

Research Article

Routing and Clustering of Sensor Nodes in the Honeycomb Architecture

Amine Rais , **Khalid Bouragba**, and **Mohammed Ouzzif**

Laboratory RITM, ENSEM, Hassan II University of Casablanca, Casablanca, Morocco

Correspondence should be addressed to Amine Rais; amine.343@hotmail.fr

Received 20 October 2018; Accepted 8 January 2019; Published 25 March 2019

Academic Editor: Sabrina Gaito

Copyright © 2019 Amine Rais et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Energy is the most valuable resource in wireless sensor networks; this resource is limited and much in demand during routing and communication between sensor nodes. Hierarchy structuring of the network into clusters allows reducing the energy consumption by using small distance transmissions within clusters in a multihop manner. In this article, we choose to use a hybrid routing protocol named Efficient Honeycomb Clustering Algorithm (EHCA), which is at the same time hierarchical and geographical protocol by using honeycomb clustering. This kind of clustering guarantees the balancing of the energy consumption through changing in each round the location of the cluster head, which is in a given vertex of the honeycomb cluster. The combination of geographical and hierarchical routing with the use of honeycomb clustering has proved its efficiency; the performances of our protocol outperform the existing protocols in terms of the number of nodes alive, the latency of data delivery, and the percentage of successful data delivery to the sinks. The simulations testify the superiority of our protocol against the existing geographical and hierarchical protocols.

1. Introduction

The clustering operation in WSN has proved its efficiency to extend the lifetime of the sensor nodes. In the hierarchical structure, the cluster head (CH) takes into charge the role of receiving, processing, and aggregating the sensed data of the member nodes and transmit them to the sink; thus, the battery lifetime of the CH is critical, and a good policy to extend it must be followed. The choice of the CHs from the nodes of the network is important to guarantee a good distribution of the nodes on the clusters, improving the load balancing and reducing the transmission costs to the CHs. In the large-scale sensor networks, the complexity of the routing is reduced by locally managing intracluster communication through the elected CHs, and it is the efficient way to decrease the energy consumption in this kind of network.

The communication between the CH and the member nodes of the cluster can be done in a single hop or in a multihop manner; the first one increases the energy consumption if the size of the cluster is large (this is the case in the large-scale networks). In addition, if we restrict

ourselves to the clusters with small sizes, the number of clusters will become very important; we can even have clusters called singletons (clusters with a single member); as result, we get closer to the flat model rather than to the hierarchical one. Thus, the choice of the multihop communication in intraclusters allows achieving the compromise between the size and the number of clusters in the network.

In a cluster with multihop routing, the neighboring nodes of the CH are critical nodes; the information sensed by the member nodes is forwarded to these nodes, and they are the gateways to the CH. The high traffic of data received by these nodes will deplete their remaining energy quickly, and the routing of the sensed data to their CH will be difficult, which will cause the collapse of the whole network. To avoid this situation, the role of the CH must be rotated among the nodes of the cluster to balance the consumed energy.

The hexagon is an ideal form for grouping nodes in WSN. Using the hexagons, a large sensed area can be partitioned into equal, adjacent, and nonoverlapping sub-zones; the hexagon is the largest polygon in terms of number

of sides that has this property. In the partition to adjacent triangular or square cells, every cell has three or four possible neighbors at one hop, but in the hexagonal cells, every cell has six adjacent cells that represent its possible next hop to all directions. More possible routes mean fewer overloads on the critical nodes.

The honeycomb is composed of a set of hexagonal, adjacent, and uniform cells. In this article, a new protocol is proposed for the grouping of sensor nodes into a virtual grid of adjacent and uniform honeycombs with multihop routing within the honeycombs. Most existing clustering protocols rebuild clusters at each round, which requires a large amount of energy, and they suffer from a bad energy distribution within the clusters, but our protocol partitions the network into honeycombs that remain unchanged. In each round, it only changes the position of the CH within the clusters to balance the energy consumption and reduce the overload on the critical nodes of the clusters.

The rest of this paper is organized as follows. Section 2 presents the related works. We describe in detail the EHCA protocol in Section 3. The results and simulations are discussed in Section 4, and we conclude this paper in Section 5.

2. Related Works

The routing in WSN consists in forwarding the sensed data from the sensor nodes to the sinks due to the limited resources of the sensors. Each node has to transmit data only to its neighbors. In order to transmit the data to the remote nodes, it is essential to pass through intermediate nodes; hence, the nodes must be located in relation to each other and create links between them to route the data, which is the role of the routing protocols. In WSNs, they can be categorized according to their topologies into 3 kinds: flat routing, hierarchical routing, and geographical routing. The protocol presented in this article is in the same time hierarchical and geographical protocol; for this reason, we focus on these 2 kinds of routing.

2.1. Hierarchical Routing. The CHs are responsible for retrieving data from the member nodes of the cluster, collecting the received data, and sending them to the base station [1, 2]. The data are merged and aggregated at the CH level to decrease the number of messages, which means that this kind of routing protocol can reduce the energy consumption and improve network performances.

Heinzelman et al. proposed the LEACH (low-energy adaptive cluster hierarchy) protocol [3], and it is a hierarchical routing protocol created for the WSN. Its main advantage is to minimize the energy consumption of the network elements. In LEACH, the nodes self-elect periodically to be CHs. Indeed, each node n takes a random value between 0 and 1; if this value is less than a threshold T , calculated as a function of the desired percentage of CHs and the number of iterations during which a node took the role of CH, the node n denotes CH. CHs inform their neighbors of their election. Each unelected node joins the nearest CH, based on the power of the received signals. Within a cluster, each node communicates in direct connection with its CH, according to a schedule TDMA

established by this latter in the formation of clusters. The nodes can put their communications system in the standby state while waiting for their turn, which allows an energy saving. At the expiration of a TDMA frame, the CH performs processing (aggregation, merging, etc.) on the data collected by the elements of its own cluster and then transmits the result directly to the sink which is supposed to be remote, which causes high energy consumption. To avoid this problem, Yu et al. propose the LEACH-R (LEACH revised) protocol [4] where the communication between the sink and the CHs is performed in a multihop manner, and this allows preserving the remaining energy of the CHs specifically for the large-scale networks. This protocol is based on the communication cost and the number of the active nodes in the network; it is a dynamic clustering algorithm which will be adapted to the changes in the remaining energy of the nodes and the scale of the network.

The LEACH authors have proposed a centralized version called LEACH-C [5]; the cluster structure is calculated in the sink to ensure an equitable distribution of the CHs on the network and a balanced cluster size. This allows balancing the energy consumption across the network and limits the energy dissipation; however, the centralized version of LEACH is not suitable for large-scale WSN.

PEGASIS (power-efficient gathering in sensor information systems) is considered as an optimization of LEACH [6], proposed by Lindsey and Raghavendra in [7]. It groups the nodes of the network in the form of a long chain based on the principle that a node can communicate only with the closest neighbor node. Thus, it adjusts its radio for a very short communication to conserve its energy. To communicate with the sink, the process is organized in rounds; during each round, a single node is allowed to communicate with the sink directly. This privilege is granted to all the nodes of the network in turn. A better conservation of energy is also obtained by aggregating the data on each node of the network. An improvement of PEGASIS called H-PEGASIS (hierarchical PEGASIS) [8] has tried to solve the problem of data delivery time by adopting parallel communications with the sink for geographically distant nodes.

Threshold sensitive energy-efficient sensor network protocol (TEEN) [9] is designed to be sensitive to sudden changes of the attributes such as the temperature. The reactivity is important for the critical applications where the network operates in a reactive mode. The architecture of the sensors network is based on a hierarchical grouping where the nodes create the clusters, and this process will be repeated until the sink is reached.

Adaptive threshold sensitive energy-efficient sensor network protocol (APTEEN) [10] is an extension of TEEN protocol that makes at the same time the gathering of the periodic data capture and responding to the critical events. When the sink forms clusters, the CHs broadcast the attributes, the threshold values, and the transmission schedule to all nodes. The CH also performs data aggregation to save energy.

The authors in [11] presented a new efficient protocol to extend the network lifetime named energy-efficient LEACH (EE-LEACH). The nodes are deployed in the sensing field based on the Gaussian distribution; the CHs are selected from the nodes with the higher remaining energy, and the

aggregation of the sensed data is based on the data ensemble. After the creation of the clusters, the forwarding nodes are selected from the nodes with the higher residual energy, and the other nodes are ignored in the routing operation, which improve the packet delivery ratio, save the energy, and extend the lifetime of the network.

HEER (Hamilton energy-efficient routing protocol) [12] is a routing protocol that takes into account energy and delays, based on node clustering and the Hamilton path concept. HEER forms clusters in the initialization phase of the network and connects the members of each cluster on a Hamilton path, built using a greedy algorithm, for data transmission purpose. No reconstitution of the cluster is required and the members of the path will become CH in turn.

There are many benefits of HEER protocol: it adopts the Hamilton path concept to connect the members of each formed cluster without the need for global nodes position information, which reduces the transmission distance for each cluster member and minimize the pressure of traffic and energy consumption at CHs. The clusters in HEER are formed only once in the first round, so the life of the RCSF can be extended. HEER traverses all the clustered nodes once for each round so the nodes are not consulted repeatedly; this feature reduces frequent access to the CH.

The authors of the article [13] introduced a hybrid routing protocol that takes energy into account for heterogeneous RCSF. H-CERP (hybrid clustering energy aware routing protocol) is designed to form efficient clusters with a number of CHs lower than the optimal estimate and uses multihop communication with gateway nodes to communicate with the base station. This new approach makes the system more advantageous when network life and sensor coverage are essential with no additional cost. By deploying H-CERP in a designed environment, the results obtained are promising in terms of energy consumption, residual energy, and node lifetime compared to generic methods such as LEACH, PEGASIS, and other recent protocols.

Low energy aggregation and routing are two well-known optimization problems that have been widely studied to extend the lifetime of the network in WSN. In EECR-PSO (energy-efficient clustering and routing-PSO) [14], clustering and routing are performed based on the PSO (particle swarm optimization) algorithm [15, 16], and a multiobjective fitness function is used for routing. The EECR-PSO protocol energy model is similar to that of LEACH. The purpose of this clustering algorithm is to maintain the energy of the nodes and balance the load in the network. EECR-PSO is significantly improved in terms of the network lifetime, the power consumption, the number of the dead nodes, and the total number of the packets transferred to the base station.

Chang and Ju proposed in [17] another hierarchical protocol named saving energy clustering algorithm (SECA); it takes into account the location of the nodes to choose the CHs and create clusters, and it is a centralized clustering algorithm that uses a modified version of the k -means algorithm to minimize the average distance between the nodes and the CHs. Then, the nodes save the power in their communication with the CHs and consequently minimize the energy consumption of the nodes and extend the lifetime of the

network. The authors have succeeded to create uniform distributed clusters and chosen the CHs based on the location and the remaining energy of the nodes; thus, the load of the network is distributed between the clusters, and the energy consumption has been balanced among them. There are other hierarchical clustering protocols that take into account the location of nodes to create clusters in WSN; they were proposed by Lloret et al. and Mehmood et al. in [18, 19].

2.2. Geographical Routing. In the geographical routing, the packets are not routed according to the identification of a specific destination, but rather in relation to a target zone. This zone can have several nodes, and any node can play the destination role for the packet. The geographical routing effectively minimizes the energy consumption, but it requires the location of the nodes that are generally deployed in a random manner. Since the positioning system GPS (Global Positioning System) is not suitable for the WSN due to the limited resources, the development of new localization techniques is necessary to be able to use this kind of routing.

GEAR (geographic and energy aware routing) [20] is a protocol that performs local broadcasting. A packet is transferred to a target region, which is then flooded. This algorithm uses metrics based on the next hop distance and the remaining energy of the nodes, so if a node has a depleted battery, the algorithm will try to avoid it. Thus a reduction in communications is obtained by locating the broadcast and a distribution of energy expenditure by taking into account the residual energy in each node. On the contrary, the protocol requires that each node knows its position, which leads to additional energy expenditure.

GAF (Geographic Adaptive Fidelity) [21] is an algorithm initially designed for classical ad hoc networks. Each node must have a location system. The algorithm then forms a virtual grid that covers the whole network and separates it into several square cells. Each node can be in active, discovery, or sleeping state, and the nodes move between these 3 states to save energy; the algorithm guarantees the connectivity of the network by ensuring that at least one active node per square cell is used, and the size of the cell must be chosen based on the radio range of the nodes to guarantee that nodes in the adjacent cells can communicate with each other. The protocol also allows managing mobile networks, where each node informs its neighbors of the estimated time it will leave its cell to update the topology of the network. The fact of having nodes extinguished saves energy, but the system remains rather rudimentary, and if the network is not very dense (few nodes per cell), the energy saving becomes very limited.

3. Architecture and Description of Our Protocol

EHCA (Efficient Honeycomb Clustering Algorithm) is a hybrid routing protocol; it is both hierarchical and geographical, and the nodes are located and grouped in clusters in the form of honeycombs. It is also a distributed protocol; the nodes make decisions in a collective manner without referring to the sinks, and all operations are performed by the nodes in an autonomous way.

Our protocol is executed on two steps: setup step and steady-state step. There are three operations that are operated in the setup step: the localization of the sensor nodes, the partition of the network into honeycomb clusters, and the cells addressing. The steady-state step can be divided into rounds, and every round contains two phases: the CH selection phase and the communication phase. In the CH selection phase, we change the location of the CH cell every round to balance the energy in intracluster. The routing in intra- and intercluster is happening in the communication phase. The flowchart in Figure 1 represents the various steps and phases of the EHCA protocol.

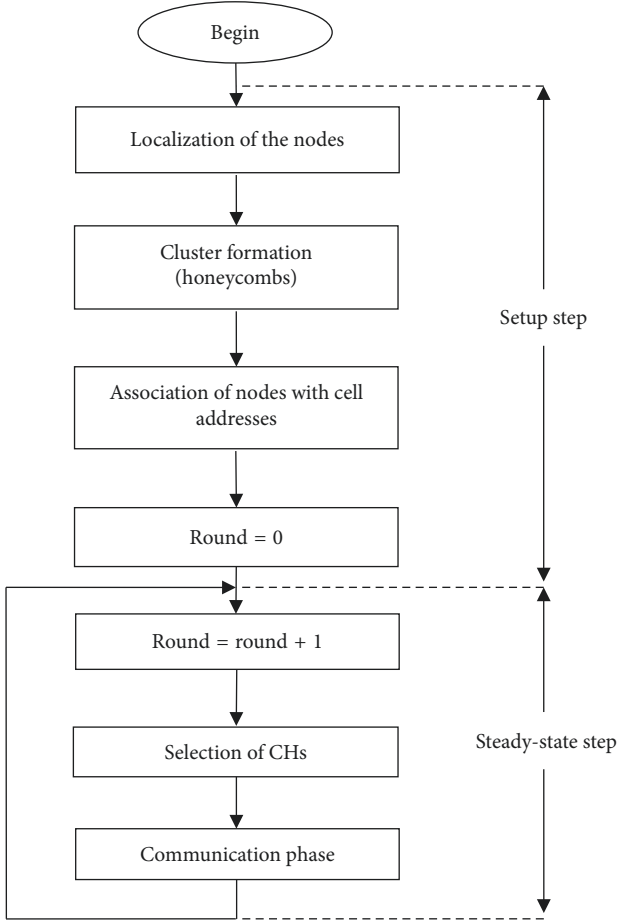


FIGURE 1: Flowchart of the protocol EHCA.

There are some assumptions about our WSN properties:

- (i) The nodes are static, and the sinks are stationary
- (ii) All nodes are homogenous and have the same resources (energy, processing, sensing, etc.)
- (iii) There are some anchor nodes in the network used to localize the nodes of our network
- (iv) The nodes are randomly distributed
- (v) The sensing field is circular and large scale with a high density of nodes
- (vi) There is at least one node in every hexagonal cell
- (vii) The nodes of the network sense data periodically

3.1. Setup Step

3.1.1. Localization of the Sensor Nodes. The localization is the operation that determines the coordinates of the various sensors and is used to identify the origin of the sensed data. In our network, the axes X and Y are crossing at the center of the network (M), and its coordinates are $(0, 0)$.

The localization is an indispensable operation to determine the capture position of the detected events. The use of the GPS technology is not an energy-efficient method to know the location of the sensor nodes [22, 23]; therefore, there are some efficient geometric methods to localize the nodes using some anchor nodes having known locations in the network. The most used methods are hyperbole method, trilateration, and triangulation [24, 25]. The determination of the distance between the nodes will be given by the following techniques: angle of arrival (AOA) [26], received signal strength indication (RSSI) [27], and time of arrival (TOA) [28]. For our protocol, we choose to use the signal strength (RSSI) to estimate the distance between nodes and the triangulation method to determine the locations of nodes in our network, and this method is used because of its simplicity of implementation in the presence of anchor nodes in the network.

3.1.2. Partition and Structure of the Network. The partition of the network into adjacent hexagons is the ideal way for clustering in WSN to preserve the remaining energy and extend the lifetime of the network [29]. The sensing area is a large-scale network with a high density of nodes; we divide the network into regular adjacent honeycomb clusters, and the clusters will be divided into a set of hexagonal cells with at least one node in a cell; thus, the network will be divided into a virtual grid of honeycombs and hexagonal cells.

Each hexagonal cell contains many nodes, and the transmission range of every node is equal to R with $R = e * \sqrt{13}$; e is the edge size of the cells. R is the longest distance between two adjacent cells (Figure 1); thus, the nodes of the adjacent cells can communicate with each other directly without any problem. We keep in the active state the node with the higher remaining energy, and we turn off the radio of the other nodes in the hexagonal cell to reduce the energy consumption.

We use the parameters f and t to specify the coordinates of the nodes in the network; from Figure 2, the values of f and t are

$$\begin{cases} f = \frac{3}{2} * e, \\ t = \frac{\sqrt{3}}{2} * e. \end{cases} \quad (1)$$

Every cluster contains n_r cell rings besides the cell in the center of the cluster; in total, there are S_c cells in every cluster, with S_c calculated as

$$S_c = 3n_r * (n_r + 1) + 1. \quad (2)$$

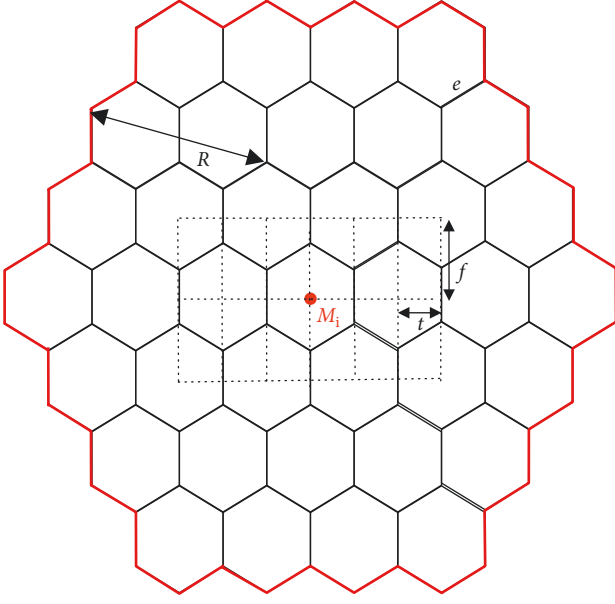


FIGURE 2: Structure of the honeycomb cluster with 3 cell rings.

In our protocol, the clusters are created depending on the position of the nodes and the location of the honeycomb centers. To associate the nodes of the network to the honeycomb clusters, we determine firstly the location of the honeycomb centers, and then every sensor node of the network joins the nearest honeycomb center to create the clusters.

The center of the sensing network M is the center of the first honeycomb cluster; there are N_r rings of the honeycomb clusters in our network, and the value of N_r depends on the size of the network; the virtual grid of the honeycomb clusters has to cover all the network. In Figure 3, we take $N_r = 3$ and $n_r = 2$; the first ring is in blue, the second one is red, and the third is orange. In every ring, we find six primary honeycomb centers (M_v), with v the identification of the cluster, and they are in red in Figure 3; the secondary honeycomb centers are in black and appear from the second ring of the clusters. Between two neighboring primary centers in the k^{th} ring there are $k - 1$ secondary centers; for instance, between the primary centers M_{19} and M_{22} in the third ring, there are 2 secondary centers M_{20} and M_{21} (Figure 3). The coordinates of the primary center M_v in the ring k are calculated as follows:

$$\begin{cases} Mx_v = k * r_1 * \cos(\alpha + (p - 1) * \beta), \\ My_v = k * r_1 * \sin(\alpha + (p - 1) * \beta), \end{cases} \quad (3)$$

where $v = k(3k + p - 4) + 1$ with $p = 1, 2, \dots, 6$ and r_1 is the distance between the centers of the adjacent honeycomb clusters. According to Figure 3, $\beta = 60^\circ$, and α and r_1 will be calculated as

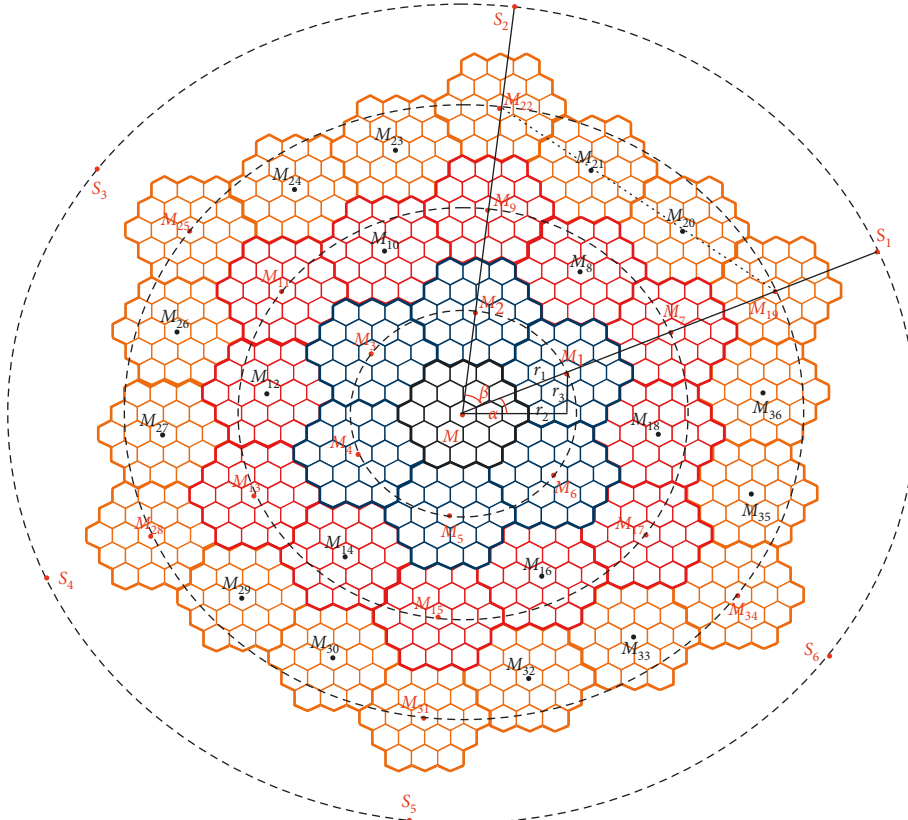


FIGURE 3: Architecture of the EHCA protocol.

$$r_1 = \sqrt{r_2^2 + r_3^2}, \quad (4)$$

with $r_3 = f * n_r$ and $r_2 = 3t * n_r + 2t$.

$$\sin \alpha = \frac{r_3}{r_1}. \quad (5)$$

The coordinates of the secondary honeycomb centers M_{v+m} between the primary honeycomb centers M_v and M_{v+k} located in the ring k are calculated as follows:

$$\begin{cases} Mx_{v+m} = Mx_v + \frac{m}{k} * (Mx_{v+k} - Mx_v), \\ My_{v+m} = My_v + \frac{m}{k} * (My_{v+k} - My_v). \end{cases} \quad (6)$$

With $m = 1, 2, \dots, k-1$, the secondary honeycomb centers M_{v+m} divide the segment (M_v, M_{v+k}) into k parts.

There are six sinks in our network, they are located in a special ring that contains the sinks, and it is after the last ring of the primary honeycomb centers (Figure 3); the coordinates of the sinks (Sx_p, Sy_p) with $p = 1, 2, \dots, 6$ are calculated with the same manner like the primary honeycomb centers. Algorithm 1 returns the location of the honeycomb centers and the sinks in our network.

Initialization:

$k \leftarrow 0$;

$N_r \leftarrow 0$;

$\beta \leftarrow 60^\circ$;

Begin

(1) **While** the sensing field is not fully cover by the virtual honeycombs **do**

(2) $k \leftarrow k + 1$;

(3) **For** $p \leftarrow 1$ to 6 **do**

(4) $v \leftarrow k * (3k + p - 4) + 1$; /*Determine the position of the primary centers*/

(5) $Mx_v \leftarrow k * r_1 * \cos(\alpha + (p - 1) * \beta)$;

(6) $My_v \leftarrow k * r_1 * \sin(\alpha + (p - 1) * \beta)$; /*Determine the position of the secondary centers from the second ring*/

(7) **If** $k \geq 2$ **do**

(8) **For** $m \leftarrow 1$ to $k - 1$ **do**

(9) $Mx_{v+m} \rightarrow Mx_v + m/k * (Mx_{v+k} - Mx_v)$;

(10) $My_{v+m} \rightarrow My_v + m/k * (My_{v+k} - My_v)$;

(11) **End for**

(12) **End if**

(13) **End for**

(14) **End While**

(15) $N_r \leftarrow k$; /*we get the number rings of the honeycomb clusters *//*Determine the position of the sinks*/

(16) **For** $p \leftarrow 1$ to 6 **do**

(17) $k \leftarrow k + 1$;

(18) $Sx_p \leftarrow k * r_1 * \cos(\alpha + (p - 1) * \beta)$;

(19) $Sy_p \leftarrow k * r_1 * \sin(\alpha + (p - 1) * \beta)$;

(20) **End for**

End

ALGORITHM 1: Location of the honeycomb centers and the sinks.

3.1.3. Cells Addressing in the Honeycomb Clusters. Erman et al. proposed in [30, 31] the algorithms of the nodes association with the cells in the honeycomb architecture and the addressing system of these cells; we use the same algorithms for each honeycomb cluster to assign the nodes to the cells and to get their addresses. The cell addressing system is used in the routing operation inside the honeycomb clusters; it is on the form of $[i, j]$ in every honeycomb cluster, with i the index of the cell rings and j the index of the cell in the ring i with $j = 0, 1, 2, \dots, i * 6 - 1$, and its value is incremented in the counter clockwise direction (Figure 4).

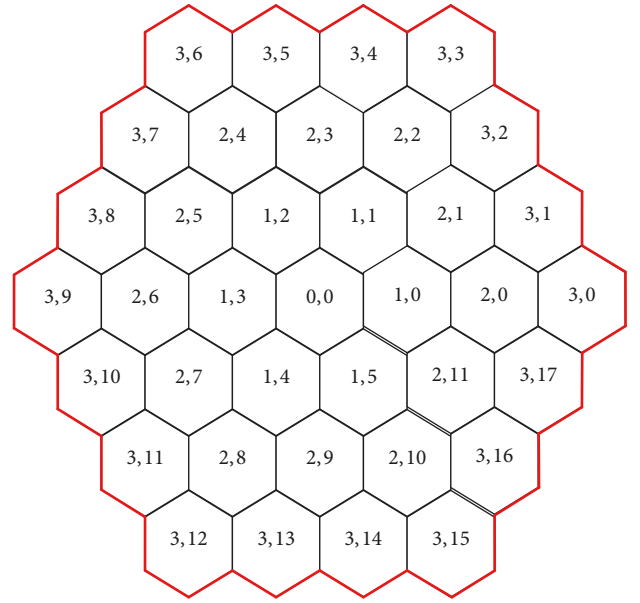


FIGURE 4: Cells addressing in the honeycomb clusters.

3.2. Steady-State Step. We partition this step into rounds; in the beginning of every round, we keep in the active state just the node with the higher residual energy in every cell; thus, we turn on its radio to become active, and we make in the sleep state the previous active node that lost its energy. Every round is also divided into two phases: CH selection phase and communication phase.

3.2.1. Cluster Head Selection Phase. The CH is responsible for aggregating the data collected in the honeycomb cluster to remove redundancy and return the aggregated data to the nearest sink. After every round, we change the location of the new CH in the honeycomb cluster, in order to balance the energy in intracuster. Then, we prolong the transmission range of the new CH to reach the sinks.

(1) **Location of the Cluster Head.** The location of the CH in our protocol will be chosen from the nodes that are located

in one of the six vertices of the honeycomb cluster. After every round, the location of the CH changes and forms in the cell at the other vertex of the cluster; therefore, the CH cell changes its location in this order: $[n_r, 0] \rightarrow [n_r, n_r * 3] \rightarrow [n_r, n_r] \rightarrow [n_r, n_r * 4] \rightarrow [n_r, n_r * 2] \rightarrow [n_r, n_r * 5] \rightarrow [n_r, 0]$.

To route the sensed data in the cluster, all the member nodes must be aware of the address of the CH cell; thus after the change of its location, the new CH sends an advertisement message (CH_AD) to all the nodes of the cluster and informs them about its cell address.

(2) *Balancing of Energy in Intracluster.* The nodes in the adjacent cells to the CH cell consume a high rate of energy, in contrast to the nodes in the cells farther from the CH cell that preserves a high value of remaining energy; in Figure 5, the light blue cells represent the high rate of energy consumption and the dark blue the low rates. The high traffic of data will deplete quickly the residual energy in the nodes located in the cells adjacent to the CH cell, which will cause a problem in the transmission of data, and the member nodes of the cluster will be unable to forward their sensed data to their target. The change of the location of the CH will be an excellent solution to the uneven energy balancing in the intracluster; thus, we will benefit from the high remaining energy of the nodes in the cells farther from the CH cell that becomes the adjacent nodes to the next CH cell after a round (Figure 5).

3.2.2. Communication Phase

(1) *Routing in Intracluster.* The nodes of the same honeycomb cluster sense data and disseminate them to the CH cell. The choice of the next hop cell will influence the data delivery latency and the energy consumption. We developed a new routing algorithm to minimize the hops count needed to forward data from the source cell where the data are sensed to the CH cell; this algorithm aims at choosing the optimal routing itinerary and saving the remaining energy of the sensor nodes; it will be divided into 3 main tasks:

- (i) Determine for every cell the list of adjacent cells: each sensor node diffuses a HELLO packet that includes its ID and the address of its cell to which it belongs. Sensor nodes in its transmission range that belong to the same honeycomb receive this HELLO packet and update their neighborhood tables with the ID and address included in the received packet. After the end of the neighborhood discovery operation, each node diffuses its ID, its cell address, its residual energy, and the data from its neighborhood table.
- (ii) Find the hop index of the cells: it starts from the adjacent cells of the CH cell and increases progressively to the farther cells (Figure 6).
- (iii) Transmission operation: The next hop cell will be chosen from the neighboring cells with the least hop index; this operation will be repeated at every hop to route the sensed data in the honeycomb cluster and reach the CH cell.

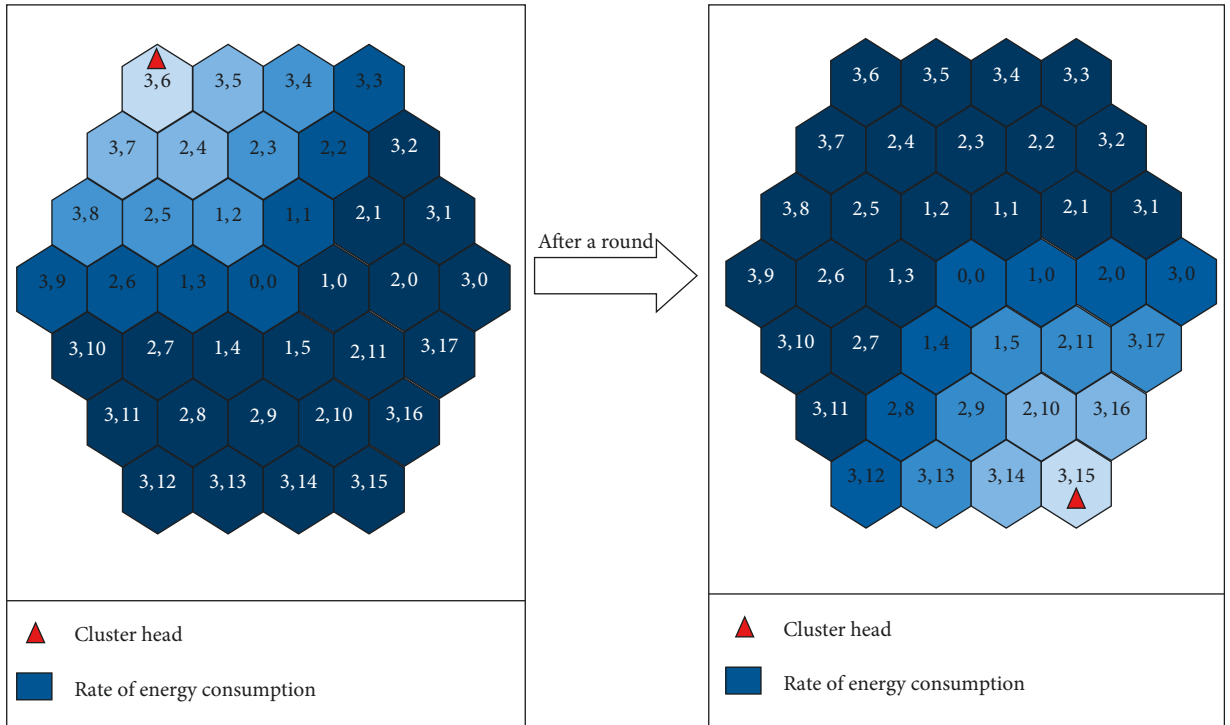
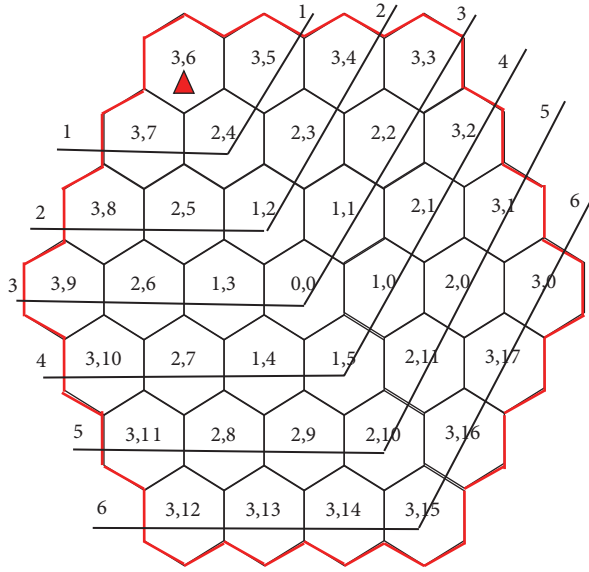


FIGURE 5: Energy balancing in the intracluster within a round.



▲ : Cluster head
— : Hop index

FIGURE 6: Hop index of the cells in the honeycomb cluster.

The hop index of the cells will change after every round according to the new CH cell direction, and the nodes from all cells forward the sensed data to this new CH cell. Algorithm 2 returns the hop index (L) of every cell in the cluster by comparing its address with the address of the CH cell.

Notation:

V, F : integer

Input

$[i, j]$: Address of a cell

$[H, J]$: Address of the CH cell

Result: The hop index cell (L)

Begin

```

(1) If  $(j \text{ DIV } i) < (J \text{ DIV } H)$  then
(2)    $V \leftarrow (J \text{ DIV } H) - 1$ ;
(3)    $L \leftarrow J - (j + ((n_r - i) * V))$ ;
(4) Else
(5)    $V \leftarrow (J \text{ DIV } H) + 1$ ;
(6)    $L \leftarrow (j + ((n_r - i) * V)) - J$ ;
(7) End if
(8) If  $L > n_r + i$  then /*the cells cannot have  $L$ 
    greater than  $n_r + i$ */
(9)    $F \leftarrow (L \text{ DIV } n_r + i)$ ;
(10)  If  $F > 1$  then
(11)    $j \leftarrow j + 6 * i$ ;
(12)  Repeat the instructions from 1 to 7 to
    find  $L$  of the new value of  $[i, j]$ ;
(13) Else  $L \leftarrow n_r + i$ ;
(14) End if
(15) End if
(16) Return  $L$ ;
End

```

ALGORITHM 2: The hop index of the cells in the honeycomb cluster.

Each cell is surrounded by six neighboring cells; from these cells, we choose the nearest one to the CH cell, which has the smallest hop index to transmit the sensed data; for example, in Figure 6, the node in the cell $[2, 9]$ is the source cell; it senses data and tries to forward them to the CH cell $[3, 6]$. In this case, the next hop cell will be one of their adjacent cells: $[2, 10]$, $[1, 5]$, $[1, 4]$, $[2, 8]$, $[3, 13]$, and $[3, 14]$; the hop indexes of these cells will be obtained by Algorithm 2 and their values are, respectively, 5, 4, 4, 5, 6, and 6. The minimum hop index is 4, and two cells have this value, the cells $[1, 5]$ and $[1, 4]$; we choose from them the cell that contains the node with the highest remaining energy to forward data, and it becomes the next hop cell; this operation is repeated, and the sensed data will be forwarded from one cell to another until they reach the CH cell (Algorithm 3).

Notation:

m, \min, power : integer

$\text{RE}(C)$: the remaining energy of the active node in the cell C

$[i', j']$: Address of the next hop cell

Input

$[i, j]$: Address of the source cell

$[H, J]$: Address of the CH cell

Initialization:

$\min \leftarrow \text{large value}$;

$\text{power} \leftarrow 0$;

Begin

```

(1)  $[i', j'] \leftarrow [i, j]$ ; /*the first next hop cell will be the
    source cell*/
(2) While  $[i', j']$  is not  $[H, J]$  do /*we stop when the next
    hop is the CH cell*/
(3)    $\text{ADJ} \leftarrow \text{Adjacent}(i', j')$ ; /*we get the list of the
    adjacent cells of  $[i', j']$  from the neighborhood table*/
(4)   For each cell  $C$  in  $\text{ADJ}$  do /*we search the next
    hop from the adjacent cells*/
(5)      $\text{index\_hop} \leftarrow \text{Level}([H, J], C)$  /*we get the
    hop index of the adjacent cell with the Algorithm 2*/
(6)     If  $\text{index\_hop} < \min$  then /*we compare the
    hop index of all adjacent cells*/
(7)        $\min \leftarrow \text{index\_hop}$ ;
(8)        $\text{power} \leftarrow \text{RE}(C)$ ;
(9)        $[i', j'] \leftarrow C$ ;
(10)    Else if  $\text{index\_hop} = \min$  and  $\text{RE}(C) > \text{power}$  then
    /* in the case of two adjacent cells with the same hop
    index we select the cell with the higher value of
    remaining energy to forward data */
(11)       $\text{power} \leftarrow \text{RE}(C)$ ;
(12)       $[i', j'] \leftarrow C$ ;
(13)    End if
(14)  End For
(15) forward data to  $[i', j']$ ;
(16) End While
End

```

ALGORITHM 3: Routing in the intracell.

Heinzelman et al. estimated in [5] the energy consumption needed to send n_b bits of a message from one node to another at d distance by the following equation:

$$E_{TX}(n_b, d) = E_{elec} * n_b + E_{amp} * n_b * d^2. \quad (7)$$

The energy consumed to receive the same message by a sensor node is

$$E_{RX}(n_b) = E_{elec} * n_b. \quad (8)$$

To run the receiver or transmitter circuitry, the node dissipates the energy E_{elec} , and the energy consumed by the transmission amplifier is E_{amp} .

The average distance between two nodes located in the adjacent hexagonal cells is the distance between the centers of these cells; thus in our case, $d = \sqrt{3} * e$, and the energy consumed to forward the data between two adjacent cells is

$$E_X(n_b, d) = 2 * E_{elec} * n_b + E_{amp} * n_b * 3 * e^2. \quad (9)$$

The routing of the sensed data in the intracluster from a source cell in the hop index L to the CH cell needs L transmissions and $L - 1$ receptions; thus, the cost of energy consumed will be calculated as

$$E_c = n_b * L * (E_{elec} + E_{amp} * 3 * e^2) + (L - 1) * E_{elec} * n_b. \quad (10)$$

(2) *Routing in Intercluster.* The CHs forward the data directly to the sinks; the energy consumed by this operation is related to the distance they are away from them. There are six sinks in our network (Figure 3); every CH node calculates the distance between it and these sinks and chooses the nearest one to transmit the data of the cluster. This will decrease the transmission delay and extend the battery lifetime of the concerned nodes. At the beginning of every round, the old sink linked to the cluster will stop receiving its data if there is another sink closer to the new CH node, so the sinks update the list of clusters that are attached to them.

4. Simulation and Results

4.1. Simulation Settings. The simulation is operated in a circular sensing field. Its radius is 600 m, and there are 1800 sensor nodes distributed in a random manner over the network; six sinks are deployed on the edge of the network, and their locations is determined according to Algorithm 1. The maximum transmission range of the nodes is 60 m; each round takes 100 seconds to finish the transmission of the gathering data from the nodes to the sinks; the IEEE 802.15.4 standard is chosen for the communication in our network. We use the ns2 simulator tool [32] to simulate and evaluate the performances of our protocol; the parameters used to evaluate the performance of our protocol are remaining of nodes alive, latency of data delivery, and percentage of successful delivery.

We execute the simulations many times to get the best results; the parameters used in the simulations are displayed in Table 1.

TABLE 1: Parameters of the simulation.

Parameters	Values
Node distribution	Random
Initial energy in battery	5 Joules
Number of nodes in the network	1800
Radio transmission range	60 m
Coverage radius of sensing	40 m
Bandwidth	0.25 Mbps
Average size of data	50 bytes
Data processing rate	50 Mbps
E_{elec}	50 nJ/bit
E_{amp}	100 pJ/bit/m ²
Critical remaining energy	0.01 J
Mac layer standard	IEEE 802.15.4

4.2. Results and Comparisons. The number of the cell rings in every cluster (n_r) is an important parameter that specifies the size and the number of the honeycomb clusters in the network. Figure 7 shows the average of energy consumption per round in a network of 1800 nodes; its value changes according to n_r , and it is obvious that the performance of our algorithm raises when n_r is between 2 and 4, less than 2 the number of the clusters become big with small cluster size; thus, there will be a big number of CHs, which aggregate small data and consume a high amount of energy. Beyond 4 cell rings, the size of the clusters will be larger and the network will contain a limited number of CHs; the nodes have to transmit their data farther to reach the CHs, which explain the raise in the energy consumption with the increase in n_r . In the next simulations, we take n_r equal to 3 for optimum results.

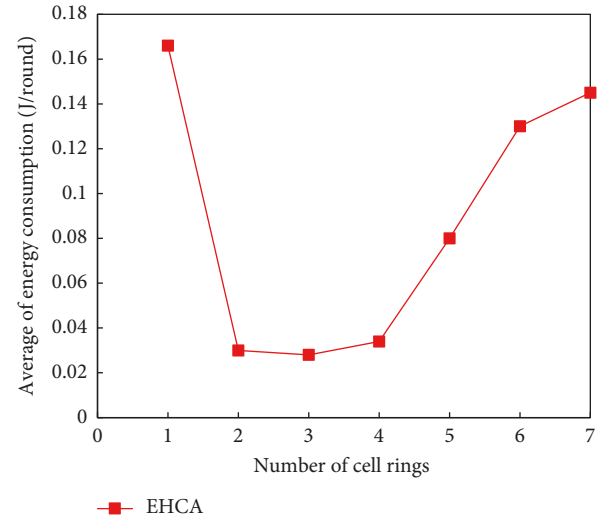


FIGURE 7: Average of energy consumption per round according to the number of cell rings.

There are 3 performance metrics used to evaluate our protocol:

- (i) Number of nodes alive: the energy of the sensor nodes will be depleted in the execution of the different operations of sensing, transmission, calculating, etc.; after several rounds, only a few nodes are alive in the network
- (ii) Average delivery latency: it is the average time taken for the transmission of the sensed data from the nodes to the sink

- (iii) Ratio of successful data delivery: it represents the percentage of the data who successfully reach the sinks compared to the total sensed data

The evolution of the performance metrics in the Figures 8–10 are presented in terms of the time slots; the packet of the sensed data is supposed to take one time slot to reach the sink. The performance of our protocol EHCA will be compared with 3 other existing protocols; the first one is HEER the hierarchical protocol in which the creation of clusters is done only once and only the CHs nodes that change after each round as in our EHCA protocol. The second protocol is GAF the geographical protocol that divides the network into rectangular adjacent cells. Finally, PEGASIS the hierarchical protocol that groups the network nodes into a long chain, based on the principle that a node can only communicate with the nearest neighboring node, and just one node will communicate with the base station.

The number of nodes alive decreases gradually for the four protocols; in Figure 8, after 100 time slots, the number of dead nodes is still approximately the same for all protocols with the death of a few number of nodes. With time progresses, the nodes deplete their energy in the different operations (communication, sensing, etc.), GAF loses its nodes alive quickly after 900 time slots, and it is the worst protocol in terms of energy consumption. The gap between it and our protocol reaches almost 900 nodes alive after 1900 time slots; the superiority of our protocol in terms of energy consumption will be explained by the honeycomb clustering and the energy balancing in intracluster that saves the battery energy of the nodes. It is obvious that the EHCA protocol performs the best result in terms of the number of nodes alive in the network.

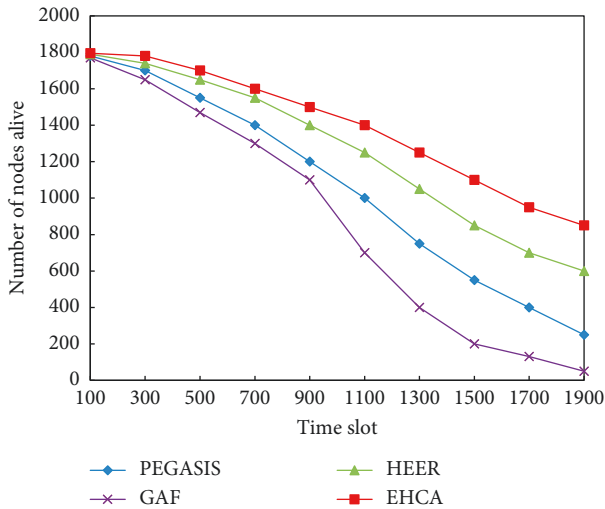


FIGURE 8: Number of nodes alive in the network per time slot.

The average data latency of the simulated protocols is presented in Figure 9; in the beginning of the simulation, the sensed data in HEER, PEGASIS, and EHCA protocols take a long time to reach the sink compared to GAF. The reason of this delay is that, in the hierarchical protocols, the CHs wait to receive the sensed data from the member nodes, and then they merge and aggregate them before sending them to

the sink. After 900 time slots, the average delivery latency of GAF exceeds the other protocols, the death of some nodes of the network will destroy some routing itineraries in GAF protocol, but its impact over the three other protocols is limited, thanks to the hierarchical routing in these protocols, which is based on the CHs that are re-elected from the member nodes if they are exhausted.

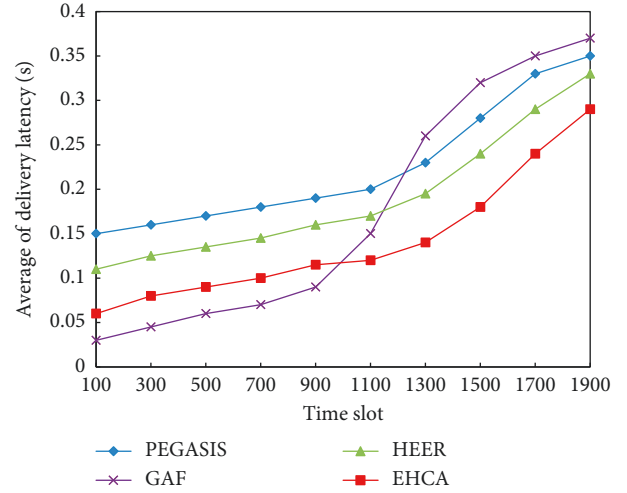


FIGURE 9: Average delay of data delivery to the sink.

Figure 10 shows that our protocol outperforms the other protocols in terms of the ratio of the successful data delivery to the sink. Contrarily, in GAF, the probability of the nodes to fail to reach their sensed data to the sink is higher compared to the other protocols. The big number of hops between the sensor nodes and the sink increases the probability of losing a lot of sensed data in the routing operation. Before 300 time slots, all the protocols get approximately the same ratio, but the gap between them gets bigger after 900 time slots; the energy of a lot of nodes is depleted, which hampers the routing of data to the sinks. In the end of the simulation, the percentage of successful delivery reaches a difference of 30% between GAF and our protocol.

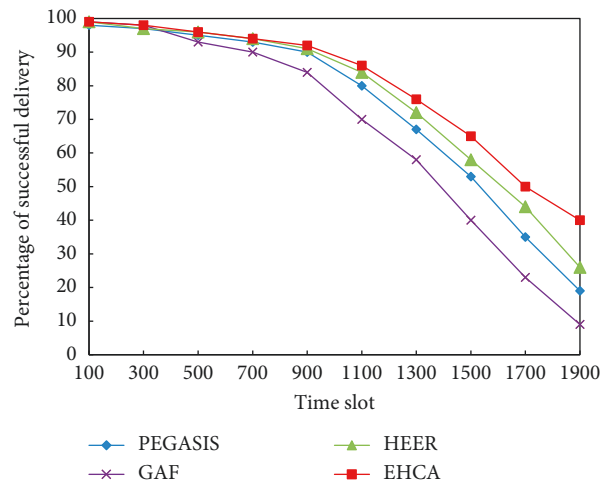


FIGURE 10: Ratio of successful data delivery to the sink.

5. Conclusion

In the protocol presented in this article, the nodes are localized, and we associate them with a virtual grid of honeycombs and hexagonal cells. The clustering by adjacent and uniform honeycombs with multihop routing in intracenter saves the remaining energy of the nodes, which do not have to transmit their sensed data for a long distance to reach the CH but forward them to a node in the neighboring cells and therefore save their energy. The change of the location of the CH cell will mitigate the load on its neighboring cells, and we will be benefited from the energy on its farther cells that have a large amount of the residual energy, achieving this way the energy balancing inside the honeycomb clusters of the network. The comparison between our protocol and the other protocols shows the superior performance of ours in terms of the number of nodes alive, the latency of data delivery, and the percentage of successful data delivery to the sinks.

In our future work, we plan to create a new method of gathering data from the network based on the multiple mobile agents. The agents have to follow optimal itineraries to lower the latency of data gathering and maximize the lifetime of the network.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] M. Dener, "A new energy efficient hierarchical routing protocol for wireless sensor networks," *Wireless Personal Communications*, vol. 101, no. 1, pp. 269–286, 2018.
- [2] Z. Manap, B. M. Ali, C. K. Ng, N. K. Noordin, and A. Sali, "A review on hierarchical routing protocols for wireless sensor networks," *Wireless Personal Communications*, vol. 72, no. 2, pp. 1077–1104, 2013.
- [3] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Maui, HI, USA, January 2000.
- [4] M. Yu, K. Leung, and A. Malvankar, "A dynamic clustering and energy efficient routing technique for sensor networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 3069–3079, 2007.
- [5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [6] L. J. Villalba, L. A. Orozco, T. A. Cabrera, and J. C. Abbas, "Routing protocols in wireless sensor networks," *Sensors*, vol. 9, no. 11, pp. 8399–8421, 2009.
- [7] S. Lindsey and C. Raghavendra, "PEGASIS: power-efficient gathering in sensor information systems," in *Proceeding of IEEE Aerospace Conference*, Big Sky, MT, USA, March 2002.
- [8] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924–935, 2002.
- [9] A. Manjeshwar and D. P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS-01)*, San Francisco, CA, USA, April 2001.
- [10] A. Manjeshwar and P. D. Agrawal, "APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," in *Proceedings of 16th International Parallel and Distributed Processing Symposium*, Fort Lauderdale, FL, USA, April 2002.
- [11] G. S. Arumugam and T. Ponnuchamy, "EE-LEACH: development of energy-efficient LEACH protocol for data gathering in WSN," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 76, 2015.
- [12] D. Yi and H. Yang, "HEER-a delay-aware and energy-efficient routing protocol for wireless sensor networks," *Computer Networks*, vol. 104, pp. 155–173, 2016.
- [13] C. P. Subha and S. Malarkkan, "H-CERP: energy efficient scheme for industrial WSN applications," *Wireless Personal Communications*, vol. 94, no. 4, pp. 1937–1950, 2016.
- [14] P. Kuila and P. K. Jana, "Energy efficient clustering and routing algorithms for wireless sensor networks: particle swarm optimization approach," *Engineering Applications of Artificial Intelligence*, vol. 33, pp. 127–140, 2014.
- [15] M. Juneja and S. K. Nagar, "Particle swarm optimization algorithm and its parameters: a review," in *Proceedings of IEEE International Conference on Control, Computing, Communication and Materials (ICCCCM)*, Allahabad, India, October 2016.
- [16] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, 2017.
- [17] J.-Y. Chang and P. Ju, "An efficient cluster-based power saving scheme for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 172, 2012.
- [18] J. Lloret, M. Garcia, D. Bri, and J. Diaz, "A cluster-based architecture to structure the topology of parallel wireless sensor networks," *Sensors*, vol. 9, no. 12, pp. 10513–10544, 2009.
- [19] A. Mehmood, S. Khan, B. Shams, and J. Lloret, "Energy-efficient multi-level and distance-aware clustering mechanism for WSNs," *International Journal of Communication Systems*, vol. 28, no. 5, pp. 972–989, 2013.
- [20] Y. Yu, D. Estrin, and R. Govindan, "Geographical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks," Technical Report UCLA-CSD TR-01-0023, UCLA Computer Science Department, Los Angeles, CA, USA, 2001.
- [21] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01)*, pp. 70–84, Rome, Italy, July 2001.
- [22] N. A. Alrajeh, M. Bashir, and B. Shams, "Localization techniques in wireless sensor networks," *International Journal of Distributed Networks*, vol. 9, no. 6, article 304628, 2013.
- [23] K. A. Paul and T. Sato, "Localization in wireless sensor networks: a survey on algorithms, measurement techniques,

- applications and challenges,” *Journal of Sensor and Actuator Networks*, vol. 6, no. 4, p. 24, 2017.
- [24] A. Boukerche, E. Nakamura, E. F. Nakamura, and A. A. F. Loureiro, “Localization systems for wireless sensor networks,” *IEEE Wireless Communications*, vol. 14, no. 6, pp. 6–12, 2007.
 - [25] Z. Fang, Z. Zhao, X. Cu, D. Geng, L. Du, and C. Pang, “Localization in wireless sensor networks with known coordinate database,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, no. 1, article 901283, 2010.
 - [26] M. Li and Y. Lu, “Angle-of-arrival estimation for localization and communication in wireless networks,” in *Proceedings of 16th IEEE European Signal Processing Conference*, Lausanne, Switzerland, August 2008.
 - [27] J. Xu, W. Liu, F. Lang, Y. Zhang, and C. Wang, “Distance measurement model based on RSSI in WSN,” *Wireless Sensor Network*, vol. 2, no. 8, pp. 606–611, 2010.
 - [28] E. Xu, Z. Ding, and S. Dasgupta, “Wireless source localization based on time of arrival measurement,” in *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Dallas, TX, USA, March 2010.
 - [29] D. Wang, L. Lin, and L. Xu, “A study of subdividing hexagon-clustered WSN for power saving: analysis and simulation,” *Ad Hoc Networks*, vol. 9, no. 7, pp. 1302–1311, 2011.
 - [30] A. T. Erman and P. Havinga, “Data dissemination of emergency messages in mobile multi-sink wireless sensor networks,” in *Proceeding of the 9th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Juan les-Pans, France, June 2010.
 - [31] A. T. Erman, A. Dilo, and P. Havinga, “A virtual infrastructure based on honeycomb tessellation for data dissemination in multi-sink mobile wireless sensor networks,” *Eurasip Journal on Wireless Communications and Networking*, vol. 2012, p. 17, 2012.
 - [32] Network Simulator (ns2), <http://www.isi.edu/nsnam/ns>.

