Research Article
Sensors Grouping Hierarchy Structure for Wireless Sensor Network

Ammar Hawbani, 1 Xingfu Wang, 1 Saleem Karmoshi, 1 Lin Wang, 1 and Naji Husaini 2

1 School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China
2 Hefei University of Technology, Hefei, Anhui, China

Correspondence should be addressed to Xingfu Wang; wangxfu@ustc.edu.cn

Received 24 October 2014; Revised 28 March 2015; Accepted 13 April 2015

Academic Editor: Maode Ma

Copyright © 2015 Ammar Hawbani 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.

There are many challenges in implementation of wireless sensor network systems: clustering and grouping are two of them. The grouping of sensors is computational process intended to partition the sensors of network into groups. Each group contains a number of sensors and a sensor can be an element of multiple groups. In this paper, we provided a Sensors Grouping Hierarchy Structure (GHS) to split the nodes in wireless sensor network into groups to assist the collaborative, dynamic, distributed computing and communication of the system. Our idea is to partition the nodes according to their geographical maximum covered regions such that each group contains a number of nodes and a number of leaders. To evaluate the performance of our proposed grouping structure, we have implemented a Grouped based routing and Grouped based object tracking. The proposed grouping structure shows a good performance in energy consumption and energy dissipation during data routing and it generates a little redundant data during object tracking.

1. Introduction

Like other distributed systems, WSNs are subject to a variety of unique constraints and challenges such as restricted sensing and communication ranges as well as limited battery capacity [1, 2]. These challenges affect the design of WSN [3] and bring some issues such as coverage, connectivity, network lifetime, self-managing, data aggregation, energy dissipation and energy balancing, clustering, and grouping [4, 5]. In WSNs, the node contains three main subsystems: the sensing subsystem contains one or more physical sensor devices and one or more analog-to-digital converters as well as the multiplexing mechanism to share them. The processor subsystem executes instructions pertaining to sensing, communication and self-organization [2]. The communication subsystem contains the transmitter and receiver for sending or receiving information. Due to limited range of communication, establishing the direct connection between a sensor and the base station may make the nodes transmit their messages with such a high power that their resources could be quickly depleted. Thus, the collaboration of nodes ensures the communications between distant nodes and base station. In this way, the intermediate nodes transmit messages so that a path with multiple links or hops to the base station is established [2, 6, 7]. However, the collaborative working of nodes is more critical and more complicated because of WSN natural and challenges like energy dissipation, energy balancing, location tracing, and latency. The sensors work collaboratively in many applications in our daily lives, including data collection, military applications, monitoring, and space applications [8]. For obtaining a stable collaborative work among sensors, the nodes must be able to organize themselves in structures (i.e., hierarchical structure) such that the objectives of network are achieved using the minimum cost of communication. Generally, the clustering is the most common used structure for WSNs. The clustering techniques for WSNs can be classified based on the overall network architectural and operation model and the objective of the node grouping process including the desired count and properties of the generated clusters. The research community has pursued it widely in order to achieve the network scalability objective [9]. Many of clustering algorithms [10–12] focused mainly on how to yield stable clusters with node reachability and route in environments using mobile nodes without much concern about
critical design goals of WSNs such as network connectivity and coverage [9]. During the recent years, there are also many clustering algorithms designed especially for WSN, most of them mainly focused on energy dissipation, energy-balancing, and so forth.

In the literature, many protocols to reduce energy consumption have been proposed. The Low Energy Adaptive Clustering Hierarchy (LEACH) protocol [13, 14] is a cluster-based routing protocol for WSN networks that perform load balancing and ensure scalability and robustness by routing via cluster-heads and implement data fusion to reduce the amount of information overhead [15]. LEACH utilizes a randomized rotation of local cluster head to evenly distribute the energy load among the sensors in the network. After LEACH, the Power-Efficient Gathering in Sensor Information System (PEGASIS) [16] comes with new improvements: in PEGASIS each node communicates only with a nearby neighbor in order to exchange data, but LEACH uses single-hop routing in which each sensor node transmits information directly to the cluster-head or the sink. Regarding clustering in WSN, [17] surveys different energy efficient clustering protocols for heterogeneous wireless sensor networks and compares these protocols on various points like location awareness, clustering method, heterogeneity level, and clustering attributes. Moreover [18] presented efficient request-oriented coordinator methods for hierarchical sensor networks.

On the other hand, the traditional structures in WSNs can be categorized into data centric and location based. References [19, 20] provided two routing protocols based on data centric. In [19], Sensor Protocols for Information via Negotiation (SPIN) uses the data negotiation scheme among sensor nodes to decrease data redundancy and save energy. Direct Diffusion [20] is another data-centric routing protocol. The data generated by sensor nodes is named attribute-value pairs. When a sink node queries a certain type of information, it will send a request and the sensed data can be aggregated and then be transmitted back to the sink node. Location-based routing protocols can get the information of nodes location via GPS or any estimation algorithms based on received signal strength. Once the location information is known, the consumption of energy could be largely minimized using power control techniques [21, 22]. Based on theoretical analysis and numerical illustration under different energy and traffic models the [23] Distance-based Energy Aware Routing (DEAR) proposed an efficiently reducing and balancing of energy consumption in WSNs. During the routing process, DEAR treats a distance distribution as the first parameter and the residual energy as the secondary parameter. In the aims at maximizing the network lifetime and minimizing the energy consumption, the Two-Level Cluster base Protocol (TLCP) [24] organizes sensor nodes into clusters and forms a cluster among the cluster heads. In TLCP each cluster head transmits its data to the header of this cluster instead of transmitting directly to the far away base station and only the header can transmit data directly to the base station. The authors in [25] describe a routing protocol for wireless sensor networks based on the inclusion of routing information in the packets when minimum cost forwarding method is used. The routing table on BS is formed in the network setup phase and updated after any change in network topology reported by sensor nodes. CCM (Chain-Cluster based Mixed routing) is explained in [26]; this algorithm makes a full use of the advantages of LEACH [13] and PEGASIS [16] and provided an improved performance. CCM algorithm divides the WSN into a few chains and runs in two stages. In the first stage, sensor nodes in each chain transmit data to their own chain head node in parallel, using an improved chain routing protocol. In the second stage, all chain head nodes group as a cluster in a self-organized manner, where they transmit fused data to a voted cluster head using the cluster based routing.

As shown in Figure 1, we assumed that the interested field is covered with the minimum number of nodes such that
the overlapped regions between the nodes are minimized and the nodes are expanded to cover the interested field completely. As well as, we assumed that the nodes have the same communication range indicated by the dotted circle. With such assumptions, unfortunately, if the CH nodes have the same communication range, the static clustering or dynamic clustering will not perform as efficient as peer-to-peer network.

This paper intended to develop and design a grouping algorithm to be used for partitioning the nodes into groups according to the maximum covered regions in the field. We will see the impact of this model in

(i) data routing and energy saving (Section 4),
(ii) object tracking and energy balancing (Section 5),
(iii) avoiding data redundancy (Section 6).

The difference between clustering and grouping is simply as below:

(i) Clustering is performed by assigning each node to a specific cluster; that is to say, each node belongs to only one cluster [8]. There is only one CH in each cluster.

(ii) Grouping model is to divide the network into groups such that each node can belong to more than a group in the same time. There is more than one leader for the same group.

The rest of this paper is organized as follows. In Section 2, the grouping model and grouping algorithm are explained. The suggested network model is described in Section 3. In Section 4, we have proposed a very simple strategy for data routing based GHS. In Section 5, based on GHS, an object tracking strategy is proposed. A mechanism for avoiding data redundancy is explained in Section 6. The simulation results are shown in Section 7. Section 8 concludes this work.

2. Grouping Structure

Our main idea is to partition network's nodes into groups according to their geographical maximum covered regions in such way that each group contains a number of nodes and a number of leaders; that is, the network in Figure 2(a) contains 8 maximum covered regions (indicated by small filled circles and given a number from small filled circle 1 to small filled circle 8). That is to say, this network can be partitioned into eight groups. In Figure 2(b) the graph vertices are shown (small filled circle 1 to small filled circle 8) and each vertex represents a group of sensors; that is, the group (small filled circle 1) contains four sensors, namely, 1, 2, 3, and 4. Two groups are said to be adjacent if they contain one or more sensors in common; that is, groups ((small filled circle 1) and (small filled circle 2)) are adjacent since they have two sensors in common, namely, 3 and 4. The common sensors are called the leader of adjacent groups and they represent the edge of network graph.

In the following, we will address the grouping problem using a simple mathematical model. Here we can present the network as a list of vectors collected by all sensors in the field. The network of n sensors is represented by a square matrix \((n \times n)\). Consider

\[
A^\ast = \begin{bmatrix}
1 & 1 & 0 \\
1 & 1 & 0 \\
0 & 1 & 1 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
a_{ij} = \begin{cases}
1, & s_i \text{ and } s_j \text{ are overlapped}, \\
1, & i = j, \\
0, & s_i \text{ and } s_j \text{ are not overlapped}.
\end{cases}
\]

Figure 2: 12 nodes deployed randomly.
For example, the network in Figure 2 is represented by a square matrix (12 × 12):

\[
A^* = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 
\end{bmatrix}.
\]

We can find the maximum covered regions by partitioning \(A^*\) into square submatrices \(\{A_1, A_2, A_3, \ldots \}\) such that all elements of submatrix \(a_{ij} = 1\), as well as the subsquare matrix, contain the maximum number of rows and columns. Each square submatrix represents a group of sensors. For example, the network in Figure 2 is partitioned into eight square submatrices listed as below:

\[
A_1 = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 
\end{bmatrix}_{g_1=(1,2,3,4)},
\]

\[
A_2 = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 
\end{bmatrix}_{g_2=(3,4,9)},
\]

\[
A_3 = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 
\end{bmatrix}_{g_3=(4,8,9)},
\]

\[
A_4 = \begin{bmatrix}
1 & 1 \\
1 & 1 
\end{bmatrix}_{g_4=(6,8)},
\]

\[
A_5 = \begin{bmatrix}
1 & 1 \\
1 & 1 
\end{bmatrix}_{g_5=(9,10)},
\]

\[
A_6 = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 
\end{bmatrix}_{g_6=(10,11,12)}.
\]

If WSN is connected then \(A^*(n \times n)\) can be partitioned into square submatrices \(\{A_1, A_2, A_3, \ldots, A_i\}\), where \(i\) is an integer and \(1 \leq i \leq n - 1\).

We divide the nodes into a set of groups dynamically such that \(G^* = \{G^1, G^2, G^3, \ldots\}\), where \(i\), \(j\), and \(k\) are integers indicating the number of sensors in the groups \(G^1, G^2, G^3\), respectively. Note that the term of groups is different from the term clusters.

A group \(G^k \in G^*\) contains \(1 + k(k - 1)\) subareas and among them there will be one area which is \(k\)-covered, \(k\) areas which are \(1\)-covered, \(k\) areas which are \(2\)-covered, and \(k\) areas which are \(k - 1\) covered. Note that when an object moves in \(x\)-covered area \((x > 1, x \in Z)\) it will be detected by \(x\) sensors. Hence, more traffic will be generated which leads to more redundant data and more energy consumption in the network. In the range of any sensor \(s \in G^k\) there are \(1 + k(k - 1)/2\) subareas, among them one area is \(k\)-covered, 1 area is \(1\)-covered, 2 areas are \(2\)-covered, 3 areas are \(3\)-covered, …, and \(k - 1\) areas are \(k - 1\) covered; that is, see Figure 3.

A group \(G^k \in G^*\) contains \(R(G^k) = 1 + k(k - 1)\) regions (i.e., Figure 3(b)). We can count the \(R(G^k)\) easily by considering the number of regions in \(G^k\) with no repetition; that is to say,

\[
R(G^k) = \sum_{j=0}^{k} R(G^k) = \sum_{j=0}^{k} R(s_j),
\]

where \(R(s_j)\) is the number of regions within the range of \(s_j\), and it is easy to see that the number of areas inside the range of node \(s_j\) is satisfying the recursive relation:

\[
R(s_j \in G^k) = \begin{cases} 
 f(k) = k + f(k - 1) - 1, \\
 f(1) = 1.
\end{cases}
\]

We can solve this equation using generation functions and obtain

\[
R(s_j \in G^k) = 1 + \frac{k(k - 1)}{2},
\]

\[
R(s_j \in G^k) = 1 + \left(\begin{array}{c} k \\ 2 \end{array}\right).
\]
By removing the repetition from (4), we can get

\[
R(G^k) = 1 + \binom{k}{2} + (k-1) + (k-2) + \cdots + (k-(k-1)),
\]

\[
R(G^k) = 1 + \binom{k}{2} + \sum_{i=1}^{k-1} (k-i) = 1 + \frac{k!}{2!} + \frac{k(k-1)}{2} = 1 + k(k-1).
\]

Among \(R(G^k)\) there will be one area which is \(k\)-covered, \(k\) areas which are \(1\)-covered, \(k\) areas which are \(2\)-covered \ldots, and \(k\) areas which are \(k-1\) covered. Let \(C_{x,k} (x \leq k)\) be the notation for \(x\)-covered region in \(R(G^k)\) and let \(\Phi(C_{x,k})\) be the number of \(C_{x,k}\) in \(R(G^k)\); hence \(\Phi(C_{0,k}) = 0, \Phi(C_{1,k}) = 1, \) and, generally, \(\Phi(C_{x,k}) = k\), where \((k-1) \geq y \geq 1\) and then it is easy to see that

\[
R(G^k) = \Phi(C_{0,k}) + \Phi(C_{1,k}) + \Phi(C_{2,k}) + \cdots + \Phi(C_{x,k}) = 1 + k(k-1).
\]

Furthermore, among the \(R(s_j \in G^k)\) there will be one area which is \(k\)-covered, 1 area which is \(1\)-covered, 2 areas which are \(2\)-covered, 3 areas which are \(3\)-covered \ldots, and \(k-1\) areas which are \(k-1\) covered; that is, see Figure 3(a). We denoted to the \(x\)-covered regions in the range of node \(s_j\) by \(C_{x,j}\) \((x \geq 0)\), and let \(\theta(C_{x,j})\) be the number of \(C_{x,j}\) regions; then, for any \(s_j \in G^k\), the \(\theta(C_{x,j}) = i\), where \((k-1 \geq i \geq 0)\). We can obtain that \(\theta(C_{0,j}) = 0\) and \(\theta(C_{k,j}) = 1\). It is obvious that

\[
R(s_j \in G^k) = \theta(C_{0,j}) + \theta(C_{1,j}) + \theta(C_{2,j}) + \cdots + \theta(C_{k,j}) = \theta(C_{k,j}) + 0 + 1 + 2 + 3 + \cdots + (k-1) = 1 + \frac{k(k-1)}{2}.
\]

Therefore, we can obtain the number of regions \(R(s_j \in G^*(s_j))\), where \(G^*(s_j)\) is the set of associated groups of \(s_j\) such that \(G^*(s_j) = \{G^* \mid G^* \in G^*\}\) and \(s_j \in G^*\). Let \(Y = |G^*(s_j)|\); then

\[
R(s_j \in G^*(s_j)) \leq \left( \sum_{G^* \in G^*} R(s_j \in G^*) \right) - (Y-1).
\]

Any object \(o_j \in M\) moves in a group \(G^k \in G^*\), its location should be detected within the group range with respect to the Probability space \(\Omega(G^k) = \{C_{1,k}, C_{2,k}, \ldots, C_{k,k}\}\). However, any object \(o_j \in M\) moves in the range of node \(s_j \in G^*(s_j)\); its location should be detected with respect to the Probability space \(\Omega(s_j \in G^*(s_j)) = \{c_{1,j}, c_{2,j}, \ldots, c_{k,j}\}\).

This grouping model has advantages since it makes the WSNs stress-free in many aspects such as data routing and data redundant avoiding. Below, we will provide a distributed grouping algorithm (Algorithm 1), which is executed inside the processing unit of the sensor node. However, it utilizes the information of nearby sensors to find the maximum covered regions. The symbols used for grouping algorithms are listed in Table 1.

Algorithm 1 shows how the sensor node \(s_j\) aware its associated groups \(G^*(s_j) = \{G^* \mid G^* \in G^*\}\) and \(s_j \in G^*\).
Find $G^*(s_i)$
Input: $s_i, S$
Output: $G^*(s_i)$ the associated group(s) of $s_i$

(1) $V_i ← s_i$. Vector
(2) for (int $k ← 0; k < |F_i|; k++$)
(3) {
(4) $V_k ← F_i[k]$;
(5) $s_k ← V_k[k]$;
(6) if ($|V_k| = 2$)
(7) {
(8) $G^k ← V_k$
(9) $G^*(s_k) += G^k$
(10) }
(11) else
(12) {
(13) for (int $m ← k + 1; m < |F_i|; m++$)
(14) {
(15) $V_m ← F_i[m]$;
(16) if ($s_k ∈ V_m$)
(17) {
(18) $G^{V_k∩V_m} ← V_k ∩ V_m$
(19) $G^*(s_k) += G^{V_k∩V_m}$
(20) }
(21) }
(22) }
(23) }

Algorithm 1: Grouping algorithm.

By analyzing Algorithm 1, we can predict the resources (i.e., memory, communication bandwidth, and energy) it requires. Assume that $n = |F_i|$ is the input size (the number of filtered vectors); then it is easy to compute the time complexity of Algorithm 1:

$$T(n) = (n−1) + (n−2) + (n−3) + \cdots + 3 + 2 + 1,$$

$$T(n) = \sum_{j=1}^{n} (n−j),$$

$$T(n) = \frac{1}{2}n(n−1).$$

The worst-case running time is

$$T(n) = \theta(n^2).$$

3. Network Model

We model the Grouped based network using a weighted graph $G$, where graph vertices correspond to groups and graph edges correspond to leader sensors. By “leader sensors,” we mean the sensors that belong to more than one group. Two groups are to be adjacent if they have one sensor or more in common.

We consider a statics Graph $G = (V, E, \mu)$ to represent the topology of sensors deployed in a sensing field, where $V$ indicates the vertices, $E$ indicates the edges, and $\mu$ indicates the weight function where $\mu : E → \mathbb{R}^+$ supplies the distance between adjacent vertices in $E$. Here each vertex in $V$ is represented by a group of sensors. Note that $V$ is different from previous approaches where each sensor node represents a vertex.

We assume that each sensor has a unique identifier (ID) and all sensors are aware of their geographical location and $\mu(v, v) = 0$ for any node $v ∈ V$. We assume that $G$ is connected; that is, there is a path of nodes connecting any pair of nodes in the network.

The nodes organize themselves into groups according to Algorithm 1 such that $G^* = \{G^i, G^j, G^k, \ldots\}$. A group $G^x ∈ G^*$ contains a number of sensors, that is, $G^x = \{s_1, s_2, \ldots, s_j\}$, $|G^x| = v$, and multiple number of leaders. The set of groups $G^*(s_j)$ is called the associated groups of $s_j$, such that $G^*(s_j) = \{G^x | G^x ∈ G^* and s_j ∈ G^x\}$. The leaders of groups are not randomly chosen (Table 2). However, the node can be a leader if and only if it belongs to more than one group; that is, in Figure 2, sensor 7 cannot be a leader since it belongs to only one group.

4. Data Routing

In this section we explain a simple data routing based on grouping model. When a source node has data to send, it first started by finding the minimum distance group to the base station (Algorithm 2) and then from the selected group chooses the next hop to forward the data. The next hop can be selected according to either minimum distance to the next target node or maximum energy of target node. After
Table 1: The symbols used for grouping algorithm.

| Symbol | Meaning |
|--------|---------|
| $S$    | A set of sensors. |
| $D$    | Euclidean distance. |

The list of neighbor nodes of $s_i$. Here we call $V$ the vector of sensor. $|V|$ is the number of sensor inside $V_i$. $D = \sqrt{(x_{s_i} - x_{s_j})^2 + (y_{s_i} - y_{s_j})^2}$

∀$s_j \in S$ do:

| $V_i$ | The list of neighbor vectors for $s_i$. |
|-------|--------------------------------------|
|       | ∀$s_j \in V_i$ do: |
|       | find $V_j$ and then add $V_j$ to $N_i$ by calling: |
|       | $N_i.add(V_j)$. |
|       | That is to find the vector for each sensor $s_j$ such that $s_j \in V_i$ $i \neq j$. |
|       | The number of vectors in $N_i$ is denoted by $|N_i|$. |

The list of filtered neighbor vectors for $s_i$. The filtering process is running as:

∀$V_a \in N_i$ do:

| $F_i$ | The list of filtered neighbor vectors for $s_i$. The filtering process is running as: |
|-------|--------------------------------------------------|
|       | ∀$s_c \in V_a$ do: |
|       | $V_x \leftarrow null$; |
|       | ∀$s_c \in V_a$ do: |
|       | $F_i.add(V_x)$ |
|       | That is to say, the sensor in the $N_i$ will be add to $F_i$ if and only if they are belong to $V_i$, otherwise they will be ignored. $|F_i|$ is the number of vectors inside $F_i$. |

$G^*(s_j)$ The associated groups of $s_i$, such that $G^*(s_j) = \{G^x | G^x \in G^* \text{ and } s_j \in G^x\}$.

$F_i[n]$ Get the filtered vector with index $n$.

4.1. Energy Saving during Data Routing. The grouping model provided an efficient energy management during communication. Consider group $G^0 = \{s_1, s_2, s_3, s_4\}$ of network depicted in Figure 2, and assume that $s_1$ has a data packet to send. Some of routing protocols just select a neighbor node to be the next hop and then transmit the data packet from $s_1$ to the selected neighbor node. In many cases, this selection could lead to wastage of energy consumption; for example, if the selected neighbor node is $s_2$ then the consumed energy to transmit the packet from $s_1$ to $s_2$ is wasted. Our grouping model avoided such situations as below.

1. In the same group, the none-leader to none-leader routing always leads to wastage of energy consumption. Note that the nodes in each group can communicate directly by one hop and the leaders of group acts as a gateway for the group.

2. In the same group, the leader to leader routing sometime leads to wastage of energy consumption.

3. In the same group, the leader to none-leader routing always leads to wastage of energy consumption.

The logical structure of grouping model for network deployed in Figure 2 is explained in Figure 4(b). When a maintaining the next hop (target node), the leader of the selected group will transmit the data directly to target node. After the operation of receiving data is finished, the target node finds its minimum distance group and next transmits the data recursively until the data reach the sink node. The flowchart of routing process is illustrated in Figure 4.
Input: \( s_i \) is the Source Node  
Output: send data to BS directly, or forward data via multi-hop path.  
(1) DataRouting\((s_j, \text{Message } msg)\)  
(2) \{  
(3) TargetGroup ← GetMinDistanceGroupFor\((s_j)\);  
(4) \( s_j ← \text{MinDistanceSensor}(\text{TargetGroup}); \) //\( s_i \) is the target node  
(5) if \( (s_i.ID \neq \text{SinkNode.ID}) \) \{  
(6) \}  
(7) \( s_i.\text{SendData}(s_j); \)  
(8) \( s_i.\text{ReceiveData}(s_j); \)  
(9) DataRouting\((s_j, \text{msg})\);  
(10) \}  
(11) else  
(12) \{  
(13) \( s_i.\text{SendData}(\text{SinkNode}); \)  
(14) \( \text{SinkNode.ReceiveData}(s_j); \)  
(15) \}  
(16) \}  

Algorithm 2: Data routing algorithm.

---

none-leader node has data to send, it just selects one of the leaders in the group and then transmits data to the selected leader. The leader can be selected upon three main parameters:

1. the residual energy of leader (i.e., the leader with more energy win),
2. the distance to source (i.e., the nearest leader win),
3. the current state of leader (i.e., busy or free).

5. Object Tracking and Energy Balancing

Consider a set of \( m \) mobile objects \( M = \{o_1, o_2, o_3, \ldots, o_m\} \) moving randomly in the range of set of \( n \) none-mobile nodes \( S = \{s_1, s_2, s_3, \ldots, s_n\} \). The nodes of network are partitioned into groups dynamically such that \( G^* = \{G^1, G^2, G^3, \ldots\} \). We assume that any node \( s_j \in S \) is aware of the associated groups to which it belongs, such that \( G^* (s_j) = \{G^x | G^x ∈ G^* \text{ and } s_j ∈ G^x\} \). Regarding energy balancing, when there is a notification message (i.e., Insert, Rescue, and Delete) to be sent from \( s_j \) to its associated groups \( G^* (s_j) \), and \( s_j \) will not be sent to all nodes in \( G^* (s_j) \) separately. However, \( s_j \) builds a Notification Tree \( NT(s_j) \) to manage the notifications (Algorithm 3), which will be sent to all nodes in \( G^* (s_j) \) with respect to energy saving of \( s_j \) and energy balancing of \( G^* (s_j) \). The Notification Tree ensures that all nodes in \( G^* (s_j) \) will consume approximately the same amount of energy during sending notifications.

5.1. Building the Notification Tree. The main objective of Notification Tree is to distribute the energy dissipation among the nodes of \( G^* (s_j) \) as evenly as possible such that each node in \( G^* (s_j) \) gets notified and consumes the same amount of energy. Each node \( s_j \) in the network has a Notification Tree \( NT(s_j) \) and its root is \( s_j \).

For node \( s_j \), consider graph \( G_j = (V_j, E_j) \), where \( V_j = \{s_x | s_x ∈ G^* (s_j)\} \) and \( E_j ⊆ [V_j]^2 \) as shown in Figure 5. The operation \( s_x ∈ G^* (s_j) \) is true if \( s_x ∈ G^* \) and \( G^x ∈ G^* (s_j) \). Consider \( s_j \) as articulation vertex or a cut vertex of \( G_j \) such that \( G_j - s_j = \{G_{j,0}, G_{j,1}, \ldots, G_{j,j}\} \), where \( G_{j,j} \) is a component or subgraph of \( G_j \).

When there is a notification message to be sent from \( s_j \) to its associated groups \( G^* (s_j) \), \( s_j \) will send \( |G_j - s_j| \) messages. That is to say, it will send a message to each component of \( G_j - s_j \). Each \( G_{j,i} \in G_j - s_j \) will build a spanning tree \( T_{j,i} \) which will be attached to the root node of \( NT(s_j) \) (Algorithm 4).

5.2. Building the Spanning of Subgraph. Each subgraph or component \( G_{j,ii} \in G_j - s_j \) will build a spanning tree \( T_{j,ii} \) taking into consideration that any node sensor in \( T_{j,ii} \) will consume approximately same energy for transferring same notification message as others. Assume that the vertex set of subgraph...
Algorithm 3: Build the Notification Tree.

\[ G_{ji} \text{ is } V_{ji}. \text{ The process of building the spanning tree can be managed in the following steps:} \]

(1) Select a vertex \( v_{\text{min}} \in V_{ji} \) with the minimum degree \( \delta(G_{ji}) \).
(2) Assume that \( v_{\text{min}} = \{s_0,s_1,\ldots,s_x\} \). Select a node \( s_b \in v_{\text{min}} (b \leq x, s_b \text{ is not a leader}) \) to be the root of \( T_{ji} \) and then link them; that is, if we select \( s_0 \) to be the root then the link operation will be as follows: \( s_j \) is the parent of \( s_1 \), \( s_1 \) will be the parent of \( s_2 \), \ldots, and \( s_{x-1} \) will be the parent of \( s_x \). Consider \( L_k = \{s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_x\} \).
(3) Select a leader node \( s_l \) from \( v_{\text{min}} \) and then link it to the tail of \( L_k \).
(4) Consider \( L_i = \{s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_x \rightarrow s_l\} \). If \( L_k \) is empty then \( l_i \) will be the root node.
(5) Mark \( v_{\text{min}} \) as visited vertex. Then select \( v_z \) as an incident of \( v_{\text{min}} \) such that \( l_j \in v_z \) and \( l_j \in v_{\text{min}} \).
(6) Repeat the steps from (2) to (4) for \( v_z \) until all nodes are marked.
Input: $G_{j,i}$
Output: $T_{j,i}$

1. $\text{minDegreeVertex} \leftarrow \text{SelectMinDegreeVertex}(G_{j,i});$
2. $T_{j,i} \leftarrow \text{null};$
3. $T_{j,i}.\text{Sensor} \leftarrow \text{subRootSensor}(\text{minDegreeVertex},G_{j,i});$
4. $\text{if } (T_{j,i}.\text{ID} = 1)/\text{the link list is Empty }$
5. $\text{LinkNodeLeaderSensors}(T_{j,i},\text{minDegreeVertex},G_{j,i});$
6. $\text{LeaderSensor} \leftarrow \text{selectLeaderSensor}(\text{minDegreeVertex},\text{LinkList});$
7. $\text{if } (\text{LeaderSensor} \neq \text{null})$
8. $\text{LinkLeaderSensor}(T_{j,i},\text{LeaderSensor});$
9. $\text{minDegreeVertex}.\text{wasVisited} \leftarrow \text{true};$
10. $\text{incidentVertex} \leftarrow \text{getIncidenVertex}(\text{LeaderSensor},G_{j,i});$
11. $\text{while } (\text{incidentVertex} \neq \text{null})$
12. {  
13. $\text{LinkNodeLeaderSensors}(T_{j,i},\text{incidentVertex},G_{j,i});$
14. $\text{LEADERsensor} \leftarrow \text{selectLeaderSensor}(\text{incidentVertex},T_{j,i});$
15. $\text{if } (\text{LEADERsensor} \neq \text{null})$
16. $\text{LinkLeaderSensor}(T_{j,i},\text{LEADERsensor});$
17. $\text{incidentVertex}.\text{wasVisited} \leftarrow \text{true};$
18. $\text{incidentVertex} \leftarrow \text{getIncidenVertex}(\text{LEADERsensor},G_{j,i});$
19. }

Algorithm 4: Build spanning tree $T_{j,i}$ for the subgraph $G_{j,i} \in G_j - s_j$.  

Figure 5: Notification tree.
6. Avoiding Data Redundancy Mechanism

An energy-efficient avoiding reporting of redundant data helps in reducing the consumed power during communications. In the following, we will mostly focus on tracking of object such that each object is allocated to one proxy node.

For $s_j \in S$ we assume that it has Sensed Object Table $SOT(s_j)$ to save the information of sensed object (i.e., current position or image), and it has a Notification Messages Table $NMT(s_j)$ to store the notification sent/received to/from other nodes; see Figure 6. When an object $o_i \in M$ is detected by $s_j$, $s_j$ will create a new entry in $SOT(s_j)$ for $o_i$ if and only if $s_j$ has no Notification record received from other $s_x \in G^*(s_j)$. When $s_j$ creates a new entry in $SOT(s_j)$ it will send an Insert notification message to $G^*(s_j)$ via Notification Tree $NT(s_j)$ to notify all nodes in $G^*(s_j)$ and ask them to add this notification message to their $NMT$ tables; hence the other nodes in $G^*(s_j)$ will not sense $o_i$ until $s_j$ send a Delete notification message indicating that $o_i$ got out of its range. In this time all nodes in $G^*(s_j)$ will delete the notification messages and $o_i$ will enter any other node in $G^*(s_j)$.

The reporting process which intended to report the sensed data to the base station is aiming for selecting the proxy node immediately. The selection of proxy node can be determined based on the $NMT$ records; for example, when there is more than only a record for $o_i$ in $NMT(s_j) = \{N_{i,j}, N_{i,j_0}, N_{i,j_1}, \ldots, N_{i,j_x}\}$ $x \neq j$, $N_{i,j_x}$ is a notification messages for object $o_x$ which is detected by node $s_j$. In this case $s_j$ will be the proxy node if it satisfied the election requirements, that is, maximum ID, residual energy, the current state, and so forth, if $s_j$ loses this round or was not elected as a proxy node it will remove $o_i$ record from $SOT(s_j)$ and remove $N_{i,j}$ notification message from $NMT(s_j)$. Worthily noted is that all $G^*(s_j)$ will have the same $NMT$. The selected proxy node will report the data and information to base station according to the reporting model explained in Section 4.

Algorithm 5 explained our strategy for controlling the tracking mechanism of objects such that at any time there will be one node selected as a proxy node for the mobile objects.

In line number 26, the function $s_p.StartTracking()$ instructs the proxy node $s_p$ to start reporting process of the sensed data. All sensed data of the objects $M = \{o_1, o_2, o_3, \ldots, o_x\}$ are saved in the $SOT(s_p) = \{r_0, r_1, r_2, \ldots, r_x\}$ where $r_i$ ($i \leq x$) is a record reference for the object $o_i$ and each $r_i = \{m_{0i}, m_{1i}, m_{2i}, \ldots, m_{xi}\}$ where $m_i$ ($i \leq x$) is the current sensed data (see Figure 5(a)). The proxy node $s_p$ reports the records in $SOT(s_p)$ sequentially. The records in $SOT(s_p)$ will be deleted after reporting that the process is finished.

7. Results and Discussion

In this work, we assumed that the radio channel is symmetric such that the energy required to transmit a message of $k$ bits from node $x$ to node $y$ is the same as energy required to transmit the same size message from node $y$ to node $x$ for a given signal to noise ratio (SNR). We assume a simple model called (First Order Radio Model). This model discussed in [13–16]. For this model, the energy dissipation to run the radio ($E_{elec}$) is 50 pJ/bit. $E_{elec}$ depends on the factors such as the digital coding, modulation, filtering, and spreading of the signal. The free space model of transmitter amplifier is $\epsilon_{fs} = 10 \text{ pJ/bit/m}^2$ and the multipath model of transmitter amplifier is $\epsilon_{mp} = 0.0013 \text{ pJ/bit/m}^2$. Both free space and multipath models are depending on the distance to the receiver and the acceptable bit-error rate. If the distance is less than a threshold $t_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}}$ (meters), the free space (fs) model is used; otherwise, the multipath (mp) model is used, as shown in (13). A sensor node will consume $E_{Tx}(d, k)$ of energy to transmit
Input: A mobile object $o_i$ get in the range of $s_j$

(1) int $x \leftarrow 0$; //message ID
(2) if (isWithinMyRange($o_i$))
(3) {
(4) if ($o_i \notin NMT(s_j)$)//no notification from other nodes
(5) {
(6) $m_x \leftarrow$ new Message(); //sensed data object
(7) $m_x$.MessageContent $\leftarrow$ “data”;
(8) $m_x$.isSent $\leftarrow$ false;
(9) if ($o_i \notin SOT(s_j)$)//the object has no SOT record
(10) {
(11) $r_i \leftarrow$ new SensedObjRecord(); //create new record in SOT
(12) $r_i$.MobileObject $\leftarrow o_i$;
(13) $r_i$.MessageTable.Add($m_x$);
(14) SOT($s_j$).Add($r_i$);
(15) INM $\leftarrow$ new NotificationMessage(); //create insert notification
(16) INM.MobileObject $\leftarrow o_i$;
(17) INM.NotificationType $\leftarrow$ Notification.Insert;
(18) INM.Source $\leftarrow s_j$;
(19) INM.Data $\leftarrow$ “data”;
(20) NMT($s_j$).Add(INM);
(21) BroadcastInsertNotification(INM, NT($s_j$)); //send the notification via notification tree
(22) $s_p \leftarrow$ SelectProxyNode($o_i$);
(23) if ($s_p = \text{null}$)
(24) {
(25) $s_p$.MyTrackingObject.Add($o_i$);
(26) $s_p$.StartTracking();
(27) }
(28) }//end if ($o_i \notin SOT(s_j)$)
(29) else //the object has a SOT record
(30) {
(31) $x \leftarrow x + 1$;
(32) UpdateSOT($o_i$).MessageTable.Add($m_x$); //update it.
(33) }
(34) }//if (!isInMyNMT($o_i$))
(35) }//end if $o_i$ within my range.
(36) else // $o_i$ get out my range.
(37) {
(38) DNM $\leftarrow$ GetNotificationMessageForObject($o_i$); //delete
(39) if ($\text{DNM} = \text{null}$)
(40) {
(41) DNM.NotificationType $\leftarrow$ Notification.Delete;
(42) DNM.Data $\leftarrow$ “data”;
(43) BroadcastDeleteNotification(DNM, NT($s_j$)); //send the notification via notification tree
(44) NMT($s_j$).Remove(DNM);
(45) }
(46) }

Algorithm 5: SOT($s_j$) and NMT($s_j$) controlling.

$k$ bits size message over distance $d$ and consume $E_{Rx}(d, k)$ of energy to receive transmit $k$ bits size message:

$$E_{Tx}(d, k) = \begin{cases} (k \cdot E_{elec}) + (k \cdot \epsilon_s \cdot d^2), & d < t_0, \\ (k \cdot E_{elec}) + (k \cdot \epsilon_{mp} \cdot d^4), & d \geq t_0, \end{cases}$$

$$E_{Rx}(k) = k \cdot E_{elec}.$$ 

For performance evaluation, we developed a software using C# (Visual Studio 2012). We will use three different coverage schemes as shown in Figure 7.

7.1. Grid Based Coverage. For grid based coverage schemes, we have selected two algorithms provided in [27], Grid Square Coverage version (1) and Grid Square Coverage version (2). Grid Square Coverage (1) algorithms have 78% coverage efficiency while Grid Square Coverage (2) has 73%.
Figure 7: The network topologies used to simulate the performance of GHS. The arrow indicated the position of base station (sink node). The green progress bar indicates the residual energy. Here all nodes are full charged 0.5 Joule.

Table 3: Simulation parameters.

| Parameter            | Value                      |
|----------------------|----------------------------|
| Network size         | $500 \times 500$ m$^2$     |
| Nodes                | 182                        |
| Radius               | 25 m                       |
| Data length          | 1024 bits                  |
| Initial energy       | 0.5 Joule                  |
| $E_{elec}$           | 50 nJ/bit                  |
| $\varepsilon_{amp}$  | 0.001 pJ/bit/m$^2$         |
| BS location          | Inside                     |
| Network topology     | Grid coverage 1            |

7.1.1. Grid Coverage Version (1). Table 3 shows the parameters we used to evaluate the performance of GHS in grid coverage version (1). The sensors deployed are as shown in Figure 7(a).

The evaluation results for using GHS on grid coverage version (1) are shown in Figures 8 and 9. In Figure 8, the residual energy (in joule) is explained after each node of 182 nodes sent five message of size 1024 bits (five rounds). It indicates that the closed nodes to base station will die out first due to network load. As the communication range of far node
is limited, the close nodes to base station will act as routers for forwarding packets that transmitted from far nodes. In addition, Figure 9 explained the dead nodes after 100, 200, and 300 rounds.

In this coverage scheme, the number of nodes in each group is 4, which means more channels for leaders to forward data. The average number of nodes in each path (the number of hops) is 4.5 when radius is 25 m. The min distance for each group is 35 m and the max distance is 49 m, that is to say, for any message (1024 bits) to be sent from node \(A\) to node \(B\) it either consumes 76288 nJ or 63744 nJ.

### 7.1.2. Grid Coverage Version (2)

Table 4 shows the parameters we used to evaluate the performance of GHS on grid coverage version (2). The network topology is shown in Figure 7(b).

In Figure 10, the residual energy (in joule) is explained after each node of 181 nodes sent five messages of size 1024 bits (five rounds). Figure 11 explained the die out node after 100, 200, and 300 rounds. In this coverage scheme, the number of nodes in each group is 2, which means there are two channels for leaders to forward data. The average number of nodes in each path (the number of hops) is 6.29 when radius is 25 m. The min distance for each group is 33 m and the max distance is 36 m, that is to say, for any message (1024 bits) to be sent from node \(A\) to node \(B\) it either consumes 62996.48 nJ or 64020.48 nJ.

### 7.2. Zigzag Based Coverage

In zigzag coverage scheme, the interest area divided into multiple zigzag patterns with multiple corners and lines segments, each node is deployed in a corner of zigzag pattern. Zigzag Pattern Scheme Deployment Algorithm expresses a very high coverage efficiency of 91%, and it expands and covers the whole interest area with minimum number of nodes, while it generates a very small coverage redundancy [28]. Table 5 shows the parameters we used to evaluate the performance of GHS on zigzag coverage scheme [16]. The network topology for zigzag is shown in Figure 5(c) (Table 5).

In Figure 12, the residual energy (in joule) is explained after each node of 138 nodes sent five messages of size 1024 bits (five rounds). It indicates that the closed nodes to base station will die out first due to network load. Due to limited communications range of far node, the close nodes to base station will act as routers for forwarding packets that transmitted from far...
Figure 11: The dead nodes (brown) after 100, 200, and 300 rounds using grid coverage 2 topology. The initial energy is 0.5 J/node with 50 nJ/bit dissipation to run the radio ($E_{\text{elec}}$). In each round, each sensor sends a packet of 1024 bits. The green progress bar indicates the residual energy.

Table 5: Simulation parameters.

| Parameter              | Value                |
|------------------------|----------------------|
| Network size           | 500 x 500 m$^2$      |
| Nodes                  | 138                  |
| Radius                 | 25 m                 |
| Data length            | 1024 bits            |
| Initial energy         | 0.5 Joule            |
| $E_{\text{elec}}$      | 50 nJ/bit            |
| $\varepsilon_{\text{amp}}$ | 0.001 pJ/bit/m$^2$  |
| BS location            | Inside               |
| Network topology       | Zigzag               |

Figure 12: The residual energy of nodes after five rounds. We assume that the battery of each node is 0.5 J. Network topology is zigzag.

Figure 13: The consumed energy of nodes after five rounds. We assume that the battery of each node is 0.5 J. Network topology is zigzag.

8. Conclusion and Further Works

This paper intended to develop a Sensors Grouping Hierarchy Structure (GHS) for partitioning the nodes into groups according to the maximum overlapped regions in the field. A group of sensors contain multiple number of nodes and a multiple number of leaders, and each node can belong to more than one group. We assumed that the interest field is covered with the minimum number of nodes such that the overlapped between the nodes is minimized and the nodes are expanded to cover the interested field completely. This structure enhances the collaborative, dynamic, distributive computing and communication of the system, and it maximizes the lifespan of nodes by minimizing energy consumption, balancing energy, and generating a little redundant data.
Figure 14: The dead nodes (brown) after 100, 200, and 300 rounds using zigzag topology. The initial energy is 0.5 J/node with 50 nJ/bit dissipation to run the radio ($E_{\text{elec}}$). In each round, each sensor sends a packet of 1024 bits. The green progress bar indicates the residual energy.

Figure 15: The dead nodes (brown) of three coverage schemes after 1000 rounds of GHS, and the battery of each sensor is 0.5 J. In each round each sensor sends a packet of 1024 bits. The energy dissipation to run the radio ($E_{\text{elec}}$) is 50 nJ/bit. The green progress bar indicates the residual energy.
Table 6: Comparison between GHS and other well-known structures over different topologies and different initial energy.

| Energy (/node) | Coverage protocol | Nodes | Structure | Round first node dies |
|----------------|-------------------|-------|-----------|-----------------------|
| 0.5            | Zigzag            | 138   | DC        | 86                    |
|                |                   |       | GHS       | 93                    |
|                |                   |       | SC        | 77                    |
| Grid           | 182               |       | SC        | 67                    |
|                |                   |       | DC        | 80                    |
|                |                   |       | PTP       | 51                    |
| 1.0            | Zigzag            | 138   | DC        | 170                   |
|                |                   |       | PTP       | 173                   |
|                |                   |       | GHS       | 190                   |
|                |                   |       | SC        | 234                   |
| Random         | 100               |       | SC        | 130                   |
|                |                   |       | DC        | 163                   |
|                |                   |       | PTP       | 102                   |
|                |                   |       | GHS       | 186                   |
|                |                   |       | SC        | 140                   |
|                |                   |       | PTP       | 243                   |
|                |                   |       | GHS       | 274                   |

Notes:
(i) PTP = peer-to-peer, DC = dynamic clustering, and SC = static clustering.
(ii) One round means that all nodes in network take a turn to send a message, that is, one round in grid means 182 messages.

For further research, we plan to extend our structure by studying the influence of nodes quick movement in the field. And we will study the object tracking intensively using this structure. The researchers are very welcomed to develop energy efficient data routing algorithms and objects tracking algorithms using GHS.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the anonymous reviewers for the helpful comments and suggestions. The National Natural Science Foundation of China (nos. 61472382, 61272472, and 61232018) and the National Science Technology Major Project (no. 2012ZX10004301-609) support this paper.

References

[1] X. Bai, Z. Yun, D. Xuan, T. H. Lai, and W. Jia, "Optimal patterns for four-connectivity and full coverage in wireless sensor networks," IEEE Transactions on Mobile Computing, vol. 9, no. 3, pp. 435–448, 2010.
[2] M. Cardei, J. Wu, M. Lu, and M. O. Pervaz, "Maximum network lifetime in wireless sensor networks with adjustable sensing ranges," in Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '05), pp. 438–445, August 2005.
[3] Y. Ma, M. Richards, M. Ghanem, Y. Guo, and J. Hassard, "Air pollution monitoring and mining based on sensor Grid in London," Sensors, vol. 8, no. 6, pp. 3601–3623, 2008.
[4] N. A. A. Aziz, K. A. Aziz, and W. Z. W. Ismail, "Coverage strategies for wireless sensor networks," World Academy of Science, Engineering and Technology, vol. 38, pp. 145–150, 2009.
[5] S. C.-H. Huang, S. Y. Chang, H.-C. Wu, and P.-J. Wan, "Analysis and design of a novel randomized broadcast algorithm for scalable wireless networks in the interference channels," IEEE Transactions on Wireless Communications, vol. 9, no. 7, pp. 2206–2215, 2010.
[6] J. Wu and S. Yang, "Coverage and connectivity in sensor networks with adjustable ranges," in Proceedings of the International Workshop on Mobile and Wireless Networking (MWN ’04), August 2004.
[7] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in Proceedings of the IEEE (INFOCOM ’05), pp. 1976–1984, March 2005.
[8] S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh, "Optimal energy aware clustering in sensor networks," Sensors, vol. 2, no. 7, pp. 258–269, 2002.
[9] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," Computer Communications, vol. 30, no. 14–15, pp. 2826–2841, 2007.
[10] V. Kawadia and P. R. Kumar, "Power control and clustering in Ad Hoc networks," in Proceedings of the IEEE INFOCOM, vol. 1, pp. 459–469, San Francisco, Calif, USA, March 2003.
[11] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, "Max-min d-cluster formation in wireless Ad Hoc networks," in Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ‘00), pp. 32–41, March 2000.
[12] A. B. McDonald and T. F. Znati, "A mobility based framework for adaptive clustering in wireless ad-hoc networks," IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1466–1487, 1999.
[13] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless micro-sensor networks," in Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS ’00), pp. 1–10, Maui, Hawaii, USA, January 2000.
[14] 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.
[15] R. Akl and U. Sawant, "Grid-based coordinated routing in wireless sensor networks," in Proceedings of the 4th Annual IEEE Consumer Communications and Networking Conference (CCNC ’07), pp. 860–864, January 2007.
[16] S. Lindsey and C. S. Raghavendra, "PEGASIS: power-efficient gathering in sensor information systems," in Proceedings of the
[17] R. Sheikhpour, S. Jabbehdari, and A. Khadem-Zadeh, "Comparison of energy efficient clustering protocols in heterogeneous wireless sensor networks," *International Journal of Advanced Science and Technology*, vol. 36, pp. 27–40, 2011.

[18] J.-S. Chen, Z.-W. Hong, N.-C. Wang, and S.-H. Jhuang, "Efficient cluster head selection methods for wireless sensor networks," *Journal of Networks*, vol. 5, no. 8, pp. 964–970, 2010.

[19] J. Kulik, W. R. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom ’99)*, pp. 169–185, Seattle, Wash, USA, August 1999.

[20] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM ’00)*, pp. 56–67, Boston, Mass, USA, November 2000.

[21] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, 1999.

[22] L. Li and J. Y. Halpern, "Minimum-energy mobile wireless networks revisited," in *Proceedings of the International Conference on Communications (ICC ’01)*, pp. 278–283, Helsinki, Finland, June 2000.

[23] J. Wang, J.-U. Kim, L. Shu, Y. Niu, and S. Lee, "A distance-based energy aware routing algorithm for wireless sensor networks," *Sensors*, vol. 10, no. 10, pp. 9493–9511, 2010.

[24] R. Sheikhpour and S. Jabbehdari, "A two-level cluster based routing protocol for wireless sensor networks," *International Journal of Advanced Science and Technology*, vol. 45, pp. 19–30, 2012.

[25] F. Derogarian, J. C. Ferreira, and V. G. Tavares, "A routing protocol for WSN based on the implementation of source routing for minimum cost forwarding method," in *Proceedings of the 5th International Conference on Sensor Technologies and Applications (SENSORCOMM ’11)*, Nice, France, August 2011.

[26] F. Tang, I. You, S. Guo, M. Guo, and Y. Ma, "A chain-cluster based routing algorithm for wireless sensor networks," *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1305–1313, 2012.

[27] A. Hawbani, X. Wang, N. Hussaini, and S. Karmoshi, "Grid Coverage Algorithm & Analyzing for wireless sensor networks," *Network Protocols and Algorithms*, vol. 6, no. 4, 2014.

[28] A. Hawbani and X. Wang, "Zigzag coverage scheme algorithm & analysis for wireless sensor networks," *Network Protocols and Algorithms*, vol. 5, no. 4, pp. 19–38, 2013.
Submit your manuscripts at
http://www.hindawi.com