also added into $cm$ and removed from $H$. Then the centroid of the $m$th group is calculated and added to the set of path-points. The algorithm is continued until $H$ is empty.

1. Add all head nodes into a set $H$;
2. Construct an empty set $W$ for the path-points;
3. $m=0$; // where $m$ is the number of clusters
4. While ($H$ is not empty);
5. $m++$;
6. Construct a new empty set $cm$ for the $m$th cluster;
7. Pick a node $n_i$ from $H$, add it into set $cm$, and remove it from $H$;
8. Find and add all two-hops neighbors of $n_i$ into $cm$, and remove it from $H$;
9. Calculate the centroid $t_m$ of the $m$th cluster, add them to the path-points set $W$;
10. End while;
11. Return path-points set $W$;

![Fig.2 Path–points identification algorithm](image)

### B. Mobile sink path planning

After the path–points identification model returns successful representation set for the terrain, all that remain is to find the best route passes through those points. The proposed path planning approach phrases this issue as a constrained optimization problem by defining a cost function favoring shorter path, and adding constrain to avoid heads’ buffer overflow. This paper discusses the use of the Bees algorithm [17] as a path optimization tool and an appropriate fitness function is developed to incorporate optimality criterions.

#### Bees Algorithm

Bees Algorithm is an optimization algorithm inspired by the natural foraging behaviors of honeybees to find the optimal solution [18]. Figure 3 shows the pseudo code for the algorithm. The algorithm requires a number of parameters to be set, namely: number of scout bees $(n_s)$, number of sites selected out of $n$ visited sites $(m_s)$, number of best sites out of $m$ selected sites $(e)$, number of bees recruited for best $e$ sites $(nep)$, number of bees recruited for the other selected $(m-e)$ selected sites $(ns_p)$.

![Fig. 3 Bees algorithm](image)

#### Methodology of Bees algorithm

This section illustrates the methodology and formulation for Bees algorithm for obtaining the optimal sink path. This methodology incorporates, problem representation, and the formulation of the fitness function that gives to each individual (i.e. possible configuration route) a measure of performance.
Problem representation

The main parameters of the Bees algorithm used in this work are summarized in Table 1. The set of path-points are numbered as i=1….N and each way-point has a coordinates (x_i, y_i). Each bee in the population specifies a possible arrangement of path-points, such configuration is encoded as a vector comprising N way-points. This vector can be interpreted as the order in which the path-points are visited.

<table>
<thead>
<tr>
<th>Bees Algorithm parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>n</td>
<td>50</td>
</tr>
<tr>
<td>Number of selected sites</td>
<td>m</td>
<td>30</td>
</tr>
<tr>
<td>Number of elite sites</td>
<td>e</td>
<td>3</td>
</tr>
<tr>
<td>Number of bees recruited for elite sites</td>
<td>nep</td>
<td>30</td>
</tr>
<tr>
<td>Number of bees recruited for the other (m-e) selected sites</td>
<td>nsp</td>
<td>10</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>itr</td>
<td>500</td>
</tr>
</tbody>
</table>

Fitness Function

In the case under investigation, the fitness function is a weighting function that measures the quality of a specific configuration. This function is minimized by the Bees algorithm. In the simplest form, the fitness function F can be considered as the total length of the mobile sink journey (eq. 1).

\[
F = \sum_{i=1}^{N-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}
\]  

(1)

However, in the proposed moving strategy, the length of each tour is restricted by the head buffer overflow deadline, as the sensed data at each head must be gathered by the mobile sink before the buffer overflows. So in order to incorporate this constraint into the algorithm, a penalty \( \lambda \) is thereby assigned to each tour length that leads to a head buffer overflow, therefore the fitness function can be described as (eq. 2).

\[
F = \sum_{i=1}^{N-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + n\lambda}
\]  

(2)

where \( n \) is the number of heads that miss their overflow deadline.

V. SIMULATION

In simulation, a complete environment comprises 2000 sensor nodes is developed to validate the proposed work. Sensor nodes are randomly dispersed into a field with dimensions 1000 meters* 500 meters square region. This work, adopted the same radio energy model described in [5]. In the adopted radio model, energy is expended to serve: 1) digital electronics, \( E_{elec} \), \( E_{elec} \) depends on factors such as digital coding, modulation, and filtering; and 2) communication, \( E_{amp} \) where \( E_{amp} \) varies according to the distance \( d \) between a sender and a receiver. For a relatively short distance, the propagation loss is modeled as being inversely proportional to \( d^2 \), whereas for longer distance, the propagation loss is modeled as being inversely proportional to \( d^4 \). Therefore, to transmit and receive \( n_b \) bits in a distance \( d \), the radio expense the following energy, respectively:

\[
E_{Rx} = \begin{cases} \frac{n_b (E_{elec} + E_{amp} \times d^2)}{d}, & \text{if } d < d_o, \\ \frac{n_b (E_{elec} + E_{amp} \times d^4)}{d}, & \text{if } d \geq d_o, \end{cases}
\]  

(3)

\[
E_{Rx} = n_b \times E_{elec}
\]

This work assumes different heads overflow deadlines, so in order to simulate real scenarios, the overflow deadline assigned to each head was based on the number of nodes joining it whereas the overflow deadline of each head deceases as the number of joining members increases. Thus, in order to avoid data overflow, the sensed data at each head has to be uploaded to the mobile sink before its buffer overflow deadline. The main simulation parameters are listed in Table 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network grid</td>
<td>From (0,0) to (1000, 500)</td>
</tr>
<tr>
<td>Total no. of sensors</td>
<td>2000</td>
</tr>
<tr>
<td>Starting point</td>
<td>(0, 250)</td>
</tr>
<tr>
<td>Mobile sink speed</td>
<td>40 m/minute</td>
</tr>
<tr>
<td>Data packet size</td>
<td>100 bytes</td>
</tr>
<tr>
<td>( E_{elec} )</td>
<td>50 nJ/bit</td>
</tr>
<tr>
<td>( E_{amp} )</td>
<td>0.0013pJ / bit/m^4</td>
</tr>
<tr>
<td>Threshold distance</td>
<td>40m</td>
</tr>
</tbody>
</table>

The initial layout of the network heads and the generated path-points are shown in figure 4(a) and 4(b) respectively. The mobile sink enters the field and starts its journey from a start point with coordinates (0m, 250m), collects the sensed data from all sensors, and returns back to the start point after a tour. It was assumed that the location information of sensors has been obtained during explanatory phase. Based on that information, the mobile sink calculates its route through the proposed moving strategy. Figure 4(c) and figure 4(d) show the mobile
sink trajectory in case of fixed, and variable heads’ overflow deadlines respectively.

![Fig. 4](image1)

Fig. 4 (a) initial layout of the network (b) Set of way-points (c) Sink moving path with fixed head overflow deadlines (d) Sink moving path with variable head overflow deadlines

![Fig. 5](image2)

Fig. 5 The average energy consumption in a round for all three strategies

In order to evaluate the energy conservation capability of the proposed algorithm, the network lifetime performance of the proposed moving strategy is compared along with that of other two data-gathering schemes: Scheme 1- a conventional strategy where a stationary sink node locates at the network center, Scheme 2- a periphery moving strategy where a sink moves along the periphery of the network[11].

Figure 5 presents the average energy consumption of the entire network per round, for the three different strategies. Ten different iterations were performed, and a cumulated average is calculated. It can be seen that the adopted moving strategy consumes less energy; it is around 30% better than the periphery strategy, and 50% better in case of the conventional strategy with static sink.

Figure 6(a) shows the network lifetimes of the three strategies with ten different node deployments, where the lifetime is the time until the first node dies (A node is considered “died” if it has lost 99 percent of its initial energy). The results indicate that, the proposed moving strategy clearly improves network lifetime over the other two strategies for all node deployment. This because the mobile sink trajectory was planned such that all heads are at most one hop away from the moving path, so each node can now send its data directly to the sink without relying on multi-hoping routing protocols. Similar results are obtained for the number of rounds until fifty percent of node death as shown in figure 6(b).

Scalability of the proposed moving strategy was also investigated. In this experiment the network lifetime of the three schemes were measured under different node densities and the results are shown in figure 7(a). It can be seen that, as the node density increases, the network life time of both the stationary scheme, and periphery scheme decreases, as the sensor nodes
near the sink have to forward more data in one gathering scheme. However, life time of the proposed moving strategy shows better scalability as the lifetime obtained under the proposed moving strategy keep increasing as the node density increases. Similar results are obtained for the number of rounds until fifty percent of node death as shown in figure 7(b).

Figure 8 demonstrates a comparison between network life time for the proposed strategy and the autonomous moving strategy proposed in [7], where the network life time for the static sink strategy is taken as reference. In this experiment the same network dimension, and number of sensor nodes considered in [7] is used for comparison purposes. The achieved results show that the proposed moving strategy outperforms the autonomous moving strategy, ensuring better scalability and longer network life time.

VI. CONCLUSIONS

Enhancing energy-efficiency is primordial in a wireless sensor network. This paper proposed a moving strategy for a mobile sink in hierarchical, large-scale networks. The mobile sink starts data gathering tour from a fixed starting point, traverses the network, collects data, and then returns to the starting point. Cluster heads upload sensed data to mobile sink when the mobile sink traverses its transmission range. Packets from all cluster heads did not need any multi-hop relays to reach the sensor network. Thus, energy consumption incurred by multi-hop routing protocols is minimized. Simulation experiments revealed that our moving strategy outperforms other data gathering schemes, including static sink strategy, periphery strategy in terms of network lifetime and scalability.

REFERENCES