

## An efficient routing algorithm to preserve $k$ -coverage in wireless sensor networks

Ali Ahmadi · Mohammad Shojafar ·  
Seyede Fatemeh Hajeforosh · Mehdi Dehghan ·  
Mukesh Singhal

Published online: 5 December 2013  
© Springer Science+Business Media New York 2013

**Abstract** One of the major challenges in the area of wireless sensor networks is simultaneously reducing energy consumption and increasing network lifetime. Efficient routing algorithms have received considerable attention in previous studies for achieving the required efficiency, but these methods do not pay close attention to coverage, which is one of the most important Quality of Service parameters in wireless sensor networks. Suitable route selection for transferring information received from the environment to the sink plays crucial role in the network lifetime. The proposed method tries to select an efficient route for transferring the information. This paper reviews efficient routing algorithms for preserving  $k$ -coverage in a sensor network and then proposes an effective technique for preserving  $k$ -coverage and the reliability

---

A. Ahmadi  
Electrical and Computer Department, Islamic Azad University,  
Qazvin Branch, Qazvin, Iran

M. Shojafar (✉)  
Department of Information Engineering, Electronic and Telecommunication (DIET),  
“Sapienza” University of Rome, via Eudossiana 18, 00184 Rome, Italy  
e-mail: m.shojafar@yahoo.com; shojafar@diet.uniroma1.it

S. F. Hajeforosh  
Electrical Engineering Department, Mazandaran University of Science  
and Technology, Babol, Iran

M. Dehghan  
Computer Engineering Department, Amirkabir University of Technology,  
Tehran, Iran  
e-mail: dehghan@aut.ac.ir

M. Singhal  
Computer Science and Engineering, University of California,  
Merced, USA  
e-mail: msinghal@ucmerced.edu

of data with logical fault tolerance. It is assumed that the network nodes are aware of their residual energy and that of their neighbors. Sensors are first categorized into two groups, coverage and communicative nodes, and some are then re-categorized as clustering and dynamic nodes. Simulation results show that the proposed method provides greater efficiency energy consumption.

**Keywords** Wireless sensor network · Localization · Energy-efficient routing · Energy saving ·  $k$ -coverage · Euclidean distance

## 1 Introduction

There has been a recent progression in micro-electro-mechanical systems of smart sensors, wireless communications platforms, and digital electronics followed by a plausible construction of tiny sensor nodes, which have low energy consumption and are cost-effective sensor nodes that are capable of connecting as wireless networks [1, 2]. These nodes contain the main elements of wireless sensor networks (WSNs), and are capable of sensing and measuring environmental factors in the vicinity, such as temperature, pressure, voice, and different types of noise pollution, for data processing and transmission [3–5].

Crucial issues for WSNs include coverage and routing. Several routing methods have been proposed in [6–8]. All of these methods concentrate only on routing regardless of their coverage. Coverage concerns how the nodes can be enabled to cover an area effectively. One of the main goals is to achieve maximal coverage for an area with the least amount of energy consumed by the sensors [9]. Coverage is one of the Quality of Service parameters in networks used for *surveillance* [9]. One of the primary major operations performed by each sensor node is to review and process the node's vicinity to discover incident events that are available in the environment, including the information required for use in application programs [10]. Techniques used to represent coverage retention in sensor networks are based on the quantity and order of coverage (the number of nodes providing coverage for an area), combined with energy consumption [11]. Interestingly, these techniques play a crucial role because increasing the order of coverage and supporting an order increment to  $k$ -order is crucial in WSNs for attaining reliable information about their environment [12, 13]. It is necessary to provide sensor node coverage for some applications, such as target tracking. Thus, improved coverage and support of  $k$ -coverage, which means that each point of an object region has coverage from at least  $k$  sensor nodes, is a prerequisite for applied WSNs [13–17].

If all of the network targets are diagnosed with coverage sensors, but all of the gathered information do not send to the sink (is the node that has unlimited energy) (i.e., the nodes do not connect with multi-hop to the sink) *connective hole* will exist. It means, Target coverage in network environment is the most important tasks of the sensor nodes for receiving the information of the target status. But, target coverage is not enough and the links between the sink and sensors should be established for transmitting information that keep the network efficiency in the network lifetime. [18–21]. This is the primary challenge that should be studied in protocol designs for networks,

since management of energy consumption aims for optimal utilization of a battery, while the relevant data should be made available from the receiving environment to the data processing performed in an optimized wireless sensor network with minimum time consumed [11, 22]. Routing algorithms are among the main elements for designing sensor networks, and retention of network coverage is one of the critical issues in the routing protocols [23–26].

In this paper, we present a method for transferring sensor information in which  $k$ -order region coverage is maintained: the method reduces communication expenses and results in a longer network lifetime and greater reliability. The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 presents a system model. Section 4 presents our proposed method, and Sect. 5 presents our simulation results. Finally, Sect. 6 discusses our conclusions and future research directions.

## 2 Related work

This section reviews the methods that are related to coverage in a WSN. One of the most important issues in the area of sensor networks understands the conditions of area coverage (AC) by sensor nodes. Whenever the Euclidean distance between a relevant target and the localization of a sensor node is less than the coverage radius, it is under the coverage of the sensor node [13, 16, 17, 27]. Different segmentations are used for sensor networks, each one subdividing the area according to the specific protocol. Segmentations can be categorized into three groups according to their target coverage; these are AC, point coverage (PC), and barrier coverage (BC) [28]. The main objective of sensor networks with respect to area coverage is to provide complete coverage and supervision in an environment, while the aim of point coverage is to provide dispersed coverage over a specific distance in the area and barrier coverage aims at eliminating noncoverage in an area. While iterative routing protocols aim to reduce energy consumption as a cost-effective strategy, the issue of coverage is more important in the following protocols [29, 30].

Hybrid Energy-Efficient Distributed Clustering (HEED) [31, 32] is a protocol that periodically selects cluster heads according to a combination of the node's residual energy and a secondary parameter, through constant time iterations. It uses the primary parameter, i.e., residual energy, to select an initial set of cluster heads. Unlike previous protocols that require knowledge of the network density or homogeneity of node dispersion in the field, HEED does not make any assumptions about the network such as density and size. Every node runs HEED individually. At the end of the process, each node either becomes a cluster head or a child of a cluster head.

Coverage-Preserving Cluster Head Selection Algorithm (CPCHSA): CPCHSA [33] is used for sustainable coverage in sensor networks based on the design of the Low Energy Adaptive Clustering Hierarchy (LEACH) [34] algorithm. The technique in CPCHSA changes the function of the LEACH cluster head to coordinate energy consumption along with improving coverage. Thus, the technique uses nodes that are available in less congested areas for cluster heads due to the high rate of cluster head energy consumption that arises from the aggregation of data and the transfer of data to the sink.

The coverage area in the CPCHSA method is based on the estimation of neighboring nodes according to the received signal potential, and the precise coverage neighbors

are not determined. Consequently, the results are merely an estimated range, which is reliable in distribution and high congestion modes. The algorithm functions with random numbers, and it is possible to encounter cluster heads with no further physical distribution in the network.

**Energy-Aware Routing Protocol (EAP):** The EAP protocol represents clusters for the routing algorithm; thus the protocol features cost-effective energy consumption for inter-network or in-network communication to create a balance of energy consumption in the entire resulting network. EAP provides an algorithm for resolving network challenges that uses the coverage available in clusters under the power control of a radio transmitter, so that nodes apply more potential for remote (far) distances, but less for close distances. A cluster head is initially appointed in each period, and EAP's representation algorithm is used for randomly chosen maintenance without a guaranty of network coverage. Whenever sufficient density is found in a distribution of nodes, there is a trend of a convenient balance of coverage [35]. It means, if the nodes density is high in the network, sensing nodes or coverage nodes have been selected with suitable distribution in the network and it is not related to the specific area of the network. Also, in the proposed method, the communicative and coverage nodes are considered fixed, and we try to change the cluster head in time interval to decrease energy consumption and increase network lifetime. Moreover, we use single-hop for transmitting information to the cluster heads.

**Priority-Based Energy-Aware and Coverage Preserving (PBEACP):** The PBEACP [36,37] algorithm is a technique for wireless sensor network routing that is based on a modification of the LEACH method; thus selection of cluster heads is crucial for improving/controlling energy consumption and reinforcing network coverage by convenient energy distribution in the line of a geographical distribution of nodes. Nodes with more neighbors are selected as candidates for cluster head in the PBEACP method, by a calculation of node density in the placement of nodes.

The PBEACP protocol uses the CPCHSA [33] method to distinguish neighboring nodes and the potential of received signals, but it is useless for the precise placement of coverage neighbors and does not provide the required precision for network coverage like CPCHSA does. But, in the proposed method, the direction and angle of the nodes related to each other are determined.

### 3 System model

In this paper, it is assumed that all sensors are fixed and all nodes are aware of their proper placement, targets, and sink. For example, we want to transmit the information (by the help of the heating sensor nodes that we were distributed and sent them by airplane in the military zone) of some gates (that are targets) of the underground military stations that are on the ground, and group of sensors near targets report the heating information of their environments based on their diagnosis radius. Sink nodes are free of any energy constraints and have a fixed placement, and all sensors are aware of this situation. Each sensor has a sensing range radius and two communication ranges that are greater than or equal to the sensing radius, nodes' communicative radiuses are greater than their sensing radiuses, with double the range covered, and all nodes

are synchronized and homogenous. Synchronized means, for instance, we have 100 network sensing nodes, all of them transmit information to each other in 1 s. All the information is transmitted with a Hello message and in one specific period. Homogeneous means, all sensor nodes are organized in one model, architecture and structure.

Number of targets are placed in the environment. The sensors provide the required information for those targets and transfer it to the sink to improve network quality (which here means network age) by retaining coverage and the necessary links through considering the target and sensors positions. Initially, the available sensors are divided into *coverage nodes* that receive and post the environmental information and *communicative nodes* with the task of transferring information received from event registry nodes to the sink by creating a route. The network is then subject to clustering and selection of active nodes (i.e., after making clustering,  $k$ -coverage nodes activate and the rests of coverage nodes in the cluster go to the idle status) in addition to establishing alternative paths from the origin to the sink as long as the network is periodic, to maintain a route to the sink that can be used for information transfers.

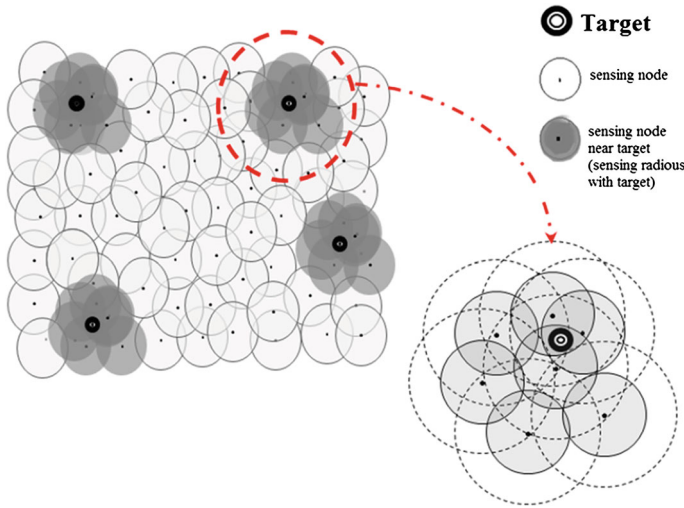
This paper provides a number of sporadic targets in an environment with a linkage between coverage nodes and the sink by retaining  $k$ -coverage, with the coverage nodes of each target clustered in a way that will meet that target. One node is appointed as the head of a cluster of coverage nodes, in addition to the  $k-1$  sensor nodes that serve as the coverage nodes. In each cluster,  $k$  nodes are set in active mode, and the remaining available nodes in that cluster idle until the end of the time period for processing of all nodes of this cluster. In this way, the cluster head provides the sink with relevant information.

The nodes in crucial areas (sensing radius  $R_s$  from each target) that are used for coverage consequently also reduce the burden of nodes in the crucial areas, as nodes stationed outside of those areas transfer the information to the sink. As a result, less energy is consumed by the sensor nodes around each target, and the network lifetime is improved.

Figure 1 depicts the coverage of discrete targets (targets are far from each other) by sensor nodes and the formation of clusters in the vicinity of the targets. This is an example of point coverage of discrete targets.

The geographical position of nodes is calculated by a global positioning system (GPS). The initial structure of each sensor is shown in Table 1. To furnish each target with  $k$ -coverage and the assumed high density of sensors: the greater the value of  $k$ , the more appropriate is the density. Indeed, sensors are unable to change their sensing and communicative radius.

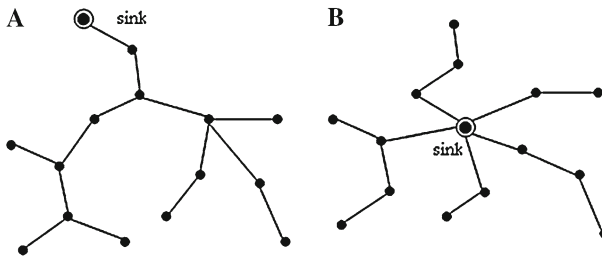
where ID ( $s_i$ ); the node  $s_i$  identity, GPS ( $s_i$ ); its geographical position of node  $s_i$  (while sensors are located and settle down in their locations, identify their positions by GPS and save it in their database), the geographical positions of the sink node [GPS(sink)] (we assume this filed is defined), energy residue [ $E_r(s_i)$ ], current status [Status ( $s_i$ )], and the targets ( $t_0 \dots t_k$ ) are recorded (here, we assume these targets are defined) in the Table 1. The placement of the sink node is crucial; otherwise it might be linked to the network by sensor nodes as depicted in Fig. 2a and cause diminishing node energy and a communicative hole [38]. It is possible to improve the network lifetime and delay time by localizing the sink as in Fig. 2b; this requires placing the sink at a proper distance from the sensors, which reduces the tree height to 3 and raises the root order to 5.



**Fig. 1** An example of point coverage for discrete targets

**Table 1** Initial sensor information

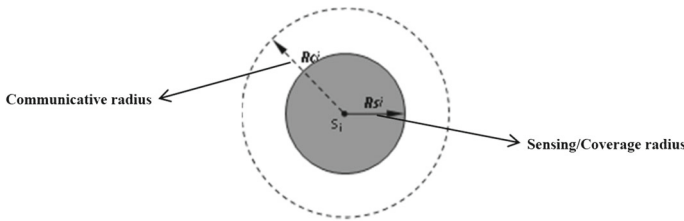
ID ( $s_i$ )
GPS ( $s_i$ )
GPS (sink)
GPS ( $t_0$ )
...
GPS ( $t_k$ )
Status ( $s_i$ )
$E_r(s_i)$



**Fig. 2** **a** Location of sink at an improper distance from sensors. **b** Location of sink at a convenient distance from sensors

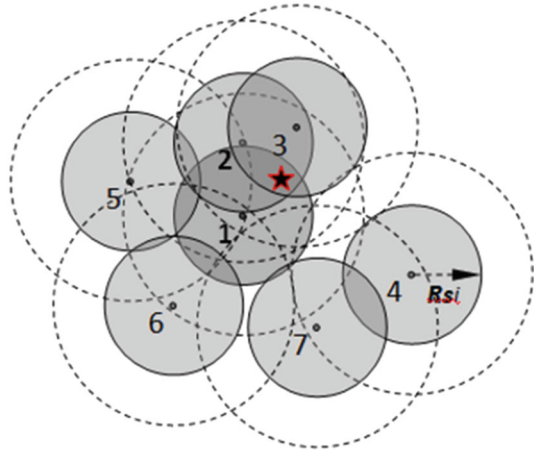
Let the range of sensor node  $s_i$  be the circular area with Coverage radius  $R_{s_i}$  (i.e., this range illustrates the effective range or coverage range of communication with other nodes as shown in Fig. 3, here  $R_s = R_{s_i}$ , for each  $i$ ). The sensor nodes' binary coverage model,  $cov(t_j, s_i)$ , is defined by Eq. (1):

$$cov(t_j, s_i) = \begin{cases} 1 & \text{if } d(t_j, s_i) \leq R_s \\ 0 & \text{otherwise.} \end{cases}, \tag{1}$$



**Fig. 3** Coverage radius  $R_{si}$  and Communicative radius  $R_{ci}$  for node  $i$

**Fig. 4** Example of coverage neighbors set



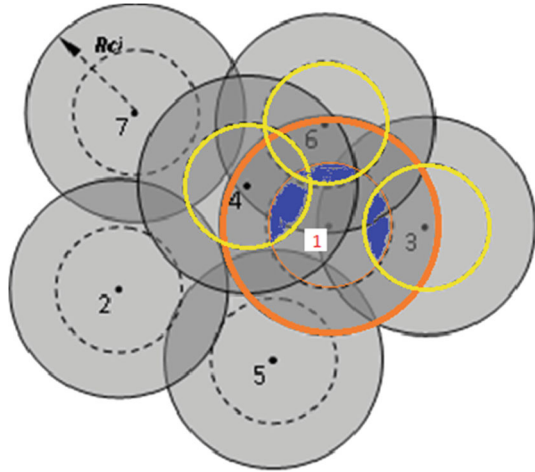
where  $d(t_j, s_i)$  is the Euclidean distance between sensor node  $s_i$  and target  $t_j$ . In the binary detection model (i.e., when detecting the targets), an explicit target is defined in the node coverage area. Each node  $s_i$  establishes a communication range with the threshold distance defined as the *communicative radius*  $R_{ci}$ . Communication shall be established as *single-hop* or *one-hop* whenever the Euclidean distance between two nodes  $s_i$  and  $s_j$  is less than or equal to the communicative radius. Figure 3 depicts the coverage provided by sensor  $s_i$ 's communicative radius besides coverage radius  $R_{si}$ .

*Sensing/coverage neighbors Set of Target  $t_j$* : In Principle, according to the proposed model, each sensor node will identify all targets in its coverage area. Hence, the coverage neighbor set of target  $t_j$  includes all of the sensor nodes that cover/sense the target  $t_j$  in their coverage area, as defined by Eq. (2). In other words, all sensing nodes whose Euclidean distances to target  $t_j$  are less than or equal to  $R_{si}$ . For example, in Fig. 4, sensing neighbors set of target  $t_j$  includes  $\{1, 2, 3\}$  considering the target is denoted as the black star.

$$N_s(t_j) = \{s_i \mid d(t_j, s_i) < R_s\}, \tag{2}$$

*Communicative neighbors set*: The communicative neighbors of sensor  $s_i$  are the set of sensor nodes inbound of node  $s_i$  for the exchange of information, as defined by Eq. (3). In other words, the communicative neighbors of sensor  $s_i$  are all the sensors whose Euclidean distance from the sensing node  $s_i$  is less than communicative radius

**Fig. 5** Example of communicative neighbors set



( $R_{C_i}$  or  $R_c$  for all  $i$ ). A node  $s_i$  is able to interact with its communicative neighbors and transfer/receive information. For example, in Fig. 5, the communicative neighbors set of node 1 is the set  $\{3, 4, 6\}$  [i.e., the internal circle of the node 1 intersected by the internal circle of these nodes: check the blue shaded areas and its intersection by the internal circle of node 1 (communicative radius) of node 3, 4, and 6].

$$N_s(s_i) = \{s_j \mid d(s_i, s_j) < R_c\}, \quad (3)$$

## 4 Proposed method

In this section, we explain our proposed method in detail. Informally speaking, we use the sink's geographical position as the coordinate origin for the Cartesian coordinate vectors (always, we have one sink and its position considered as a origin of our Cartesian coordinate vector). Thus, the target and sensor positions shall be defined by the geographical location of sensors and targets relative to the sink's geographical position. The algorithm consists of three phases:

- Phase I: Collection and processing of primary information.
- Phase II: Creation of coverage clusters for targets and selection of cluster heads.
- Phase III: Selection of active transmitting nodes for information transfer to the sink.

### 4.1 Phase I: collection and processing of primary information

In this phase, the primary required information to be used for the following phases is gathered that consists of sensing status, the initialized energy for each sensor node and the position of each sensor. We assume that as a default, all sensor nodes are in an idle communicative position, so that the initial value of the Status field is 0. Each sensor node's Status field ( $s_i$ ) shall be set in one of the positions specified in Table 2:



**Table 2** Sensor Positions/status

Status ( $s_i$ )	Descriptions
0	Idle communicative
1	Idle coverage
2	Active communicative
3	Active coverage
4	Cluster heads

**Table 3** Target Table for node  $i$  ( $s_i$ )

ID ( $s_i$ )
$D_{\text{sink}}(t_k)$
$d(s_i, t_k)$
GPS ( $t_k$ )

**Table 4** Hello message

ID ( $s_i$ )
GPS ( $s_i$ )
$E_r$ ( $s_i$ )
Status ( $s_i$ )
ID ( $t_k$ )

Each sensor updates its target table by updating the initial sensor information table (Table 1) by processing its target table as follows: if it covers a target, then it transforms from communicative to coverage position and set its status field to 1, otherwise it retains the status value 0. The nodes' with state 1, try to fill their Table 3 by the information they gain from the Table 1. Specifically,  $D_{\text{sink}}(t_k)$  is the Euclidean distance between sink from the  $t_k$  that is calculated and added to the Table 3 of the node  $i$ , also,  $d(s_i, t_k)$  is Euclidean distance between node  $i$  or  $s_i$  from the  $t_k$  that is calculated and added. Last two fields come directly from Table 1. In the records of Table 1, the nodes whose Euclidean distances from the  $t_k$  are less than or equal to  $R_c$  are inserted into the Table 3 (e.g., if each node in its coverage radius has  $t_k$ , it is added into the Table 3 as  $s_i$  and all other parameters are calculated for this record). The coverage target table is depicted in Table 3. This table includes the coverage target's distance to the sink denoted as  $D_{\text{sink}}(t_k)$ , the distance of the sensor to the target denoted as  $d(s_i, t_k)$ , the target's position [GPS ( $t_k$ )], and the target's identity or ID ( $s_i$ ).

Each sensor node then transmits its identity [ID ( $s_i$ )], coordinates [GPS ( $s_i$ )], residual energy [ $E_r$  ( $s_i$ )], current status [Status ( $s_i$ )], and its coverage target identity [ID ( $t_k$ )] to single-hop neighbors with a hello message. Whenever a sensor fails to cover a target, the ID ( $t_k$ ) will be null. The hello message format is depicted in Table 4.

On the top of the hello message a sensor node updates its neighbors table. The format of sensor  $s_i$ 's neighbors table, with  $s_j, j = 1 \dots m$  (the number of node  $s_i$ 's neighbors), is depicted in Table 5.

where sensor  $s_j$ 's neighbors table consists of  $s_j$ 's identity [ID ( $s_j$ )],  $s_j$ 's coordinates [GPS ( $s_j$ )],  $s_j$ 's residue energy [ $E_r(s_j)$ ],  $s_j$ 's sensor status [Status ( $s_j$ )], the identity of

**Table 5** Sensor  $s_i$  neighbors table (node  $s_j$  is a neighbor)

ID ( $s_j$ )
GPS ( $s_j$ )
$E_r$ ( $s_j$ )
Status ( $s_j$ )
ID ( $t_k$ )
$d$ ( $s_j, s_i$ )
$D_{\text{sink}}$ ( $s_j$ )

the target covered by  $s_j$  [ $ID(t_k)$ ], the distance to sensor  $s_i$  [ $d(s_j, s_i)$ ], and the distance to  $s_j$ 's sink [ $D_{\text{sink}}(s_j)$ ]. At the end of this phase, each node is aware of its neighbors' details to apply the required information in the following phases.

In a nutshell, each sensor node contains three tables: initial sensor information Table (Table 1), Target Table (Table 3) and Neighbors Table (Table 5). Table 3 is established based on the position/status of the current node and the targets positions.

Algorithm 1 presents pseudocode for Phase I.

**Algorithm 1.** Pseudocode for Phase I (First\_node\_setup ( $s_i$ ))

```

Input:  $s_i$  list
Output: First_node_setup ( $s_i$ )
1 : Begin
2 . Repeat // continue until the targets are available the target table (Table 3)
3 . Compute destination  $s_i, t_k$  with  $d(s_i, t_k) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ ;
4 . If ( $d(s_i, t_k) < \text{sensing range}$ ) then  $s_i.\text{status} = 1$ ;
5 :  $s_i$  broadcasts hello message to its 1-hop(single-hop) neighbors , id, GPS,  $E_r$ , Status,  $ID(t_k)$ ;
6 : If (Hello message from neighbors sensor is received)
7 :   Begin
8 :     Update neighboring set ( $s_i$ );
9 :     Compute destination between  $s_i$  &  $s_j$  as  $d(s_i, s_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ ;
10 :    Compute destination between sink &  $s_j$  as  $D_{\text{sink}}(s_j) = \sqrt{(x_{\text{sink}} - x_j)^2 + (y_{\text{sink}} - y_j)^2}$ ;
11 :   End If
12 : Until ( $t_k$  is available)
13 : Return
14 : End
    
```

\* Here, we calculate the time needs for each target, which is defined as time taken to find the positions (Table 1) + time for updating the targets table (Table 3) + time taken to send Hello message to the sensor nodes neighbors (one-hop) (Table 4)

#### 4.2 Phase II: creation of coverage clusters for targets and selection of cluster heads

In this phase, a cluster is initially created for each target. Members of the cluster are nodes located within the maximum distance  $R_c$  from the target whose coverage includes the target. Then the cluster head is nominated as follows.

In the first step, each node creates a coverage neighbors record from its coverage target. It is best to assign a cluster head for each target to provide a specific mechanism for transferring the collected information to the sink, i.e., node 1 processes its neighbors table and adds the neighbors that have similar object coverage nodes to node 1 to node 1's coverage neighbors table (or Table 6). The coverage neighbor table format for target  $t_k$  for node  $s_i$  is shown in Table 6.

**Table 6** Coverage neighbors table format for target  $t_k$  for node  $s_i$  (i.e., here,  $s_j$  is one of  $s_i$ 's neighbors that has the same  $t_k$ )

ID ( $s_j$ )
GPS ( $s_j$ )
$E_r(s_j)$
Status ( $s_j$ )
ID ( $t_k$ )
$d(s_i, s_j)$
$D_{\text{sink}}(s_j)$
$d(s_j, t_k)$
CHeff( $t_k$ )
$\theta(t_k, s_j)$

where  $\theta(t_k, s_j)$  is the angle of the sensor node to the target node, CHeff( $t_k$ ) is the fitness function of the cluster head (CHeff is the acronym of Cluster Head Efficient), and  $d(s_j, t_k)$  is the distance of the sensor node from the target node. The sensors that are in the coverage table for each target node make a *cluster* with the name of that target.

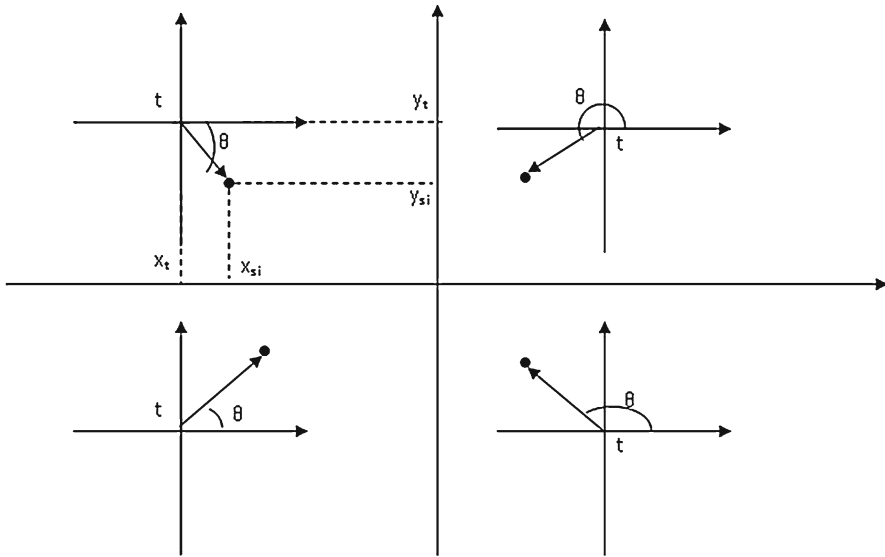
Now, we explain the mechanism that helps us to select the cluster head for each target node. Then, we will be able to transmit the information that is gathered by the covering sensors to the sink node.

In the second step, a fitness function is used to identify the cluster head, based on Eq. (4). The values  $\alpha_0$  and  $\alpha_1$  define the impact strength coefficients (probabilities) of each of the parameters for the cluster head fitness function (each of them is in the range[0,1], where 0 means that the parameter has no effect on the cluster head fitness function and 1 means the parameter has full effect on the cluster head function); these values are assumed to be 0.7 and 0.3, respectively (we consider the residual energy of each node to be more important than the distance of that node to the target node). The node with the highest value is defined as the cluster head. If the assigned node has no alternative, a timed loop will indicate that no message has been received from its neighbors and either it will change its position to cluster head (change its value to 4) or it will provide a convenient message to its coverage neighbors concerning the relevant target; otherwise it waits to receive messages from its coverage neighbors and updates the information in the coverage table. Here,  $E_{\text{max}}$  is the total energy that is considered for each node. Note that, if we have multiple nodes with the same CHeff, our algorithm will select the node that has the smallest  $D_{\text{sink}}(s_i)$ .

$$\text{CHeff}(t_k) = \alpha_0 \frac{E_r(s_i)}{E_{\text{max}}} + \alpha_1 \frac{D_{\text{sink}}(t_k)}{D_{\text{sink}}(s_i)}. \tag{4}$$

In the third step, it is necessary to know the angle of each node with respect to the coverage target to find the proper locations of *other*  $k-1$  nodes. Cluster head selects active coverage node by Eq. (5). Equation (5) is the usual formula for calculating the angle of each coverage target:

$$\theta(t_k, s_j) = \begin{cases} \pi + \tan^{-1} \left( \frac{y_{s_i} - y_{t_k}}{x_{s_i} - x_{t_k}} \right) & (x_{s_i} - x_{t_k}) < 0 \\ \tan^{-1} \left( \frac{y_{s_i} - y_{t_k}}{x_{s_i} - x_{t_k}} \right) & (x_{s_i} - x_{t_k}) > 0 \end{cases}, \tag{5}$$



**Fig. 6** Location of each sensor in the neighbor of each target

Figure 6 shows the location of each sensor in the neighbor of each target node.

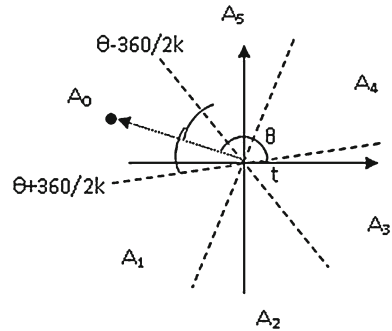
In principle, assume that the cluster head is in the bisector of sector  $A_0$ , it is in sector  $[(\theta - 2\pi)/2k \bmod 2\pi, (\theta + 2\pi)/2k \bmod 2\pi]$ . Consider that the circumference of the target that is segmented into  $k-1$  sectors is used to calculate the bisection of each sector. Interestingly, we select each node in each bisector as an active coverage node and change its status field to 3. The bisector of sectors  $A_j, B_s(A_j), j = 1, 2, \dots, *$  is calculated as in Eq. (6):

$$B_s(A_j) = \left( \theta + \left( j * \frac{2\pi}{k} \right) \right) \bmod 2\pi \quad j = 1, 2, \dots, k - 1, \tag{6}$$

We use the coverage fitness function to appoint the coverage node for each sector, where the node with the highest value is assigned to be the sector coverage node. In the coverage fitness function or *coverage efficient function* denoted as  $\text{cov}_{\text{Eff}}$  and expressed in Eq. (7), values  $\beta_0, \beta_1$ , and  $\beta_2$  are the impact strength coefficients (probabilities) applied to the parameters in the cluster head fitness function; the relevant values are 0.6, 0.2, and 0.2, in that order. Figure 7 depicts a target and closed sectors.

$$\text{cov}_{\text{Eff}}(s_i) = \beta_0 \frac{E_r(s_i)}{E_{\text{max}}} + \beta_1 \frac{1}{|B_s(\text{sec}_j) - \theta(t_k, s_i)| + \varepsilon} + \beta_2 \frac{d(s_i, t_k)}{R_{s_i}}, \tag{7}$$

where  $E_{\text{max}}$  is the total energy that is considered for node  $s_i$ . Obviously,  $B_s(\text{sec}_j)$  is defined in (6),  $R_{s_i}$  is coverage radius for node  $s_i$ ,  $\theta(t_k, s_j)$  is angle of the sensor node to the target node,  $d(s_j, t_k)$  is the distance of the sensor node from the target node,  $\varepsilon$  the fault tolerance parameter and  $E_r(s_i)$  is  $s_i$ 's residual energy.

**Fig. 7** Target and closed sectors

The cluster head sends a message to each coverage node following their nomination and waits define duration to receive a confirmation message; thus the target has coverage of order  $k$ , and its information is subsequently transferred to the cluster head by the coverage nodes and is finally received by the sink. The coverage nodes with a status value of 1 then enter sleep mode following termination of this stage until the next stage. In a nutshell, cluster head selects active coverage node. Particularly, cluster head is responsible to find active coverage node and set the active coverage node statues to 3. Note that, this could lead to multiple nodes covering a particular sector, but the only active node is the node that receives an active message from the cluster head.

Pseudocode for Stage II is provided in Algorithm 2.

#### 4.3 Phase III: selection of active transmitting nodes for information transfer to the sink

This stage is implemented concurrently with step 3 of Phase II to select the next route for sending information to the sink. Each node evaluates its neighbors for sending information to the next hop node using the neighbors' table information and communication fitness function (or coverage efficient function) values. The node with the highest value (i.e., these values are achieved within coverage efficient function) is nominated for packet transmission. To transmit packets from the cluster head to the sink, there should be an available route between them, which can be found by eliciting information using the *hop-to-hop* method.

In the hop-to-hop method, each node uses information it receives from the upstream nodes to select an appropriate node for the next hop. The *next-leap* node (or next hop node) should be selected from the communicative tables of each node's neighbors. The selection is made as follows.

First, it is necessary to determine whether a communicative and active node is available in the table. If one is found and in addition its distance to the sink node is less than the distance of the current node to the sink node, we select this node as the next-leap node; otherwise, if the distance is greater than the distance to the sink, the communicative fitness function [shown in Eq. (8)] will be used for nomination. In this case, the node with the greatest value will be chosen for the next-leap node.

**Algorithm 2:** Pseudocode for Phase II (Sensing\_node\_setup ( $s_i, t_u$ ))**Input:**  $s_i$  and  $t_u$ **Output:** Sensing\_node\_setup ( $s_i, t_u$ )

```

1 : begin
2 : Add information node  $s_i$  into coverage neighboring set ( $s_i, t_u$ );
3 : from table neighboring set ( $s_i$ ) While(ID( $s_i$ ))
4 : If (  $t_u == t_k$  ) then
5 :   begin
6 :     Update coverage neighboring set ( $s_i, t_u$ );
7 :     Compute destination  $s_i, t_k$  with  $d(s_i, t_k) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$ ;
8 :     Compute cluster_head efficient  $s_j$  with  $CHeff = [\alpha \times E_r(s_j) / E_{max}] + [\alpha \times D_{sink}(t_k) / D_{sink}(s_j)]$ 
9 :     for  $i=1$  to  $d-1$ 
10:      Compute angle  $s_j$  to target with  $\theta(t_k, s_j) = \begin{cases} \pi + \tan^{-1} \left( \frac{y_{s_j} - y_{t_k}}{x_{s_j} - x_{t_k}} \right) (x_{s_j} - x_{t_k}) < 0 \\ \tan^{-1} \left( \frac{y_{s_j} - y_{t_k}}{x_{s_j} - x_{t_k}} \right) (x_{s_j} - x_{t_k}) > 0 \end{cases}$ 
11:    end if
12:    Find max from field  $CHeff$  in table neighboring set ( $s_i$ );
13:    If  $CHeff$  is not unique then
14:      Find Minimum ( $D_{sink}(s_z)$ ) [ $Z = \{ \text{list of nodes that have same } CHeff \}$ ]
15:       $CHeff = CHeff(s_n)$ ;  $n$  is the node belongs to  $z$  and has Maximum  $CHeff$ 
16:    If (ID( $CHeff\_max$ ) == ID( $s_i$ )) then
17:      begin
18:         $status(s_i) = 4$ ;
19:         $s_i$  broadcasts its ID, Status to its 1-hop(single-hop) neighbors;
20:         $\Theta = \theta(t_k, s_i)$ ;
21:         $CHeff = s_i$ ;
22:      end if
23:    else if (receive message from neighbors) then
24:      Update coverage neighboring set ( $s_i, t_u$ ) && Update neighboring set ( $s_i$ );
25:    for ( $m=1$  ;  $m < k$  ;  $m++$ )
26:      begin
27:         $B_s(sec_m) = \Theta + (m * 2\pi / k) \bmod 2\pi$  ;
28:         $cov_{Eff}(max) = 0$ ;
29:        While( $\theta > B_s(sec_m) - 2\pi / 2k \bmod 2\pi$  &&  $\theta < B_s(sec_m) + 2\pi / 2k \bmod 2\pi$  )
30:          begin
31:            Compute coverage efficient sector  $m$  with

$$cov_{Eff}(s_i) = \beta_0 \cdot \frac{E_r(s_i)}{E_{max}} + \beta_1 \cdot \frac{1}{|B_s(sec_m) - \theta(t_k, s_i)| + \epsilon} + \beta_2 \cdot \frac{d(s_i, t_k)}{R_s}$$

32:            if ( $cov_{Eff}(s_i) > cov_{Eff}(s_{max})$ ) then //find node with maximum energy; i.e.  $s_{max}$  is the maximum energy of all nodes
33:               $cov_{Eff}(s_{max}) = cov_{Eff}(s_i)$ ;
34:              ID( $s_{max}$ ) = ID( $s_i$ );
35:            End while
36:             $CHeff$  broadcasts ID( $s_{max}$ ), Status=3 to its 1-hop neighbors;
37:          end for
38:        Return
39:      End

```

This procedure is repeated until the sink is reached. So that there always a route exists for transmission from the cluster head to the sink [39].

The reason for using the distance from the sink in the fitness function for coverage is to prevent deviation in the main route for packets to the sink. In addition, it is used for finding the fewest leaps in establishing a route to the sink.

Second, after selecting the next-leap (next hop node), that node is notified by a special message that it has been selected. Then the node changes its status to 2 and announces to its single-hop neighbors that it is an active communication node. Finally, this node selects the next node based on this same method, and this continues until the sink node is reached.

Next, after finishing an each iteration of the loop, as the next iteration commences, the nodes that had been activated in the previous iteration change their status to

the normal state. Also, these nodes send their residual energies information back to their single-hop neighbors so that they can update their neighbors table. Equation (8) shows the communicative fitness function or *communicative efficient function* denotes  $comm_{Eff}$ . This function shows the communicative value for  $j$ th neighbor node ( $s_j$ ) of current node  $s_i$ . Each node knows the number of its active neighbors. The maximum degree of each node is  $m$ . This value helps the node to find better neighbors based on their proximity to the sink, high residual energy, and the longest available distance based on the current node’s communicative radius. This is shown in detail in Eq. (8).

$$comm_{Eff}(s_j) = \gamma_0 \frac{E_r(s_j)}{E_{max}} + \gamma_1 \frac{D_{sink}(s_i)}{D_{sink}(s_j)} + \gamma_2 \frac{d(s_i, s_j)}{R_c} \quad j = 1, 2, \dots, m, \quad (8)$$

where  $\gamma_0$ ,  $\gamma_1$  and  $\gamma_2$  define as the impact strength coefficients (probabilities) or the contribution of each parameter to the communicative fitness function; their values are 0.6, 0.2, and 0.2, respectively.  $E_{max}$  is the total energy that is considered for node  $s_j$ .  $R_s$  is coverage radius for each node,  $d(s_i, s_j)$  is the distance of the sensor node  $s_i$  to the its neighbor node  $s_j$ ,  $D_{sink}(s_j)$  is the distance of the sensor node  $s_j$  to the sink,  $R_c$  is communicative radius of each node and  $E_r(s_j)$  is  $s_j$ ’s residual energy. In a nutshell, the cluster head picks a next hop to forward based on the  $comm_{Eff}$  value among neighboring nodes, and the selected node then picks another node and so forth.

Algorithm 3 provides pseudocode for Phase III.

---

**Algorithm 3:** Pseudocode for Phase III (communicative\_node\_setup ( $s_i, t_u$ ))

---

**Input:**  $s_i$  and  $t_u$   
**Output:** communicative\_node\_setup ( $s_i, t_u$ )

```

Begin
2: from table neighboring set ( $s_i$ )
   While (GPS( $s_i$ )!=GPS(sink))
3:   Begin
4:   If (Status ( $s_i$ )==2) then
5:     Begin
6:     If ( $D_{sink}(s_j) < D_{sink}(s_i)$  &&  $E_r(s_j) > E_{max}/2$ ) then
7:       next-hop= $s_j$ ;
8:     End If
9:     Else If (Status ( $s_j$ )==0) then
10:      Begin
11:      Compute  $comm_{Eff} s_j$  with  $comm_{Eff}(s_j) = \gamma_0 \cdot \frac{E_r(s_j)}{E_{max}} + \gamma_1 \cdot \frac{D_{sink}(s_i)}{D_{sink}(s_j)} + \gamma_2 \cdot \frac{d(s_i, s_j)}{R_c}$ 
12:      Find max from field  $comm_{Eff}$  in table neighboring set ( $s_i$ );
13:      If ( ID( $comm_{Eff} (s_{max})$ )==ID( $s_i$ )) then
14:        Begin
15:          Status( $s_i$ )=2;
16:          broadcasts ID( $s_i$ ), Status to 1-hop neighbors;
17:          next-hop= $s_j$ ; // for next-leap in routing
18:        End If
19:      End Else If
20:    End If
21:  End While
22: Return
23: End

```

---

In the following example, we describe aforementioned algorithm (e.g., Algorithm 3) in various figures below.

We have filled Table 7 based on the value obtained in Tables 1, 2, 3, 4, 5, 6 and calculated  $comm_{Eff}$  based on Eq. (8). In the Fig. 8a, each node with orange color has data to transmit to the sink; this node is interested to select a next node among its

**Table 7** Connectivity table format for node  $s_i$ 

$E_T(s_j)$	$E_{\max}$	$D_{\text{sink}}(s_i)$	$D_{\text{sink}}(s_j)$	$d(s_i, s_j)$	$R_c$	$\text{comm}_{\text{eff}}$
------------	------------	------------------------	------------------------	---------------	-------	----------------------------

communicative neighbor's nodes. For selecting next hop among nodes (the nodes with blue colors) that are in communicative radius (the dotted circle in the Fig. 8a), the node with maximum  $\text{comm}_{\text{eff}}$  is selected as a next node and the information transmitted to it.

The node 50 (i.e., it is the cluster head or is the Intermediate node in the path of sending packets to the sink) tries to send the information to the sink and it has 6 choices (these choices are related to the communicative neighbors node with one hop such as node 46, 47, 51, 61, 62, and 66). Regarding the value  $\text{comm}_{\text{eff}}$  in Table 8, we select the node 61 for the next step of sending information.

In Fig. 8b, node 61 has 7 choices based on its neighbors (i.e., nodes 65, 62, 60, 50, 46, 76, 66). Regarding the value  $\text{comm}_{\text{eff}}$  in Table 9, we select the node 59 for the next step of sending information. Also, in Fig. 8c, node 65 has 7 choices based on its neighbors (i.e., nodes 59, 60, 61, 66, 75, 76, 80). Based on the value  $\text{comm}_{\text{eff}}$  in Table 9, the node 61 is selected as the next hop transmitting data. Moreover, in Fig. 8d, node 59 has 7 choices based on its neighbors (i.e., nodes 34, 44, 58, 60, 65, 74, 75). Based on the value  $\text{comm}_{\text{eff}}$  in Table 9, node 58 is selected as the next hop for transmitting data. In addition, in Fig. 8e, node 58 has 7 choices based on its neighbors same before (i.e., nodes 34, 45, 57, 59, 73, 74, and 77). Based on the value  $\text{comm}_{\text{eff}}$  in Table 9, the node 45 is selected as the next hop for transmitting data. Next, in Fig. 8f, node 45 has 8 choices based on its neighbors (i.e., nodes 29, 31, 32, 34, 52, 57, 58, and 73). Based on the value  $\text{comm}_{\text{eff}}$  in Table 9, node 52 is selected as the next hop for transmitting data. Finally, in Fig. 8g, due to the fact that node 52 is able to reach directly to the sink, it delivers the data directly to the sink.

## 5 Performance evaluation

### 5.1 Experimental setup

Since implementation of these techniques is expensive and time-consuming, a simulation was used to evaluate their performance. This section presents an evaluation through a simulation and provides the relevant results.

Glomosim is used for special network simulations, with TCP, routing, and multicast protocols in a wireless network and satellites [40]. Glomosim can simulate myriad nodes using C functions and Parsec. Glomosim is free open-source code which can be assembled and run on Linux- and Windows-based operating systems [40].

Our simulations assumed an environment spanning (TERRAIN-DIMENSIONS)  $200 \times 200$  sq m, with 1,000 sensor nodes deployed randomly in that space, including random nodes. We consider each node radius  $R_s$  to have a uniform distribution between [4, 6] m. As we see, each node is able to cover several nodes based on its radius, also,  $R_c$  is considered one meter (fixed) larger than its  $R_s$  for each node. The nodes receive



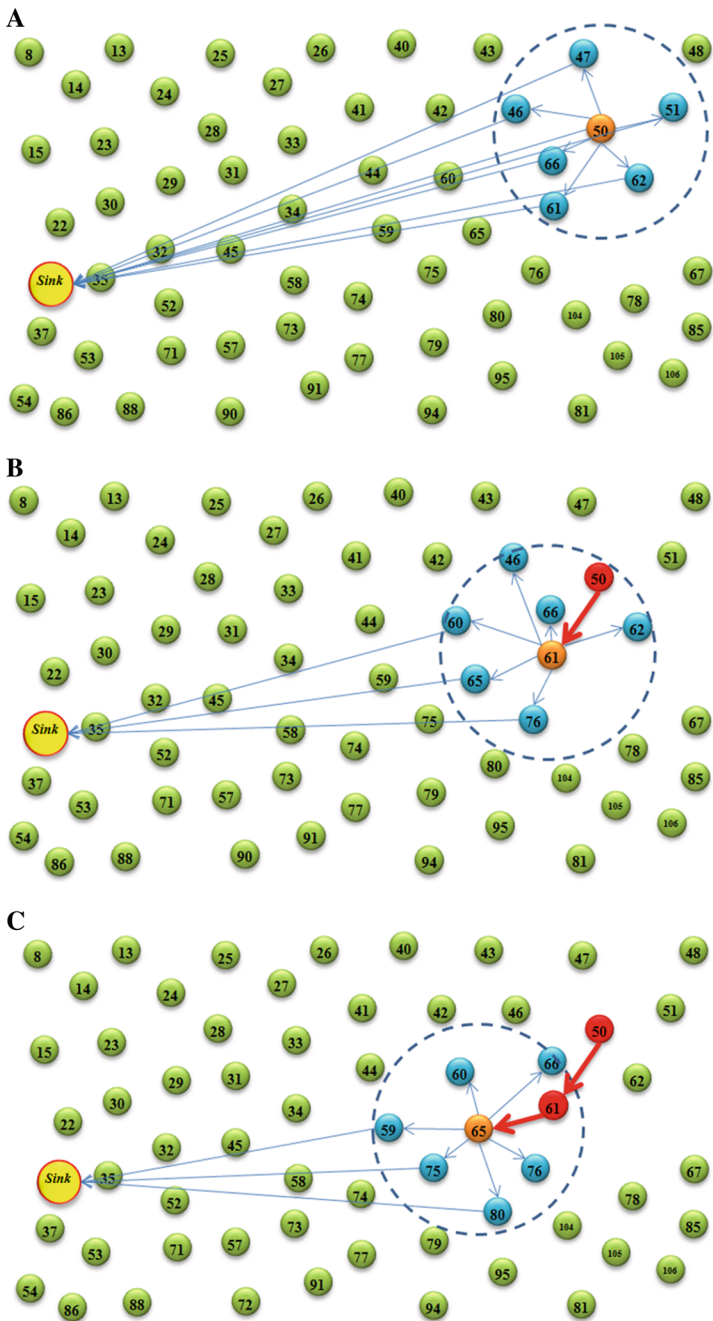


Fig. 8 An example of selection of active transmitting nodes for information transfer to the sink

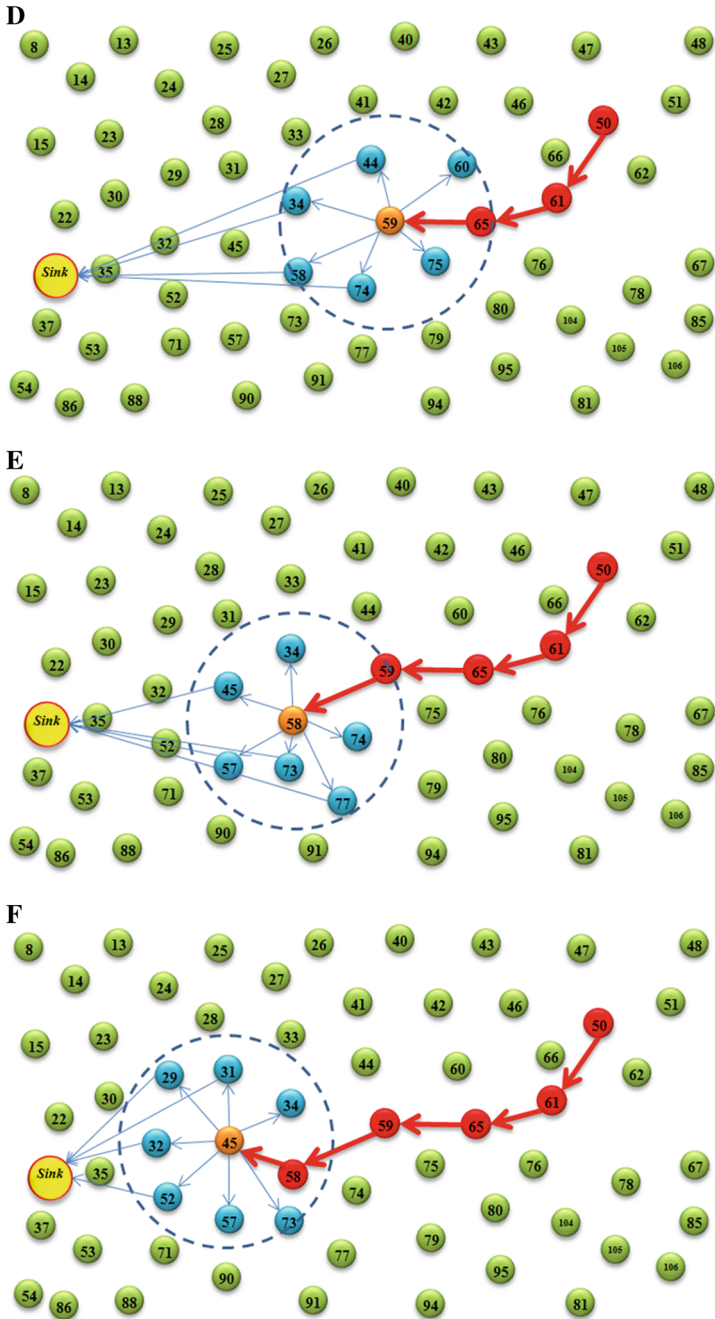


Fig. 8 continued

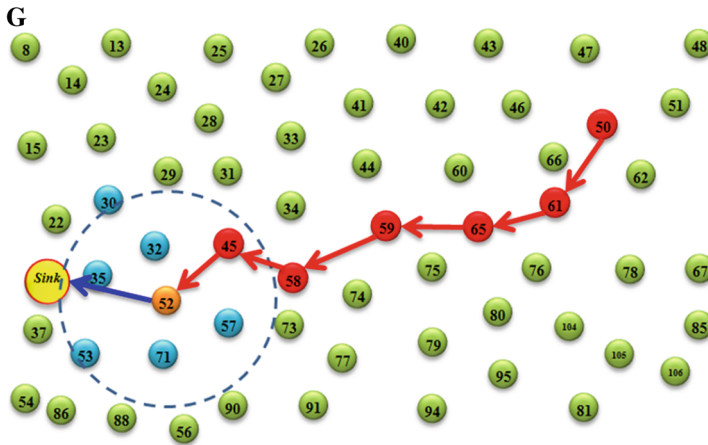


Fig. 8 continued

Table 8 Table 7 values for the Fig. 8a

	Node 46	Node 47	Node 51	Node 61	Node 62	Node 66
$E_r(s_j)$	4.5	4.6	4.4	4.7	4.5	4.3
$E_{max}$	5	5	5	5	5	5
$D_{sink}(50)$	11.303	11.303	11.303	11.303	11.303	11.303
$D_{sink}(s_j)$	9.757	11.521	12.913	9.975	11.841	10.146
$d(50,s_j)$	1.252	1.034	1.035	1.333	0.727	0.644
$R_c$	2	2	2	2	2	2
$comm_{Eff}$	0.89689	0.851616	0.806564	0.923927	0.803613	0.803207

object information and transfer it through cluster heads and communicative nodes to a sink location situated at the center of the environment. The simulation parameters are listed in Table 10.

Specifically, the objects are varied in their circumstances; the initial energy of the sensors is 10,000 mJ while the transmitted energy is 10 mJ across the greatest distance; and there is 5 mJ for each supervisory action.

### 5.2 Experimental results

The proposed algorithm is compared with the EAP technique with respect to objects, links, and energy consumption. The coverage and relevant results are presented. Compared with the two classic protocols for sensor networks, HEED [32] and LEACH [34], the performance of EAP [31,35] is more optimized due to its network coverage and information transfer. Each test was carried out in different conditions and with different settings. The graphs that are displayed depict the average of the several test results. In conclusion, we use EAP because it pays close attention to clustering, energy-efficiency and energy consumption of nodes in whole networks, and each node uses broadcasting

**Table 9** comm<sub>Eff</sub> for node 50, 61, 65, 59, 58, 45

Node 50	Node 46	Node 47	Node 51	Node 61	Node 62	Node 66		
comm <sub>Eff</sub>	0.89689	0.851616	0.806564	0.923927	0.803613	0.803207		
Node 61	Node 46	Node 50	Node 60	Node 62	Node 65	Node 66	Node 76	
comm <sub>Eff</sub>	0.857669	0.789802	0.89266	0.824263	0.937314	0.735758	0.862815	
Node 65	Node 59	Node 60	Node 61	Node 66	Node 75	Node 76	Node 80	
comm <sub>Eff</sub>	0.931403	0.768289	0.804273	0.822358	0.82506	0.805914	0.832951	
Node 59	Node 34	Node 44	Node 58	Node 60	Node 65	Node 74	Node 75	
comm <sub>Eff</sub>	0.961639	0.832788	0.999217	0.721859	0.754224	0.863411	0.77271	
Node 58	Node 34	Node 45	Node 57	Node 59	Node 73	Node 74	Node 77	
comm <sub>Eff</sub>	0.819106	0.915616	0.907777	0.781994	0.787971	0.774351	0.845857	
Node 45	Node 29	Node 31	Node 32	Node 34	Node 52	Node 57	Node 58	Node 73
comm <sub>Eff</sub>	0.903718	0.812091	1.008896	0.769576	1.009626	0.830378	0.755753	0.845506

**Table 10** The simulation parameters

Parameter	Value
SIMULATION-TIME	Default value (15 ms)
SEED	1
TERRAIN-DIMENSIONS	(200, 200) m
Number of sensors (NUMBER OF NODES)	1,000
Initial energy	10,000 mJ = 10 J
MOBILITY	NONE
Transmitted energy (communication energy)	10 mJ
Supervisory action of each transmission	5 mJ
NODE PLACEMENT	UNIFORM
ROUTING PROTOCOL	PROPOSED ALGORITHM
RADIO-TYPE	RADIO-ACCNOISE
RADIO FREQUENCY FOR EACH NODE	2.4e9 (Hz)
RADIO-BANDWIDTH	2M (Hz)
MAC-PROTOCOL	CSMA
TEMPERATURE	290 (K)
NETWORK-PROTOCOL	IP
APPLICATION	FTP

and has residual energy, it supports in-network, it extends LEACH [34] algorithm and makes a good balance between network parameters such as connectivity, coverage and power consumption.

To demonstrate the range of diversity of active sensors in connection with the proposed technique, the number of targets is varied from 1 to 10 in these simulations, and this was used to calculate the number of active sensors.

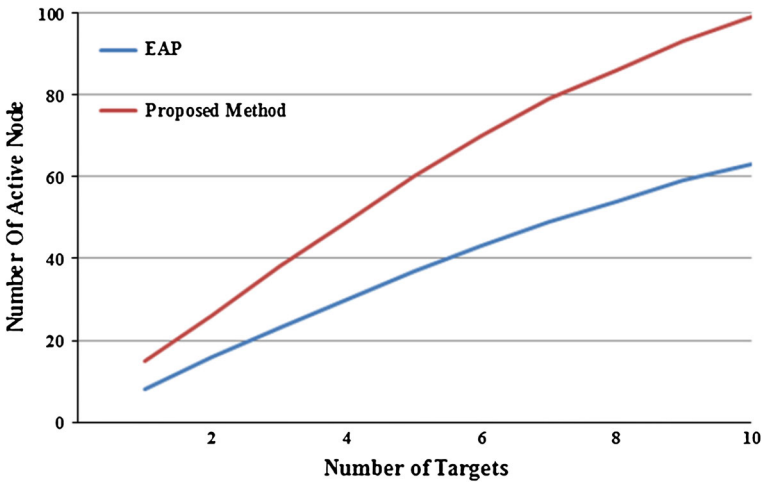


Fig. 9 Number of active nodes proportional to the number of targets

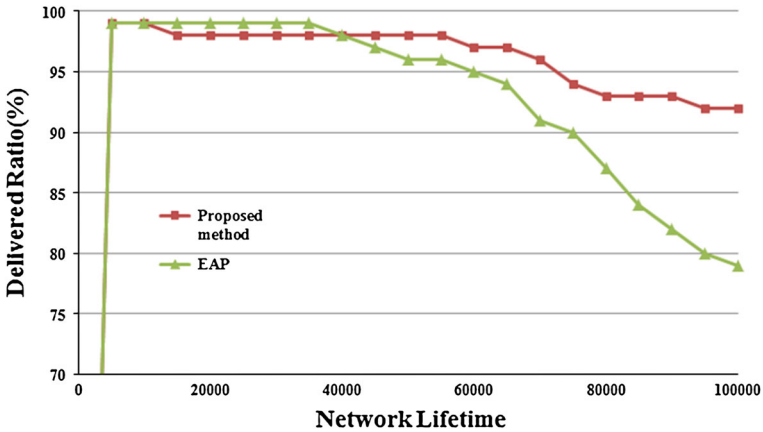
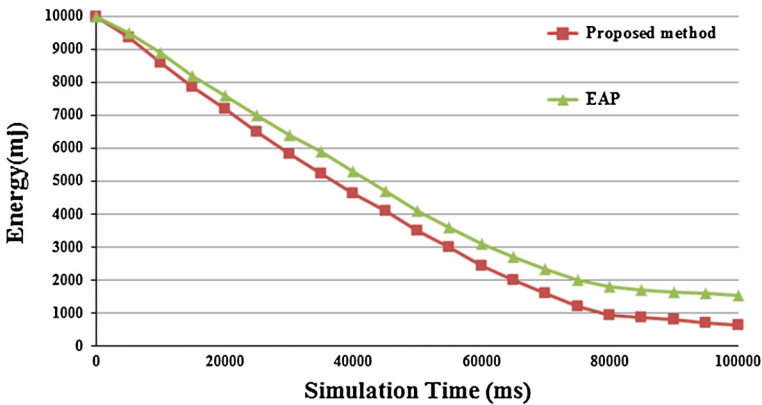


Fig. 10 Percentage of delivered packets that are received in the network (lifetime in ms)

For comparing the proposed technique with the EAP technique, the range of coverage was assumed to be in status (position number) number 4 (i.e., according to Table 2), since the range of coverage affects the simulation results. Figure 9 shows the number of active nodes as a function of the targets’ number. The number of active nodes in the proposed technique is higher than in the EAP technique, and the communicative radius of sensors remains at the highest value. This decreases the proposed technique’s rate of energy consumption in the network, because we are able to use the *k-coverage* again to transmit information to the sink with the help of other nodes that are in the coverage area.

The percentage of delivered packets is shown in Fig. 10.

Delivered ratio means the ratio of the (received) delivered packets in sink into produced and transmitted packets over times. The proposed method has better performance by increasing the time than does EAP, because the packets are transferred to



**Fig. 11** The range of energy consumption in the network (mJ/ms)

the sink by creating a route through the cluster head and multi-hop transferring in the proposed technique, but it takes the packets longer to reach the sink due to the delivery of packets to the sink as multi-hop transfers in EAP. Furthermore, in the proposed method, the range of delivered messages is 100 % when there is little congestion at the communication nodes, but the range is reduced over time with the range reaching 93 % at the termination of the network's life, whereas the EAP technique successfully delivers approximately 86 % of the packets by the end of the network lifetime.

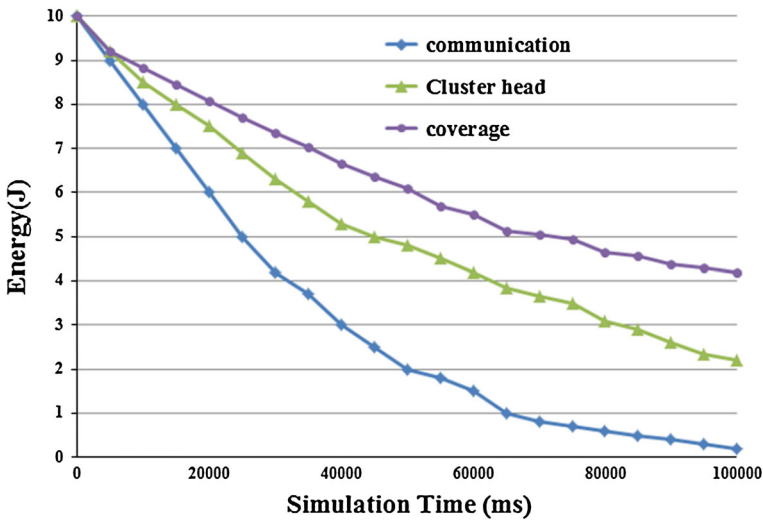
In Fig. 11, because the EAP technique increases the communicative radius to communicate with clusters, the energy consumption in the cluster head node increases. However, the proposed technique results in less energy consumption than the EAP technique because the least number of sensors are used for communication between cluster heads and for information. The coverage nodes do not need to distribute the entire information toward their neighbors, and consequently the range of energy consumption is better than for the EAP algorithm.

Figure 12 depicts the range of energy consumption for different available nodes in the network in the proposed technique. The graph shows the demonstrated energy range for three categories of nodes:

- Nodes that participate in the coverage process.
- Nodes that only take participate in information transmission.
- Nodes that only assume the role of coverage cluster head.

The energy consumption in available coverage nodes is less and has the smallest range of energy consumption due to a failure of information transmission and a lack of complete information about neighbors at the beginning of the algorithm.

The cluster head nodes spend more energy than the coverage nodes due to the information received by coverage nodes and the information transferred to communicative nodes; the information transmitted to the sink consumes maximal energy in the network due to the amount of information transferred to the sink.



**Fig. 12** Residual energy for different categories of nodes in the proposed technique (J/ms)

## 6 Conclusion and future research

Regional coverage is one of the main characteristics in a wireless sensor network, with the main aim being to achieve the maximum coverage in the area with the least energy consumption by the sensors. The initial and main operation of each sensor node is to review its vicinity to provide the required information for operating purposes. The routing protocol for coverage should provide low energy consumption to increase the network lifetime and the warranty of network connection. High coverage is advantageous for promoting error tolerance and the validity of received information. Our simulation results and comparison of the routing algorithms' retention of coverage prove that the proposed algorithm preserves the network with its low energy consumption.

The paper reviews the relevant characteristics of sensor nodes, the functions and applications represented, as well as the types of sensor and their structures and the constraints they encounter. The constraints of these networks are energy limitations, network lifetime, and network coverage. Therefore, relevant strategies are provided to control the network coverage. The proposed technique is a rapid reduction of nodes around the sink. Only a few nodes are used around the sink in spite of a high node density assumed in the environment. These nodes assume the burden of relaying information to the sink resulting in a reduction of energy consumption in the nodes. One way to address this challenge is the enervation of the sink in the environment.

It is plausible to increase the range of coverage by  $k$  nodes to upgrade the tolerance against faults in the future, but that strategy is insufficient because the connection of the sink with an object or part of the network is vulnerable to de-linkage in the case of a discrepancy along the route or a discrepancy of the supervisory node. Therefore, it is desirable to provide a convenient strategy for upgrading the network's resistance to network discrepancy. Specifically, we mention that, if a hole exists, we try to find

another path around hole to bypass the hole and it might be a longer path. In the worst case, if our network breaks into two completely isolated sub-networks, no algorithm can find a path.

## References

1. Akyildiz I, Su W, Sankarasubramaniam Y, Cayirci E (2008) A survey on sensor networks. *Comput Netw* 52(12):2292–2330
2. Pottie GJ, Kaiser WJ (2000) Wireless sensor networks. *Commun ACM* 43(5):51–58
3. Labrador MA, Wightman PM (2009) Topology control in wireless sensor networks. In Springer, Berlin ISBN: 978-1-40209589-9
4. Shojafar M, Pooranian Z, Shojafar M, Abraham A (2013) LLLA: New Efficient Channel Assignment Method in Wireless Mesh Networks. Springer International Publishing, *Innovations in Bio-inspired Computing and Applications* 237:143–152
5. Akyildiz IF, Melodia T, Chowdhury KR (2007) A survey on wireless multimedia sensor networks. *Comput Netw* 51(4):921–960
6. Sayyad A, Ahmadi A, Shojafar M, Meybodi MR (2010) Improvement multiplicity of routs in directed diffusion by learning automata new approach in directed diffusion. In: International conference on computer technology and, development, pp 195–200
7. Sayyad A, Shojafar M, Delkhal Z, Ahamadi A (2011) Region directed diffusion in sensor network using learning automata: RDDLA. *J Adv Comput Res* 1(3):71–83
8. Sayyad A, Shojafar M, Delkhal Z, Meybodi MR (2010) Improving directed diffusion in sensor network using learning automata: ddla new approach in directed diffusion. In: 2nd International conference on computer technology and development (ICCTD 2010), pp 189–194
9. Ghosh A, Das SK (2008) Coverage and connectivity issues in wireless sensor networks: a survey. *Pervasive Mobile Comput* 4(1):303–334
10. Younis M, Akkaya K (2008) Strategies and techniques for node placement in wireless sensor networks: a survey. *Elsevier Ad Hoc Netw* 6:621–655
11. Stojmenovic I, Lin X (2001) Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Trans Parallel Distrib Syst* 12(10):1045–9219
12. Yang S, Dai F, Cardei M, Wu J (2006) On connected multiple point coverage in wireless Sensor networks. *J Wirel Inf Netw*
13. Huang Y, Tseng Y (2003) The coverage problem in a wireless sensor networks. In: Proceedings of the 2th ACM international conference on information proceedings in sensor networks and applications, pp 115–121
14. Benyuan L, Dousse O, Nain P, Towsley D (2013) Dynamic coverage of mobile sensor networks. *IEEE Trans Parallel Distrib Syst* 24(2):301–311
15. Li S, Kao HC (2010) Distributed K-coverage self-location estimation scheme based on Voronoi diagram. *Commun IET* 4(2):167–177
16. So AM, Ye Y (2005) On solving coverage problems in a wireless sensor network using Voronoi diagrams. In: Proceedings of workshop on internet and network economics, vol 3828, pp 584–593
17. Okabe T, Boots B, Sugihara K, Chiu SN (2000) Spatial tessellations: concepts and applications of Voronoi diagrams. 2nd edn. Wiley, New York
18. Powers RA (1995) Batteries for low power electronics. *Proc IEEE* 83(4):687–693
19. Shen CC, Srisathapornphat C, Jaikao C (2001) Sensor information networking architecture and applications. *IEEE Pers Commun* 8(4):52–59
20. Howard A, Mataric M, Sukhatme G (2002) Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem. In: Distributed autonomous robotic systems, vol 5. Springer, Berlin, pp 299–308
21. Hefeeda M, Bagheri M (2007) Randomized K-Coverage algorithms for dense sensor networks. In: Proceedings of 26th IEEE international conference on communications pp 2376–2380
22. Nath S, Gibbons PB (2007) Communicating via fireflies: geographic routing on duty-cycled sensors. In: Proceedings of the 6th international conference on information proceedings in sensor, networks, pp 440–449



23. Anastasi G, Conti M, Francesco MD, Passarella A (2009) Energy conservation in wireless sensor networks: a survey. *Ad Hoc Netw* 7(3):537–568
24. Wang L, Xiao Y (2006) A survey of energy-efficient scheduling mechanism in sensor networks. *Mobile Netw Appl* 11(5):723–740
25. Chakrabarty K, Iyengar S, Qi H, Cho E (2002) Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Trans Comput* 51(12):1448–1453
26. Jin Yu, Jian P, Zhou S, LinYa P (2009) A survey on position-based routing algorithms in wireless sensor networks. *Algorithms* 2(1):158–182
27. Xing G, Wang W, Zhang Y, Lu C, Pless R, Gill C (2005) Integrated coverage and connectivity configuration in wireless sensor networks. *ACM Trans Sensor Netw (TOSN)* 1(1)
28. Ilyas M, Mahgoub I (2005) *Handbook of sensor networks: compact wireless and wired sensing systems*. CRC Press, Boca Raton ISBN: 0-8493-1968-4
29. Cordeschi N, Shojafar M, Baccarelli E (2013) Energy-saving selfconfiguring networked data centers. *Comput Netw* 57(17): 3479–3491.
30. Zhang H, Hou J (2005) Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc Sensor Wirel Netw* 1(12):89–123
31. Younis O, Fahmy S (2004) HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Trans Mobile Comput* 3(4):366–379
32. Luchmun R, Pyanee M, Khedo KK (2012) Hierarchical hybrid energy efficient distributed clustering algorithm. *Int J Comput Distrib Syst* 2(1)
33. Tsai Y (2007) Coverage-preserving routing protocols for randomly distributed wireless sensor networks. *IEEE Trans Wirel Commun* 6(4):1240–1245
34. Heinzelman WR, Chandrakasan A, Balakrishnan H (2000) Energy efficient communication protocols for wireless micro sensor networks. In: *Proceedings of Hawaii international conference on system sciences*
35. Liu A, Jin X, Cui G, Chen Z (2013) Deployment guidelines for achieving maximum lifetime and avoiding energy holes in sensor network. *Inf Sci* 230:197–226
36. Dong Y, Quan Q, Zhang J (2008) Priority-based energy aware and coverage preserving routing for wireless sensor network. In: *Proceedings of vehicular technology conference*, pp 138–142
37. Jin Y, Jo J, Wang L, Ki Y (2008) ECCRA: an energy-efficient coverage and connectivity preserving routing algorithm under border effects in wireless sensor networks. *Comput Commun* 31(10):2398–2407
38. Ahmed N, Kanhere SS, Jha S (2005) The holes problem in wireless sensor networks: a survey. *ACM SIGMOBILE Mobile Comput Commun Rev* 9(2):4–18
39. Omranpour H, Ebadzadeh MM, Barzegar S, Shojafar M (2008) Distributed coloring of the graph edges. In: *Proceedings of 7th IEEE international conference on cybernetic intelligent systems 2008 (CIS2008)*, UK, pp 1–5
40. Zeng X, Bagrodia R, Gerla M (1998) GloMoSim: a library for parallel simulation of large-scale wireless networks. In: *Proceedings of parallel and distributed simulation, (PADS 98)*, vol 161, p 154