EEDARS: An Energy-Efficient Dual-Sink Algorithm with Role Switching Mechanism for Event-Driven Wireless Sensor Networks

Mohammadreza Eslaminejad[1], Shukor Abd Razak[1] and Abdul Samad Haji Ismail[1]
[1] Faculty of Computer Science and Information Systems,
Universiti Teknologi Malaysia (UTM),
Skudai, 81300 – Malaysia
[e-mail: eslaminejad@ieee.org; {shukorar, abdsamad}@utm.my]
*Corresponding author: Mohammadreza Eslaminejad

Received June 26, 2012; revised September 16, 2012; accepted October 8, 2012;
published October 29, 2012

Abstract

Energy conservation is a vital issue in wireless sensor networks. Recently, employing mobile sinks for data gathering become a pervasive trend to deal with this problem. The sink can follow stochastic or pre-defined paths; however the controlled mobility pattern nowadays is taken more into consideration. In this method, the sink moves across the network autonomously and changes its position based on the energy factors. Although the sink mobility would reduce nodes' energy consumption and enhance the network lifetime, the overhead caused by topological changes could waste unnecessary power through the sensor field. In this paper, we proposed EEDARS, an energy-efficient dual-sink algorithm with role switching mechanism which utilizes both static and mobile sinks. The static sink is engaged to avoid any periodic flooding for sink localization, while the mobile sink adaptively moves towards the event region for data collection. Furthermore, a role switching mechanism is applied to the protocol in order to send the nearest sink to the recent event area, hence shorten the path. This algorithm could be employed in event-driven and multi-hop scenarios. Analytical model and extensive simulation results for EEDARS demonstrate a significant improvement on the network metrics especially the lifetime, the load and the end-to-end delay.

Keywords: Wireless sensor networks, energy-efficiency, dual-sink, sink mobility, event-driven applications, multi-hop routing.

Mohammadreza Eslaminejad would like to thank Dr. Hassan Chizari for his help and support.

http://dx.doi.org/10.3837/tiis.2012.10.002

1. Introduction

Despite of ad-hoc networks that a pair of nodes plays the role of source and destination, in WSNs, sensor nodes produce data packets and transmit them to one or more sinks. This collected data will be later sent to the end users to be processed. In such networks, sensor nodes intensively suffer from constrained resources specially battery and computational power [1]. In many cases, a large number of sensor nodes should be released in a harsh environment for a long period of time. Therefore, replacing the batteries of dead nodes is not as easy as deploying them in an interested area and in many cases like battlefield scenarios is even impossible. Such nodes are not able to do their duties such as inter-communication or sensing job.

In recent years, network researchers suggest a variety of protocols for data delivery and diffusion in WSNs. However, they are different from the application point of view. In a *time-driven* scenario, the sink periodically receives the data packets from all sensor nodes. Unlike the previous category, in a *query-driven* case the sink send a request to the sensor field in order to achieve the interested data. Eventually, in an *event-driven* scenario the sensor nodes only report the portion of sensed data that is worth sending [2].

It seems a large number of approaches in literature covers the time-driven tree-based [3][4] or hierarchical [1][5][6] scenarios. In fact, event-driven and multi-hop manner is more suitable for many applications such as animal movement tracking, seismic activity monitoring and intrusion detection [7]. Therefore, we only attend to this scenario in our paper. In fact, minimizing the energy utilization in sensor nodes in order to prolong the network lifetime is the most important goals of these solutions. Although most of these approaches focus on different layers of the protocol stack, there is a high tendency for energy-efficient mechanisms to relay data over multiple hops from the sensor nodes to the sink at the network layer [5].

One of the issues of multi-hop forwarding is the phenomenon of energy depletion around the sink that called “sink neighborhood problem”. These nodes lose their remaining energy much faster than the other nodes which are far from the sink in the network [8][9][10][11]. The reason is that the sink’s neighbors with one hop distance are involved in relaying the data generated from all over the network. Therefore, the sink’s neighbors stop their functionality so that the void area made by the dead nodes around the sink leaves the sink unreachable by other sensor nodes. The researchers propose two solutions for this problem. The first approach refers to deploying some supplementary sensors by the help of mobile robots during the network lifetime. Therefore, we follow the second one which is using multiple sinks [12]. It seems employing both mobile and static sinks in super large scale networks for mission-critical applications from fire detection to environmental monitoring will be feasible in the near future [12]. With an efficient mobile sink scheme, the neighboring nodes around the sink changes periodically. It can improve network connectivity and lifetime [12][13]. However, Sink mobility in WSNs is very challenging.

The first issue should be addressed here is where the sink could go. Many factors make it so hard but the most important one is the unlimited possible locations that the sink could be moved to. Additionally, wireless sensor networks are dynamic naturally because the sensors’ condition and the sources of data are variable and may change time by time. Therefore an optimization scheme is necessary periodically each time the sink decides to change its position [4][14]. Since our proposed algorithm addresses only the tandem events scenario, moving one of the sinks directly towards the event region seems a smart choice. In addition, we have applied a distance threshold to the scheme based on the sink speed to ensure that it could return to the central point and play the role of static sink before the next round. The second problem caused by sink mobility is that sensor nodes lose the location of sink after each movement. In such case, the sink should flood its position through the network periodically imposing large amount of overhead on the sensor nodes. Furthermore, there should be a tradeoff between the number of times the sink change its position and the overhead caused by reroute discovery messages [12][13]. It is known as "offset problem," [12] which can cancel out the lifetime gain from sink mobility in WSNs. This problem is addressed at EEDARS algorithm where the new sources which sensed an event send their first packets to the central static sink without any localization overhead. This temporary static sink is responsible to inform the source nodes of the nearest sink for sending the further packets.

In this paper, an event-driven scenario is analyzed as a general case. However, we focus on a special situation where only one event happened in the field at any moment [12]. Thus, an animal movement tracking scenario with a random walk model is intended in this research [7]. The main contribution is to send the nearest sink to the event region to track the object while dispatch another sink to the center of field to avoid any flooding for sink localization at the next round. Furthermore, a switching mechanism changes the sinks' role as static or mobile nodes for sending the nearest sink to the event region. In fact, minimizing the number of control packets and the number of hops between the event sources and the sink is the goal of this approach. Both simulations and analytical modeling show the efficiency of novel dual-sink protocol with role switching pattern. The results are compared to five alternative algorithms to justify the solution. The network metrics show that EEDARS is most suitable for the event-driven multimedia applications [15] in which the number of packets reported by the events is usually high.

The rest of this paper is organized as follows. Related work including some energy-efficient approaches using multiple sinks and/or mobile sinks is presented in Section 2. In Section 3, we discuss the network model and preliminaries. This part consists of energy model for EEDARS algorithm. Based on these analytical modeling, in Section 4, we show how one of the sinks will be chosen by role switching algorithm to move towards the event region. We present our simulation results in Section 5 and finally conclude the paper in Section 6.

2. Related Work

Recently, there are a large number of proposals targeting the lifetime improvement in wireless sensor networks. Among them, multiple sink and mobile sink approaches are two main topics in this area. However, a combination of these two methods is taken into the consideration. In fact, due to the sink isolation problem which caused by energy exhaustion around the sink, distributing the energy utilization in whole sensor field is the most important goals of these solutions.

The authors of [16] presented a random walk pattern with a passive data collection method for data gathering in WSNs. By using passive manner (pull strategy), the sink propagates a request for one-hop or k-hop neighbors in order to motivate them to send their information. They argue that stochastic mechanism can prolong the network lifetime. However, it may increase the end-to-end delay. An algorithm based on predefined (fixed path) sink movement and joint routing is investigated in [17] where a discrete mobility strategy and a continuous mobility pattern are examined in the protocol. The main drawback of continuous movement is that the sink localization should be supported by flooding strategy. According to this approach, the network lifetime could be maximized if the sink moves on the periphery of the sensor field.

The methods proposed in [3][4][18][19][20] use a controlled sink mobility to periodically dispatch the single sink to the places where some of the network metrics especially lifetime and delay will be improved. In [18], the sink collects the sensed data from cluster heads (CHs) when it sufficiently getting close to them. The methods proposed in [3] construct a tree-based routing from each source node to the single sink. It tries to relocate the sink to the places with more safety and better performance by using the concept of neural network. The algorithm presented in [4] uses the same routing structure but the sink will be directed to the sites with high traffic loads with the help of new relay nodes and power control scheme. In [19], the sink repositioning will be occurred according to the maximum stay-value parameter which defined by the number of neighbors and the average residual energy. This method is a combination of AODV-based multi-hop routing protocol and sink mobility. The mobile sink in [7] tries to forecast the events and moves towards them gradually. A group of source nodes in each event point sends the event packets to the sink in a multi-hop manner.

Some researchers employ a hybrid of multi-sink and mobile sink mechanisms in their proposals. In [21], for instance, a heuristic approach is applied to simultaneously control the multiple mobile sinks. The sensor field is divided to several sites and the sink relocates to an unoccupied site if it recognizes that the movement can prolong the network lifetime. A virtual grid pattern is applied in [22] where a Local Event ANnouncer (LEAN) node collects the event packets from the source nodes inside the corresponding grid cell. Later, the nearest mobile sink can do data gathering in one-hop manner from the LEAN node. A hash table-based technique is employed in this protocol to find the location of mobile sink and LEAN node. The algorithm proposed in [23] engages two mobile sinks to move and stop periodically on a predefined diamond-shape path and collect data through multi-hop routing mechanism. Dual-sink approach that presented in [12] and [13] is a kind of hybrid methods in which one of the sinks is permanently static while another one is moving through the field to directly collect the sensed data from one-hop or k-hop neighbors. In fact, this model uses the advantageous of static and mobile sink approaches efficiently. In this paper, we develop a dual-sink protocol to be suitable for event-driven multimedia applications, where events occur in tandem and the number of event packets is high.

3. Network Model and Preliminaries

In this section, we describe our network model as well as mathematical analysis on the power consumption during the network activity. The application which is considered here is an event-driven scenario where the source nodes (sensors which detect an event) only report the portion of sensed data that is more than or less than a specific threshold. The rest of the network should relay the event packets to the sink node in a multi-hop method. Commonly, more than one source node is involved on the same event simultaneously. The packets produced by these sensor nodes are sent to the sink in a parallel manner. The total energy consumption for all events is computed in this section. According to these analyses, we propose an efficient technique for sink relocation to minimize power consumption in Section 4.

3.1. Topology and Data Routing

Our model is a square shape network in which N sensor nodes are located on the grid cross points in $\sqrt{N}$ rows and $\sqrt{N}$ columns. The field is not completely dense, since there are some void areas [2][24] among the network. There are two sink nodes engaged for data collection. Initially, one of them placed at the center of field $(x_S^C,\ y_S^C)$ as temporary static sink while another one is located at a random position $(x_S^M,\ y_S^M)$ through the network to play the role of temporary mobile sink. R_{sense} is the sensing range of each sensor. Since an event-driven application is considered for the network, all nodes that sense the *event* Evt_i $(i \in \{1, 2, 3\ldots, I\})$ within a circle $C_{Evt_i,R_{sense}}$ should periodically send data packets to the sink, until the event exists. These packets will be forwarded to the sink in a multi-hop manner, since the sensor nodes only can relay the messages to neighbors within their radio range R. In our model, $R = R_{sense}$ and the radio range is adjusted in such a way so that each node only can communicate with the maximum four nearest neighbors on the vertical and horizontal lines. Furthermore, there is not any data aggregation among the sensors. Each *round* of protocol is a period of time in which all data related to one event will be sent to the sink node successfully. We employ a multi-hop geographic routing [2] where each node is aware about its coordination, the coordination of sinks and also its one-hop neighbors. We assume all sensor nodes can achieve this location information through GPS-free [25] mechanisms at the time of deployment.

We employ a geographic short path routing in our model both for analytical computations and simulations. As seen in Fig. 1, there is more than one shortest geographic path between the source node and the sink. We only show three routes out of six in this figure. Nevertheless, two perimeter paths (path 1 and path 2) are considered equivalently by the protocol. It avoids choosing path 3 (dotted arrow) which is an interior path [23]. The philosophy behind this is that perimeter paths could decrease the traffic load on the sensors at central part of the field.

3.2. Events

Generally, an *event* is a situation in which something important happened in sensor field. Therefore, it is worth reporting the data related to that event to the sink node. For example, moving an animal within the monitored area could be a desired event in target tracking applications [7]. In this state, the event is a stochastic subset $Evt_i(t)$ of $\sqrt{N} \times \sqrt{N}$. At time t, the event motivates the i^{th} node at position P_i that fits in the condition of Equation 1.

$$d(p_i\ , Evt_i(t) \le\ R_{sense}) \tag{1}$$

At a snapshot of system, there is only one active event where it's coordinate (x_{Evt_i}, y_{Evt_i}) is exactly located on one of the grid cross points. Thus, there are no simultaneous events supposed in the field. When an event appears, all sensor nodes which are in vicinity of the event point and located within the circle $C_{Evt_i,R_{sense}}$, $(i \in \{1, 2, 3\ldots, I\})$ become source node. Thus, they start to send data packets towards the sink [7].

3.3. Energy Consumption Model

In order to calculate the total energy consumption that an event imposes on the sensors, we should add up all send and receive energies. In **Fig. 2**, the sensors that marked as gray are the

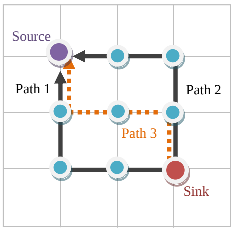


Fig. 1. Path selection in EEDARS.

source and forwarding nodes cooperated in previous events (*Evt₁* and *Evt₂*) while others in dark purple and blue respectively, are related to the current event (*Evt₃*) in the system. The distance D_i which is used in data routing indicates the *geographic* distance of sink S from event Evt_i based on the radio range R (Equation 2). However, the *Euclidean* distance E_i that is calculated in Equation 3 is different and applied to the algorithm in order to determine the nearest sink.

$$D_i = \left|(x_{Evt_i} - x_s)\right| + \left|(y_{Evt_i} - y_s)\right|, \quad i = 1, 2, 3, \dots, I \tag{2}$$

$$E_i = \sqrt{(x_{Evt_i} - x_s)^2 + (y_{Evt_i} - y_s)^2}, \quad i = 1, 2, 3, \dots, I \tag{3}$$

We use the same energy consumption model presented in [4] to compute the total energy consumption. The energy needed to send a k bit message by the transmitter is presented in Equation 4. In this formula, λ_{11} is energy/bit used by transmitter circuits, λ_2 is the energy consumed in transmitter amplifier, k is the number of bits in the message and d is the distance that the message traverses in one hop. In our model, d is equal to radio range R.

$$Energy_{tx} = (\lambda_{11} + \lambda_2 d^n)k \tag{4}$$

Equation 5 denotes the energy requirements to receive a k bit message where λ_{12} is energy/bit dissipated by receiver circuits.

$$Energy_{rcv} = \lambda_{12} k \tag{5}$$

The total energy needed to transmit a message on a path with h hops is presented in Equation 6. It is worth mentioning that the receiving energy related to the sink in last hop is not included in this equation, since we assume that the sink nodes are not energy-constrained.

$$P(h) = (h-1)Energy_{rcv} + Energy_{tx}\, h \tag{6}$$

The Equation 7 is achieved by substituting (4) and (5) into (6) as follows.

$$\begin{gathered} P(h) = (h-1)\lambda_{12} k + h(\lambda_{11} + \lambda_2 d^n)k \\ P(h) = k[(h-1)\lambda_{12} + h(\lambda_{11} + \lambda_2 d^n)] \end{gathered} \tag{7}$$

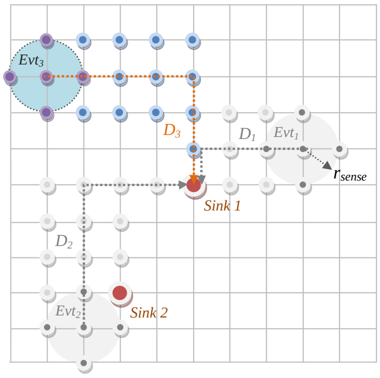


Fig. 2. Past events (Evt₁ & Evt₂) and new event (Evt₃) accompany with sensors involved in routing.

The *average* hop-count $\bar{h}$ and the number of source nodes N_{src} that are sensing an event Evt_i are directly proportional with the total energy needed to report that event to the sink. This amount of energy is calculated in Equation 8 where $\text{Energy}_{\text{hop}}$ is the energy required to emission a message in one hop at distance *d* [7].

$$Energy_{total}^{i} = N_{src}\, Energy_{hop}\, \bar{h} \tag{8}$$

As it mentioned before, all sensor nodes which are in vicinity of the event point and located within the circle $C_{Evt_i,R_{sense}}$ become active. The number of source nodes involved in an event (N_{src}) in our model is achieved by Equation 9 where ρ is the density of sensor nodes which are distributed all over the field.

$$N_{src} = \pi(r_{sense})^2\rho \tag{9}$$

We can estimate the average number of hops through Equation 10 (justification is presented in [7]).

$$\bar{h} \approx \max\left(1, \frac{D}{h}, \frac{D}{h} - 0.5\right), \tag{10}$$

Now, assuming that the total energy consumption on a desired path could also be defined by Equation 11 as follows:

$$P(h) = Energy_{hop}\, \bar{h} \tag{11}$$

Thus, $Energy_{total}^{i}$ is approximated by substituting (11) and (9) into (8) as seen in Equation 12.

$$Energy_{total}^{i} = N_{src}\, P(h) \tag{12}$$

The expansion of (12) could be well expressed by Equation 13.

$$Energy_{total}^{i} \approx \begin{cases} S_1: \pi(r_{sense})^2\rho k(\lambda_{11} + \lambda_2 d^n), \\ \qquad\qquad\qquad\qquad \text{if: } x_{Evt_i} = x_s \text{ and } y_{Evt_i} = y_s \\ S_2: \pi(r_{sense})^2\rho k\left[\left(\frac{D}{h} - 1\right)\lambda_{12} + \frac{D}{h}(\lambda_{11} + \lambda_2 d^n)\right], \\ \qquad\qquad\qquad\qquad \text{if: } x_{Evt_i} \neq x_s \text{ and } y_{Evt_i} \neq y_s \\ S_3: \pi(r_{sense})^2\rho k\left[\left(\frac{D}{h} - 1.5\right)\lambda_{12} + \left(\frac{D}{h} - 0.5\right)(\lambda_{11} + \lambda_2 d^n)\right], \\ \qquad\qquad\qquad\qquad \text{if: } x_{Evt_i} = x_s\ or\ y_{Evt_i} = y_s \end{cases} \tag{13}$$

In fact, there are *I* serial events during the network lifetime instead of one. Therefore, the total energy consumption of the whole sensor field is given by Equation 14 where $Energy_{total}^{i}$ is the energy needed to report Evt_i to the sink node, and is presented in (13).

$$Energy_{total}^{SF} = \sum_{i=1}^{I} Energy_{total}^{i} \tag{14}$$

4. The EEDARS Algorithm

In this section, we are going to give the protocol details. First of all, network initialization and network functionality as two main procedures will be investigated. Then, we proceed to explain role switching scheme in which the nearest sink moves toward the event point to collect data and the other one goes straight to the center of field to facilitate sink localization for next event. Finally, a Joint Routing Strategy and Sink Mobility will be presented to maximize network lifetime.

4.1. An Overview

The EEDARS algorithm includes two major functions in which the instructions are organized as follows.

4.1.1. Network Initialization

At the time of deployment, one of the sinks is located on the center of network and another one moves within the sensor field and stops at a random position. The attribute *state* shows that which one is static or mobile at the time. Since a geographic routing applied in this protocol, all nodes should be aware about their own position as well as the position of their neighbors and also sink node. Thus, the static sink propagates its position plus the location of mobile sink once at the time of network initialization. During this stage, each sensor will be aware of its neighbors' position and consequently is able to make its neighbor table.

4.1.2. Network Functionality

When the network elements and parameters are initialized successfully, it goes to the waiting mode until an event occurs. At this time, the sensor nodes within the event region start to send the first packets towards the central sink with static mode. During this time, all source nodes buffer the rest packets and waits for the sink acknowledgment. Since the sinks can communicate with each other directly, they can share some information about previous events. By the help of this information, the sink with static mode which receives the first packets calculates the average communication time $\bar{T}_{tx}^{Evt}$ for previous events reported to the sinks. It's an estimated time needed for sink to come back to the center of field during the period of data gathering by the other sink at next event. As seen in Equation 15, the maximum movement threshold $R_{threshold}$ could be computed by the use of the time $\bar{T}_{tx}^{Evt}$ and the maximum speed of the sink node V_{max}.

$$R_{threshold} = V_{max}\bar{T}_{tx}^{Evt} \tag{15}$$

It is worth noting that the number of hops the mobile sink goes towards the center of event depends on two factors: firstly, the number of packets a source node produce in each round. On the other hand, the time needed for source node to send all packets related to that event. Secondly, the maximum speed of sink nodes. A tradeoff is needed between these factors.

The shaded area in **Fig. 3** indicates the boundary in which the nearest sink to the event can move towards the source nodes. It cannot exit from this boundary in order to guarantee the return time to the center. Now, the central sink calls the sink-switching procedure with $R_{threshold}$ as input argument to redefine the role (state) of dual sinks in the face of the current event.

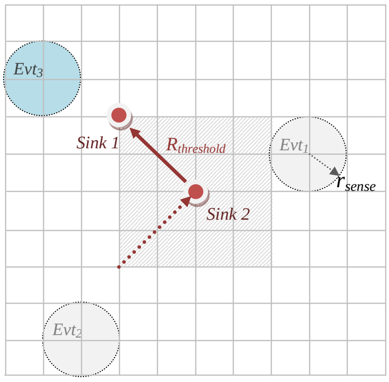


Fig. 3. $R_{threshold}$ as maximum range for sink movement from the center of field based on the sink speed.

4.2. The Role Switching Scheme

As illustrated in **Algorithm 1**, at the beginning of *sink-switching* phase, the static sink calculates the Euclidian distance from event point to both sink1 and sink2 by Equation 3 (line

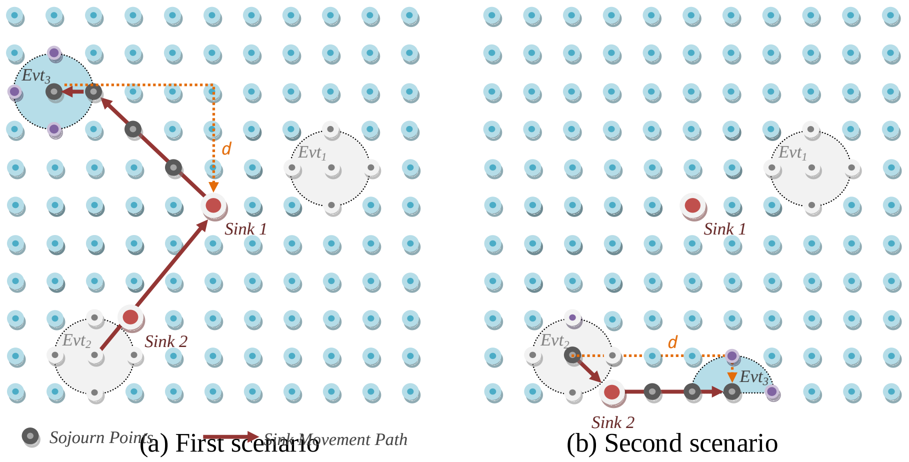
```
Algorithm 1:
1:  procedure sink-switching ($R_{threshold}$) begin
2:      The sink with static state calculates both $E_i^{sink1}$ and $E_i^{sink2}$;
3:      if ($E_i^{sink1} < E_i^{sink2}$) then
4:          if (sink2→position <> ($x_S^C$, $y_S^C$)) then
5:              Sink2→state = static;
6:              Move sink2 to ($x_S^C$, $y_S^C$) to become static sink;
7:          end if
8:          Sink1→state = mobile;
9:          do
10:             Procedure movement-pattern (sink1);
11:             Mobile sink collects the next packets;
12:         while (event exists);
13:     else
14:         if (sink1→position <> ($x_S^C$, $y_S^C$)) then
15:             Sink1→state = static;
16:             Move sink1 to ($x_S^C$, $y_S^C$) to become static sink;
17:         end if
18:         Sink2→state = mobile;
19:         do
20:             procedure movement-pattern (sink2);
21:             Mobile sink collects the next packets;
22:         while (event exists);
23:     end if
24: return ();
25: end proc
```

Algorithm 1. Pseudo-code of the role switching algorithm used by double sinks.

2). Whichever has the lowest distance will be chosen to collect data from the source nodes of relevant event. In **Algorithm 1**, lines 3-13 express the situation in which the sink1 is closer to the event point than the sink2. As said before, each sink has an attribute called state. Thus, the sink1 initializes its state to mobile and moves towards the event region step by step through executing the movement-pattern procedure (discussed later in 4.3). Simultaneously, the sink2's state becomes *static*. Then, it returns to the field center $(x_S^C,\ y_S^C)$, if it is not in this position. It is worth noting that the sink2 can receive the first packets of *next event* without any sink localization overhead when it arrives to the point $(x_S^C,\ y_S^C)$. The lines 14-22 show the similar code when the sink2 is closer to the event point. Thus, it is the candidate for data gathering while the sink1 moves to the field center to facilitate the sink localization. It is the power point of the EEDARS algorithm. **Fig. 4** presents two scenarios for the EEDARS protocol. In **Fig. 4(a)**, the static sink1 which is closer to the new event Evt_3 leaves its position to the center of event region. At the same time, the sink2 that already finished data gathering from previous event Evt_2 is coming back to the center of field to fill the empty place of sink1. In this way, they switch their duties with each other for the next round. In second scenario which shown in **Fig. 4(b)**, the sink2 that recently finished data collection from Evt_2 is closer to the new event Evt_3. Therefore, it will be chosen to continue its duty which is data gathering at the next round by getting close to the event region (Evt_3). Meanwhile, the sink1 still remains at its position among the sensor network.

4.3. Joint Routing Strategy and Sink Mobility

By calling movement-pattern procedure for the sink with *mobile* state, it goes one step forward toward the event point. If the mobile sink is not located on the center of event region and its distance to the center of field is less than $R_{threshold}$ (the sink is inside the shaded area in **Fig. 3**), it could move towards the center of event region. Now, if both x and y coordinates of mobile sink is contradict the position of event point, the sink goes one step diagonally towards the event region. Otherwise, it should move one hop vertically or horizontally. Diagonal movement could reduce the length of path as much as two hops at one stage and consequently save the energy significantly. However, by each vertical or horizontal step the sink gets close to the event point as much as one hop. The sink should stop at sojourn places located on grid



(a) First scenario (b) Second scenario

Fig. 4. Two scenarios for role switching mechanism in EEDARS. a) Sink1 and sink2 switch their duties to each other at next round. b) Sink1 and sink2 continue their duties at next round.

cross points at each stage. In this state, it could communicate with source nodes without any packet loss caused by sink mobility.

As it expressed before in Equation 12, the total energy consumption $Energy^{i}_{total}$ for an event with N_{src} number of sources is equal to $N_{src}\ P(h)$. Therefore, the optimal state for the EEDARS algorithm is well illustrated by Equation 16 where E^{src}_{tx} is the sum of energy needed for the transmitter of source nodes to disseminate their data.

$$\lim_{h \to 1} N_{src}\ P(h) = E^{src}_{tx} \tag{16}$$

This Equation shows that whenever the sink arrives at the center of event region with one-hop distance from all the source nodes, the total energy consumption for that event is equal to the sum of energy used by the sources' transmitter for only sending the data along one hop.

5. SIMULATION RESULTS

We used NS2 (ver. 2.33) framework to simulate all proposed mobile sink strategies. We arranged the sensor nodes in a grid structure in which the number of sensors varies from 7×7 to 23×23 with the increment steps of 4. The field is a square area of size L=2000 m for each edge. Both the maximum communication range R and the distance between two adjacent nodes d were considered 75 m. Therefore, each node can only communicate with its 4 neighbors in north, south, west and east. The sensing range of each sensor R_{sense} was also fixed to 75 m. Since the center of circular event area is assumed to locate on one of the grid cross points, maximum 5 sensors are considered as source nodes for each event (as seen in **Fig. 1**). All nodes were loaded with 0.25 Joule of energy at the beginning of simulation. We employed the energy model presented in [26] for transmitter and receiver power consumption. These parameters are shown in **Table 1** completely.

The results are shown with 95% confidence intervals, since they are averaged over 100 simulations runs. If a node has lost 99% of its initial energy, it is considered "died". When an event occurs, the source nodes at the event region generate and send 1 packet every 6 seconds. We consider two *scenarios* for number of packets reported by source nodes. In the first scenario, each source overall sends 10 packets [23] for an individual event. In the second one, we increase the packets up to 30. The period of 6 seconds is the time needed for the sink to relocate from a sensor node to one of its neighbors in 8 directions. In this state, the sink should move with the maximum speed of 20 m/s. In this simulation, the MAC layer is IEEE 802.11. However, we adjust the time between two consecutive packets in such a way that there is no collision in the network for all methods.

Table 1. Energy parameters for simulation [26]

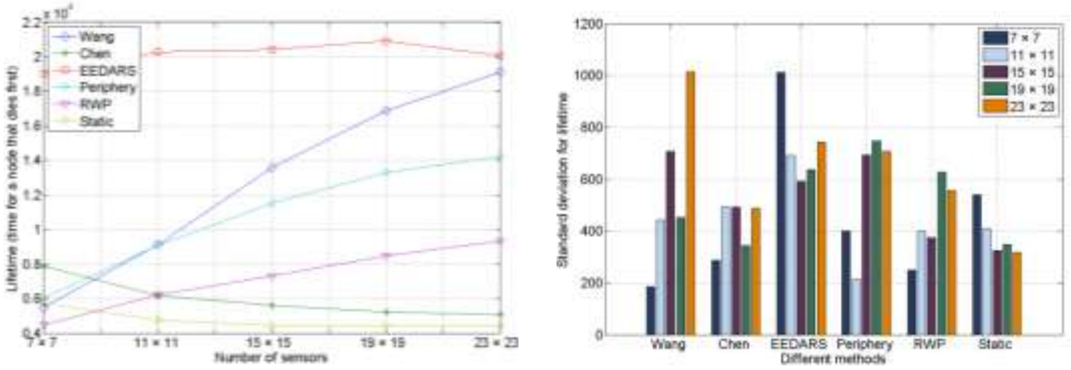
Simulation parameters	Value
Data packet size (k)	100 bits
$\lambda_{11} = \lambda_{12}$	50 nJ/bit
λ_2	0.0013 pJ/bit/m^4
Threshold distance (d)	75 m
Propagation loss	1/d^2

5.1. Mobile Sink Strategies

We compared our method with five other approaches to show the quantity of its performance increase. The first one, called *Wang* [23] (according to the name of author), proposes to employ two mobile sinks in the network. They move on a predefined diamond-shaped path in the opposite side of each other. We named the second approach *Chen* [12], based on the name of first author. One of the sinks in this method is static permanently at the center of the field while the mobile sink collects a portion of information reported by events to decrease the traffic load on the static one. The third one, called *Periphery* [17], supposes that the sink moves on the boundary of the network. However, we adapted this strategy for square-shaped network. *RWP* (random way point) [7] is the forth strategy in which, the sink follows a stochastic mobility pattern. Finally, *Static* presents a network with a static sink at the center of field.

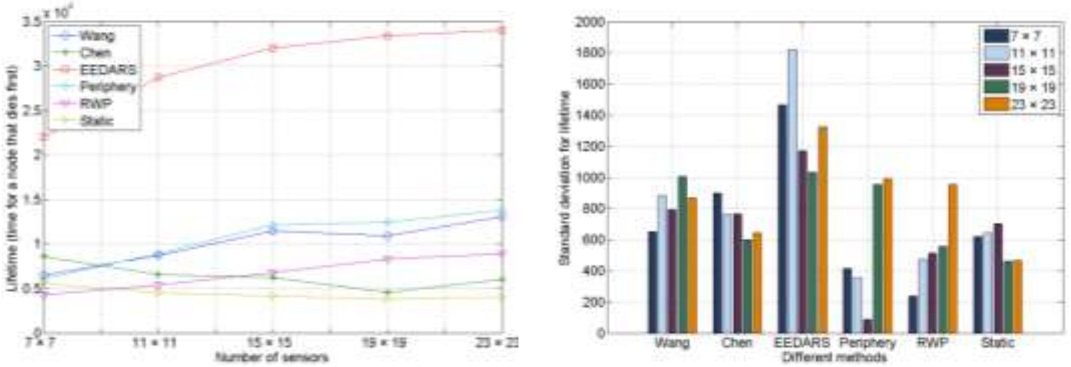
5.2. Network Lifetime

The network lifetime is defined as the time in which the first node will be died [26]. **Fig. 5(a)** and **Fig. 5(b)** respectively illustrate the average lifetime of the network and related standard deviation value for all six strategies when the source nodes report 10 packets per event. As seen in **Fig. 5(a)**, our proposed algorithm, EEDARS, has the highest average lifetime with low fluctuation. Although the Wang algorithm has an upward trend, its standard deviation for the maximum size of the field (column 23×23 in **Fig. 5(b)**) is the largest value among all methods. Thus, it is not so stable against the changes in network size. **Fig. 6** shows the results of second scenario when the source nodes produce 30 packets per event. This time the average lifetime of our protocol in **Fig. 6(a)** has a dramatic difference rather than the other methods. The reason behind this improvement is that when the number of packets reported by source nodes is increased, the sink node has enough time to reach the center of event area. In this state, it can collect a remarkable number of packets only in one hop. Since the average lifetime of EEDARS shows a significant difference, its larger standard deviation (in **Fig. 6(b)**) could be neglected.



(a) Average lifetime (b) Standard deviation for lifetime

Fig. 5. Average lifetime and related standard deviation value (source nodes produce 10 packets).



(a) Average lifetime (b) Standard deviation for lifetime

Fig. 6. Average lifetime and related standard deviation value (source nodes produce 30 packets).

5.3. Residual Energy

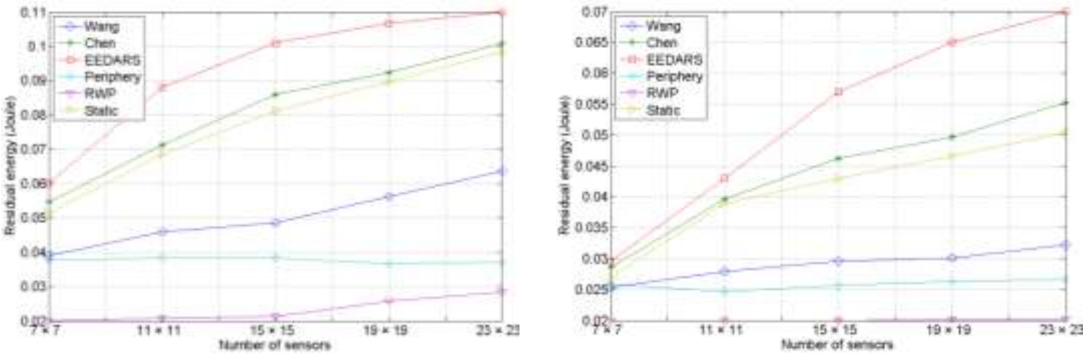
Fig. 7 indicates the residual energy in both scenarios. Almost all protocols show a growing trend when the number of sensor nodes would be increased in the network. However, the EEDARS Algorithm experiences the most residual energy and the RWP scheme has the lowest trend among all strategies. Nevertheless, the average residual energy of the first scenario is higher than the second one.

5.4. Number of Nodes Alive

The number of nodes alive at the end of simulation is represented in **Fig. 8**. A uniform distribution of energy consumption across the network makes our method more energy-efficient so that the number of nodes alive in this protocol is higher than the others. After that, Chen and Static strategies have an acceptable result in both scenarios. Among them, RWP is the worst case.

5.5. End-to-End Delay

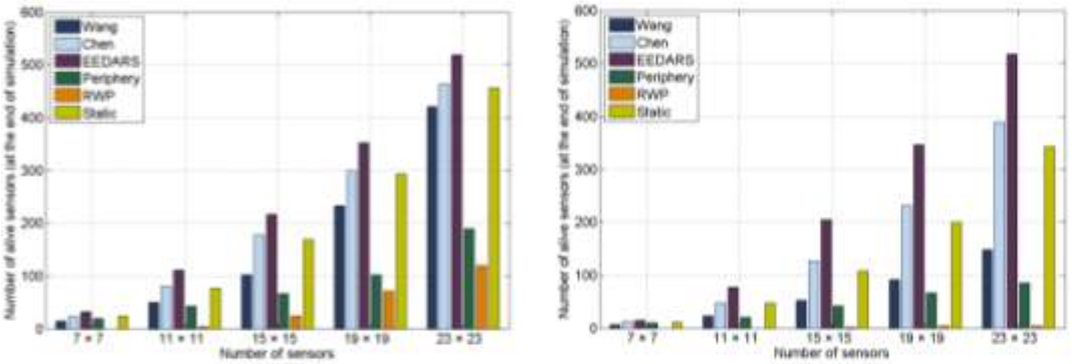
The EEDARS algorithm outperforms all the other strategies from the delay point of view. As seen in **Fig. 9(a)** and **Fig. 10(a)**, in both scenarios, the average delay for all protocols would be increased when the network size is growing. However, in the second scenario as shown in **Fig. 10(a)**, the curve of average end-to-end delay for our proposal has a gentle slope in comparison with the other one. Just like lifetime metric, when the number of event packets would be increased, the sink has the chance to reach the event region and receive the rest of packets through one hop transmission. In this way, the average delay for all packets would be decreased significantly. The Periphery scheme has the highest delay among all strategies. It seems the Wang algorithm which follows the Periphery paradigm has also a large meaningful delay. The philosophy behind this is that when the sink node moves on the boundary of sensor field, it may get far from the events at the opposite side on the network. As a result, the data packets should traverse a long path to reach the sink node. On the other hand, in Static strategy in which the sink node is fixed at the center of network, the packets from all over the field can reach the sink uniformly. It is the reason that the Static mechanism has a moderate delay among the other schemes. **Fig. 9(b)** and **Fig. 10(b)** indicate that the standard deviation of EEDARS is the minimum value among all other strategies in both scenarios.



(a) Source nodes produce 10 packets

(b) Source nodes produce 30 packets

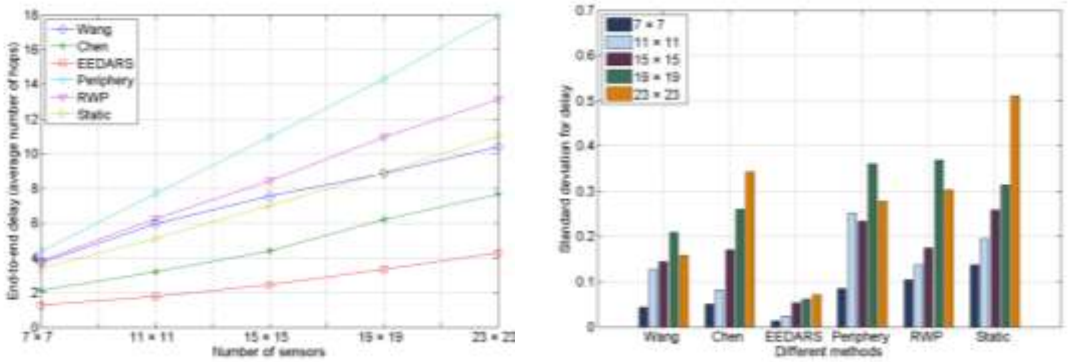
Fig. 7. Average residual energy.



(a) Source nodes produce 10 packets

(b) Source nodes produce 30 packets

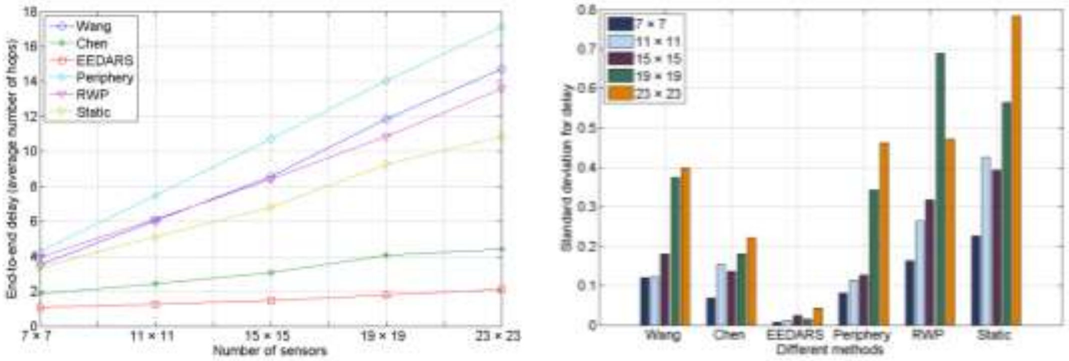
Fig. 8. Average number of nodes alive.



(a) Average delay

(b) Standard deviation for delay

Fig. 9. Average end-to-end delay and related standard deviation value for all six strategies (source nodes produce 10 packets).



(a) Average delay (b) Standard deviation for delay

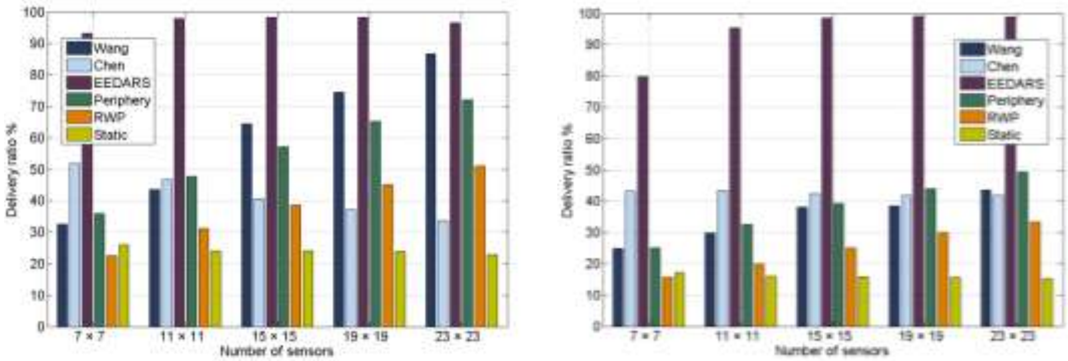
Fig. 10. Average end-to-end and related standard deviation value for all six strategies (source nodes produce 30 packets).

5.6. Delivery Ratio

Delivery ratio is defined as the number of packets successfully reached the sink nodes [27]. In static strategy, when the sink neighbors are forced to relay the data from all over the network to it, they may quickly deplete their energy. In this condition, the sink node may be isolated from the rest of the sensor field, hence unsuccessful packet delivery. It is the main reason that the Static method in both scenarios has the lowest delivery ratio (as seen in **Fig. 11(a)** and **Fig. 11(b)**). In EEDARS algorithm, more than 97% of generated packets in the first scenario could reach the sink. This figure is about 94.2% for the second scenario. When the number of event packets is low (**Fig. 11(a)**), both Wang and Periphery strategies have nearly more than 59% and 41.8% average delivery ratio, respectively. This amount of packet delivery is less than 38% for the second scenario. The reason for this decline is that much more packets will be lost in second scenario, if the event occurs at the vicinity of died sensor nodes.

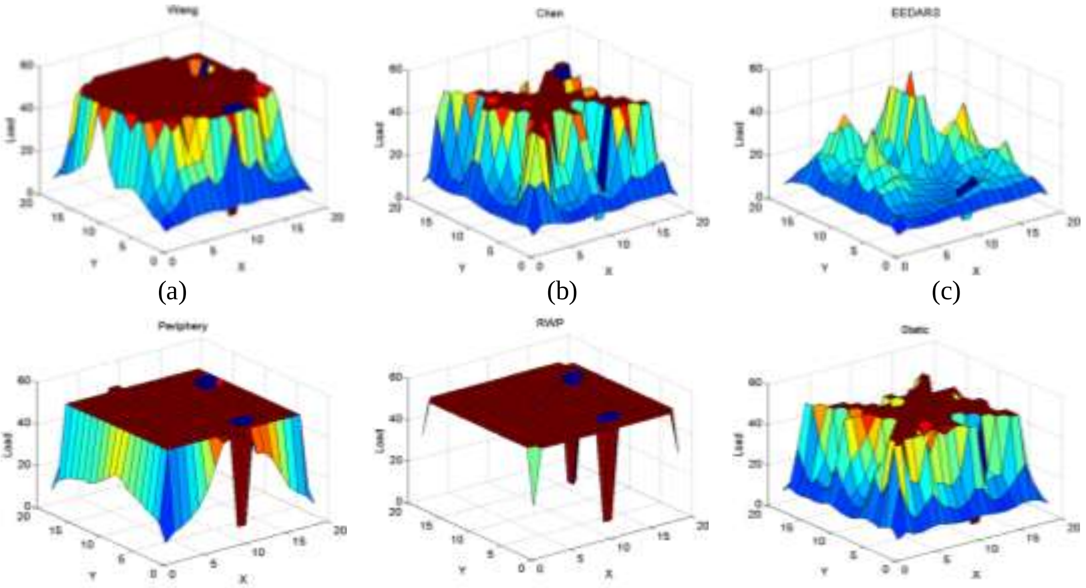
5.7. Network Load

The network load is computed as the total energy consumption divided by the total number of packet reported [23]. **Fig. 12** presents the load of individual sensors in the whole network for all strategies. In this figure, the dark red areas indicate the nodes with high load. It is in contradiction to blue parts. As it seen in **Fig. 12(a)**, the most energy consumption is related to the nodes located in a circle-shape area limited to the boundary of network. It is because of the diamond trajectory of the sink node. For Periphery and RWP models in **Fig. 12(d)** and **Fig. 12(e)** respectively, the power is used uniformly all over the field. However, in Periphery paradigm, the sensors on the boundary of network still keep their energy. The reason is that they mostly engage intermediate nodes to relay their packets to the sink. In Chen (**Fig. 12(b)**) and Static (**Fig. 12(f)**) strategies that have a permanent static sink in the center of field, the energy utilization at the midpoints and also diameters of network is very high. Finally, our proposed algorithm in which the static and mobile sinks switch their duties to each other periodically has the lowest load among all other mobile sink schemes. It could be observed in **Fig.** It could be observed in **Fig. 12(c)**. It is worth noting that the holes in the red surface of **Fig . 12(a)**, **Fig. 12(d)** and **Fig. 12(e)** indicate the void areas.



(a) Source nodes produce 10 packets (b) Source nodes produce 30 packets

Fig. 11. Average delivery ratio.



(a) (b) (c)

(d) (e) (f)

Fig. 12. Network load at the end of simulation for all six strategies.

5.8. Energy Consumption

The energy consumption of sensor nodes in Wang, Chen and EEDARS algorithms as three multi-sink paradigms is shown in Fig. 13. The blue points denote the void areas in the field.

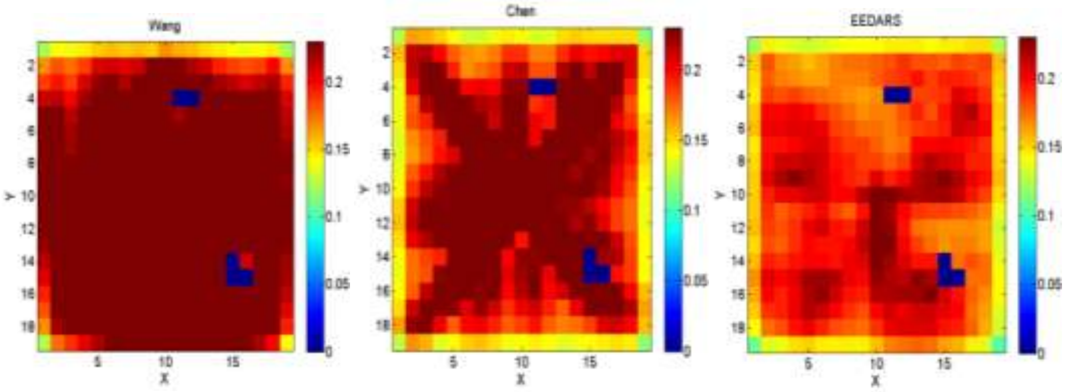


Fig. 13. Energy consumption for three multi-sink strategies (Dark red areas indicate heavy use of energy).

6. Conclusion

In this paper, we developed a dual-sink algorithm, called EEDARS, to fit in the event-driven applications with multi-hop routing for improving the wireless sensor networks lifetime. The protocol is suitable for the scenarios in which all events occur in tandem (non simultaneous events in system). The main contribution of proposed algorithm is that one of the sinks periodically stays at the center of network to eliminate beacon flooding for the sink localization at the next round. At the same time, the other sink collects the data from all sources related to current event through getting close to the event region hop by hop. A distance threshold is applied to the algorithm in order to ensure that the mobile sink has enough time to come back to the center of field if necessary and shift from mobile state to static position before the next round. This role switching mechanism enables the protocol to choose the nearest sink to the event area. In this way, it could shorten the path between the source nodes and the sink, and hence conserve much more energy. The simulation results with random event occurrence are compared to 5 other approaches to show the efficiency of proposed algorithm. We consider two scenarios for all experiments as follows: at the first one, 10 packets are reported by each source node while in the second one, this number is increased up to 30 packets. In average, the results indicate that EEDARS prolongs the network lifetime in the scale of 119.75% and 305.6% with 95% confidence intervals rather than two other multiple mobile sink mechanisms, Wang and Chen, respectively. This rate of lifetime improvement is 125.3% for Periphery pattern, 251.5% for random (RWP) strategy, and 467% for Static sink method.

References

[1] J.N. Al-Karaki and A.E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6-27, Dec. 2004. Article (CrossRef Link).

[2] M. Eslaminejad and S. A. Razak, “Fundamental Lifetime Mechanisms in Routing Protocols for Wireless Sensor Networks: A Survey and Open Issues,” *Sensors*, vol. 12, no. 10, pp. 13508-13544, Oct. 2012. Article (CrossRef Link).

[3] W. Youssef and M. Younis, "A Cognitive Approach for Gateway Relocation in Wireless Sensor

Networks," in *Proc. of the Nineteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2006)*, pp. 484-489, May 11-13, 2006.

[4] K. Akkaya, M. Younis and M. Bangad, "Sink Repositioning for Enhanced Performance in Wireless Sensor Networks," *Computer Networks*, vol. 49, no. 4, pp. 512-534, Nov. 2005. Article (CrossRef Link).

[5] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, May 2005. Article (CrossRef Link).

[6] M. R. Eslaminejad, M. Sookhak, S. A. Razak and M. Haghparast, "A Review of Routing Mechanisms in Wireless Sensor Networks," *International Journal of Computer Science and Telecommunications*, vol. 2, no. 7, pp. 1-9, Oct 2011.

[7] Z. Vincze, D. Vass, R. Vida, A. Vidács and A. Telcs, "Adaptive Sink Mobility in Event-Driven Densely Deployed Wireless Sensor Networks," *Ad Hoc & Sensor Wireless Networks*, vol. 3, no, 2-3, pp. 285–315, 2007.

[8] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of the 33rd Annual Hawaii International Conference on System Sciences*, pp. 223-233, Jan 2000. Article (CrossRef Link).

[9] S. Gandham, M. Dawande, R. Prakash and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," in *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 377-381, Dec 2003. Article (CrossRef Link).

[10] S. Olariu and I. Stojmenovic, "Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting," in *Proc. of the 25th IEEE International Conference on Computer Communications*, pp. 1-12, Apr 2006. Article (CrossRef Link).

[11] K. Akkaya, M. Younis and W. Youssef, "Positioning of base stations in wireless sensor networks," *IEEE Communications Magazine*, vol. 45, no. 4, pp. 96-102, April, 2007. Article (CrossRef Link).

[12] J. Chen, M. B. Salim and M. Matsumoto, "Modeling the Energy Performance of Event-Driven Wireless Sensor Network by Using Static Sink and Mobile Sink," *Sensors*, vol. 10, no. 12, pp. 10876-10895, Dec 2010. Article (CrossRef Link).

[13] X. Wu and G. Chen, "Dual-Sink: Using Mobile and Static Sinks for Lifetime Improvement in Wireless Sensor Networks," in *Proc. of the 16th International Conference on Computer Communications and Networks*, pp. 1297-1302, Aug 2007. Article (CrossRef Link).

[14] W. Youssef and M. Younis, "Intelligent gateways placement for reduced data latency in wireless sensor networks," in *Proc. of the 2007 IEEE International Conference on Communications (ICC'07)*, pp. 3805-3814, Jun 2007. Article (CrossRef Link).

[15] E. Gürses, Ö. B. Akan, "Multimedia Communication in Wireless Sensor Networks," *Annals of Telecommunications, Springer*, vol. 60, no. 7-8, pp. 872-900, 2005. Article (CrossRef Link).

[16] I. Chatzigiannakis, A. Kinalis and S. Nikoletseas, "Sink mobility protocols for data collection in wireless sensor networks," in *Proc. of the international Workshop on Mobility Management and Wireless Access*, pp. 52-59, Oct 2006. Article (CrossRef Link).

[17] J. L. Jean and P. Hubaux, "Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks," in *Proc. of the IEEE INFOCOM 2005. The Conference on Computer Communications - 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1735-1746, Mar 2005. Article (CrossRef Link).

[18] B. Nazir and H. Hasbullah, "Mobile Sink based Routing Protocol (MSRP) for Prolonging Network Lifetime in Clustered Wireless Sensor Network," in *Proc. of the 2010 International Conference on Computer Applications and Industrial Electronics*, pp. 624-629, Dec 2010. Article (CrossRef

Link).

[19] T. Chang-geng, X. Ke, W. Jian-xin and C. Song-qiao, "A sink moving scheme based on local residual energy of nodes in wireless sensor networks," *Journal of Central South University of Technology*, *Springer*, vol. 16, no. 2, pp. 265-268, 2009. Article (CrossRef Link).

[20] P.D. Hossein Zadeh, C. Schlegel and M.H. MacGregor, "Distributed optimal dynamic base station positioning in wireless sensor networks," *Computer Networks*, vol. 56, no. 1, pp. 34-49, Jan 2012. Article (CrossRef Link).

[21] S. Basagni, A. Carosi, C. Petrioli and C.A. Phillips, "Heuristics for Lifetime Maximization in Wireless Sensor Networks with Multiple Mobile Sinks," in *Proc. of the IEEE International Conference on Communications*, pp. 1-6, Jun 2009. Article (CrossRef Link).

[22] H. Sabbineni and K. Chakrabarty, "Datacollection in Event-Driven Wireless Sensor Networks with Mobile Sinks," *International Journal of Distributed Sensor Networks*, vol. 2010, no. 2010, pp. 1-12, Jul 2010. Article (CrossRef Link).

[23] B. Wang, D. Xie, C. Chen, J. Ma and S. Cheng, "Deploying Multiple Mobile Sinks in Event-Driven WSNs," in *Proc. of the IEEE International Conference on Communications*, pp. 2293-2297, May 19-23, 2008. Article (CrossRef Link).

[24] X. Wu, J. Cho, B. J. d'Auriol and S. Lee, "Mobility-Assisted Relocation for Self-deployment in Wireless Sensor Networks," *IEICE Trans. on Communications*, vol. E90-B, no.8, pp.2056-2069, Aug, 2007. Article (CrossRef Link)

[25] S. Capkun, M. Hamdi and J. Hubaux, "GPS-free Positioning in Mobile Ad-hoc Networks," *Cluster Computing*, *Springer*, vol. 5, no. 2, pp. 157-167, 2002. Article (CrossRef Link).

[26] E. M. Saad, M. H. Awadalla, and R. R. Darwish, "Adaptive Energy-Aware Gathering Strategy for Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 6, pp. 834-849, Nov 2009. Article (CrossRef Link).

[27] L. Guo, R. Beyah and Y. Li, "SMITE: A Stochastic Compressive Data Collection Protocol for Mobile Wireless Sensor Networks," in *Proc. of the INFOCOM, 2011 Proceedings IEEE*, pp. 1611-1619, Apr 2011. Article (CrossRef Link).

Mohammadreza Eslaminejad received Bachelor of Science in Software Engineering from Payame Noor University in 2001, and Master of Science at the same field in 2006 from the Islamic Azad University, Arak, Iran. He is pursuing his Ph.D. programme in the Faculty of Computer Science and Information System, Universiti Teknologi Malaysia (UTM). He worked as an Instructor in the Department of Computer Science & Information Technology at Islamic Azad University and simultaneously in the Mathematical Department at Payame Nour University from 2004 to 2009. He also was the Manager of Computer Group at Neyriz Islamic Azad University (Fars, Iran) during 2004 to 2008. His areas of interest include File Sharing Services in Wireless Mobile Peer to Peer Networks and Energy-Efficient Routing and Security issues in wireless sensor networks. He is a student member of IEEE.

Shukor Abd Razak is a senior lecturer at the Universiti Teknologi Malaysia (UTM). His research interests are on the security issues of mobile ad hoc networks, mobile IPv6 networks, vehicular ad hoc networks, and network security. He is the author and co-author of many journal and conference papers at national and international levels.

Abdul Samad Ismail received the B.S. degree in Math and Computer Science from University of Wisconsin Superior, Wisconsin, USA in 1985, and M.S. degree in Computer Science from Central Michigan University, Mt. Pleasant, Michigan, USA in 1988, and PhD degree in Collaborative Environments from University of Wales Swansea, Swansea, UK in 2002. He is currently professor in Department of Computer Systems and Communication (CSC), and Director of Quality in Office of Chancellery at Universiti teknologi malaysia. His research interests are computer networking and distributed computing.