Article

Sensors Grouping Hierarchy Structure (GHS) for Wireless Sensor Network

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

The National Natural Science Foundation of China (NO.61272472, 61232018, 61202404) and the National Science Technology Major Project (NO. 2012ZX10004301-609) support this paper.

[1]University of Science and Technology of China http://en.ustc.edu.cn/
[2] Hefei University of Technology http://www.hfut.edu.cn/ch/

***** Author to whom correspondence should be addressed; E-Mail: *wangxfu@ustc.edu.cn*;

Received: / Accepted: / Published:

Abstract: *There are many challenges in implementation of wireless sensor network systems: clustering and grouping are being 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 as well as it generate a little redundant data during object tracking.*

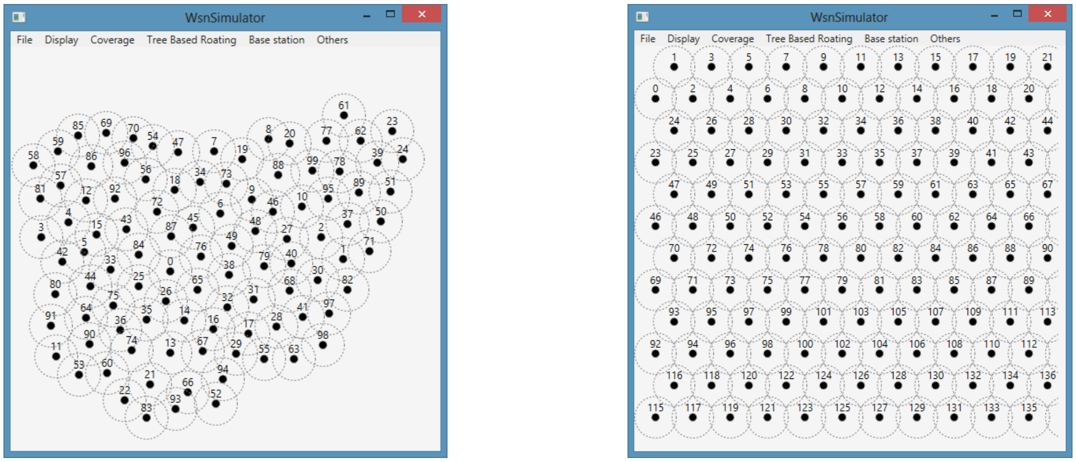
1. Introduction

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 [6, 7, 8]. 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 [9]. 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 [10]. Many of clustering algorithms [11, 12, 13] 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 [10]. During the recent years, there are also many clustering algorithms design especially for WSN, most of them mainly focused on energy dissipation, energy-balancing .etc.

In the literature, many protocols to reduce energy consumption have been proposed. The Low Energy Adaptive Clustering Hierarchy (LEACH) protocol [14-15] 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 [16]. 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 Systems PEGASIS [17] is comes with a 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 to clustering in WSN, the [18] 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 [19], presented efficient request-oriented coordinator methods for hierarchical sensor networks.

On other hand, the traditional structures in WSNs can be categorized into data centric and location based. The [20-21] provided two routing protocols based on data centric. In [20], Sensor Protocols for Information via Negotiation (SPIN) uses the data negotiation scheme among sensor nodes to decrease data redundancy and save energy. Direct Diffusion [21] is another data-centric routing protocol. The data generated by sensor nodes is named by 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 [22-23]. Based on theoretical analysis and numerical illustration under different energy and traffic models the [24] (DEAR) Distance-based Energy Aware Routing 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) [25] 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 [26] 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 [27], this algorithm makes a full use of the advantages of LEACH [14] and PEGASIS [17], 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 ownchain head node in parallel, using an improved chain routing protocol. In the second stage, all chain head nodes group as a cluster in a selforganized 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 between the nodes is minimized and the nodes are expanded to cover the interested field completely. As well as, we assumed that the nodes has the same communication range indicated by the dotted circle. With such assumptions, unfortunately, if the CH nodes has the same communication range, the static clustering or dynamic clustering will not perform as efficient as Peer-to-peer network.



(a) Random 100 nodes.

(b) 137 nodes deployed using Zigzag sachem.

Figure 1: Two different networks. The filled small circle indicates the location of node, and the dotted circle indicates the communication range of node.

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

- Data routing and energy saving (section 4).
- Object tracking and energy balancing (section 5).
- Avoiding data redundancy (section 6).

The difference between clustering and grouping is simply as below:

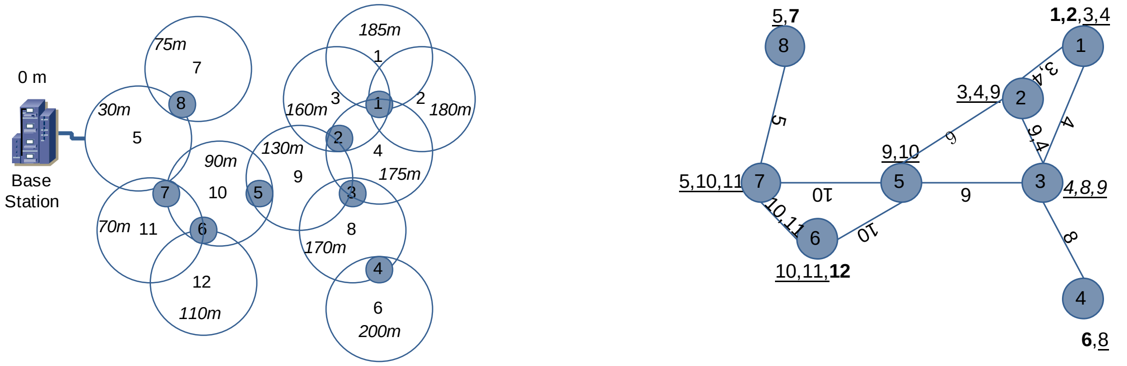
- Clustering is performed by assigning each node to a specific cluster that is to say each node belong to only one cluster [9], There is only one CH in each cluster.

- Grouping model is to divide the network into groups such that each node can belong to more than group in the same time. There are more than 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* such that each group contains a number of nodes and a number of leaders i.e., the network in Figure 2(a) contains 8 *maximum covered regions* (indicated by small filled circles and given a numbers from ① to ⑧). That is to say, this network can be partitioned into eight groups. In Figure 2(b) the graph vertices are shown (①to⑧) each vertex represented a group of sensors i.e., the group (①) contains four sensors namely (1, 2, 3, 4). Two groups are said to be adjacent if they contain one or more sensor in common i.e., the groups (①) and (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.



(a)Sensors deployed in an interest area. The small filled circle are the maximum overlapped regions (the maximum covered regions). The big white unfilled circle indicates the communication rang of the node.

(b) Grouping Graph, underlined sensors are the leaders of groups.

Figure2: 12 nodes deployed randomly.

In what follows, we will address the grouping problem using a simple mathematical model. Here we can express 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 $(n\ x\ n)$.

$$A^* = \begin{bmatrix} 1 & 1 & & 0 \\ 1 & 1 & & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 1 & & 1 \\ 0 & 0 & & 1 \end{bmatrix}_{n \times n} , \qquad a_{ij} = \begin{cases} 1 & s_i\ and\ s_j\ are \text{ overlapped} \\ 1 & i = j \\ 0 & s_i\ and\ s_j\ are\ not \text{ overlapped} \end{cases}$$

For example, the network in Figure 2 is represented by a square matrix （12×12）

$$A^* = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

We can find the *maximum covered regions* by partitioning A^* into square sub-matrices $\{A_1, A_2, A_3 \dots\}$ such that all elements of sub-matrix $a_{ij} = 1$,as well as the sub-square matrix contains the *maximum number of rows and columns* . Each square sub-matrix represent a group of sensors. For example, the network in Figure 2 is partitioned into eight square sub-matrices listed as below:

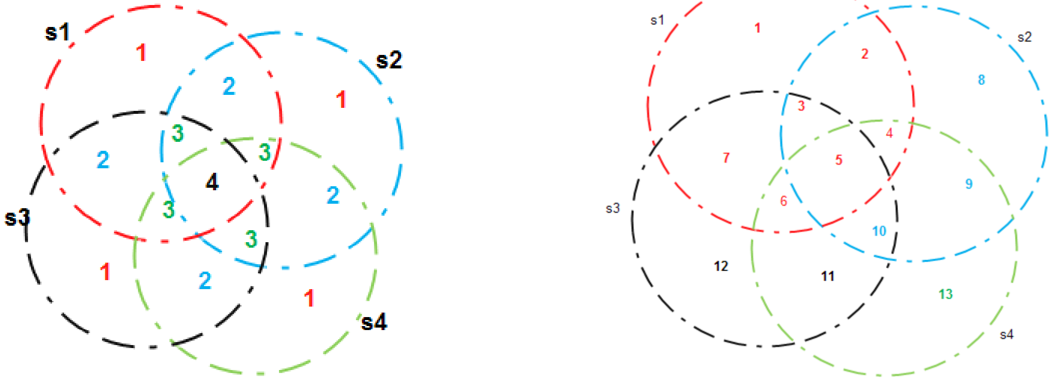
$$A_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}_{g1=(1,2,3,4)} , A_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{g2=(3,4,9)} , A_3 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{g3=(4,8,9)}$$

$$A_4 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}_{g4=(6,8)} , A_5 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}_{g5=(9,10)} , A_6 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{g6=(10,11,12)} , A_7 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{g7=(5,10,11)} , A_8 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}_{g8=(5,7)}$$

If WSN is connected then $A^*(n\ x\ n)$ can be partitioned into square sub-matrices $\{A_1, A_2, A_3 \dots A_\text{i}\}$ where i is an integer and $1 \leq i \leq n - 1$.

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

A group $G^k \in G^*$ contains $\mathbf{1 + k(k-1)}$ *sub-areas* among them there will be one area is $k - covered$, k areas are $1 - covered$, k areas are $2 - covered$ ...and k areas are $k - 1\ covered$. Note that when an object moves in $x - covered$ area ($x > 1 , x \in \mathbb{Z}$) 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 $1 + k(k-1)/2$ sub-areas, 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 $k - 1\ covered$ i.e., see Figure 3.



(a) The coverage degree of G^4 (b) The 13 sub-areas of group G^4

Figure 3: A group G^4

A group $G^k \in G^*$ contains $\mathrm{R}(G^k) = 1 + k(k-1)$ 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:

$$R(G^k) = \sum_{j=0}^{k} R(s_j) \dots\dots\dots..(1)$$

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

$$R(s_j \in G^k) = \begin{cases} f(k) = k + f(k-1) - 1 \\ k \geq 1 \\ f(1) = 1 \end{cases}$$

We can solve this equation using generation functions and obtain:

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

$$R(s_j \in G^k) = 1 + \binom{k}{2}$$

By removing the repetition from (1), we can get:

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

$$R(G^k) = 1 + \binom{k}{2} + \sum_{i=1}^{k-1} (k - i) =$$

$$1+\frac{k!}{2!\,(k-2)!}+\frac{k(k-1)}{2}=$$
$$1+\frac{k(k-1)}{2}+\frac{k(k-1)}{2}$$
$$=1+k(k-1)$$

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

$$R(G^k)=\Phi\left(C_{0,k}\right)+\Phi\left(C_{1,k}\right)+\ \Phi\left(C_{2,k}\right)+\cdots\Phi\left(C_{k,k}\right)$$
$$=\sum_{i=0}^{k}\Phi\left(C_{i,k}\right)=0+k+k+\cdots+1$$
$$=1+k(k-1)$$

Furthermore, among the $R\left(s_j\in\ G^k\right)$ there will be *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 $k-1\ covered$, i.e., 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 $\theta\left(c_{x,j}\right)$ be the number of $c_{x,j}$ regions, then for any $s_j\in\ G^k$, the $\boldsymbol{\theta\left(c_{i,j}\right)=i}$ where $(k-1\geq i\geq 0)$. We can obtain that $\theta\left(c_{0,j}\right)=0$ and $\theta\left(c_{k,j}\right)=1$. It is easy to see that:

$$R\left(s_j\in\ G^k\right)=\theta\left(c_{0,j}\right)+\theta\left(c_{1,j}\right)+\theta\left(c_{2,j}\right)+\cdots\theta\left(c_{k,j}\right)$$
$$=\theta\left(c_{k,j}\right)+\sum_{i=0}^{k-1}\theta\left(c_{i,j}\right)=\theta\left(c_{k,j}\right)+0+1+2+3+\cdots+(k-1)$$
$$=1+\frac{k(k-1)}{2}$$

Therefore, we can get number of regions $R\left(s_j\in\ G^*\left(s_j\right)\right)$, where $G^*\left(s_j\right)$ is the set of *associated groups* of s_j such that $G^*\left(s_j\right)=\left\{G^x|G^x\in\ G^*\text{ and } s_j\in\ G^x\right\}$. Let $\Upsilon=\left|\ G^*\left(s_j\right)\right|$ then:

$$R\left(s_j\in\ G^*\left(s_j\right)\right)\leq\left(\sum_{G^x\in\ G^*\left(s_j\right)}R\left(s_j\in\ G^x\right)\right)-(\Upsilon-1)\ \dots\dots\dots..(2)$$

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

This grouping model has advantages since it makes the WSNs stress-free in many aspects such as Data routing, Data redundant avoiding …etc. Below, we will provide a *distributed* Grouping

Algorithm, which executed inside the processing unit of the sensor node. However, it utilize the information of nearby sensors to find the *maximum covered regions*.

Table **1**: The **S**ymbols used for **g**rouping algorithm

Symbol	**Meaning**
$\boldsymbol{S}$	A set of sensors.
$\boldsymbol{D}$	Euclidean distance.

$\boldsymbol{V_i}$	The list of neighbor nodes of s_i. Here we call V as the vector of sensor. $\lvert V_i \rvert$ Is the number of sensor inside V_i.
	$D = \sqrt{(\mathrm{x}_{\mathrm{s_i}} - \mathrm{x}_{\mathrm{s_j}})^2 + (\mathrm{y}_{\mathrm{s_i}} - \mathrm{y}_{\mathrm{s_j}})^2}$
	$\forall\, s_j \in S\ do$:
	{
	$if\left(r_i + r_j < D\right)\ do$:
	$V_i\_add\left(s_j\right)$;
	}

$\boldsymbol{N_i}$	The list of neighbor vectors for s_i. $\forall\, s_j \in V_i\ do$: $find\ V_j\ and\ then\ add\ V_j\ to\ N_i\ by\ calling$: $N_i\_add\left(V_j\right)$. 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 $\lvert N_i \rvert$.

$\boldsymbol{F_i}$	The list of filtered neighbor vectors for s_i. The filtering process is running as:
	$\forall\, V_a \in N_i\ do$:
	{
	$V_x \leftarrow null$;
	$\forall\, s_c \in V_a\ do$:
	{
	$if\ (s_c \in V_i)\ do$:
	{
	$V_x.\ add(s_c)$;
	}
	}
	$F_i\_add(V_x)$
	}
	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. $\lvert F_i \rvert$ is the number of vectors inside F_i.

$G^*(s_j)$	The associated groups of s_i , such that $G^*(s_j) = \{G^x \mid G^x \in G^* \text{ and } s_j \in G^x\}$.
$\boldsymbol{F_i[n]}$	Get the filtered vector with index n.

Algorithm 1 shows how the sensor node s_i aware its associated groups $G^*(s_j) = \{G^x \mid G^x \in G^* \text{ and } s_j \in G^x\}$.

Algorithm 1(*Grouping Algorithm*)

Find $G^*(s_i)$

Input: s_i, S

Output: $G^*(s_i)$ *the* associated groups of s_i

```
$V_i \leftarrow s_i.Vector$
$\boldsymbol{for}(\boldsymbol{int}\ k \leftarrow 0; k < |F_i|; k++)$
{
  $V_k \leftarrow F_i[k];$
  $s_k \leftarrow V_i[k];$
  $\boldsymbol{if}(|V_k| = 2)$
    {
      $G^k \leftarrow V_k$
      $G^*(s_i) += G^k$
    }
  $\boldsymbol{else}$
    {
        $\boldsymbol{for}(int\ m \leftarrow k+1; m < |F_i|; m++)$
          {
            $V_m \leftarrow F_i[m];$
            $\boldsymbol{if}(s_k \in V_m)$
              {
                $G^{|V_k \cap V_m|} \leftarrow V_k \cap V_m$
                $G^*(s_i) += G^{|V_k \cap V_m|}$
              }
          }
    }
}
```

By Analyzing **Algorithm 1**, we can predict the resources (i.e., memory, communication bandwidth, 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 complexly 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 belongs to more than one group. Two groups are said to be adjacent if they have one sensor or more in common.

We consider a **statics Graph** $\boldsymbol{G = (V, E, \mu)}$ to represent the topology of sensors deployed in a sensing field, where V indicates the vertices, E indicates the edges and μ indicates the weight function where $\mu: E \to \mathbb{R}^+$ 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 their geographical location and $\mathrm{\mu(v, v) = 0}$ for any node $\mathrm{v \in V}$. We assume that G is connected, i.e., there is a path of nodes connects 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 $G^x \in G^*$ contains a number of *sensors i.e.,* $G^x = \{s_1, s_2 \ldots s_v\}, |G^x| = v$ and multiple number of *leaders*. The set of groups $G^*(s_j)$ is called the associated groups of s_j , such that $G^*(s_j) = \{G^x | G^x \in G^* \text{ and } s_j \in G^x\}$. The leaders of groups are not randomly chosen. However, the node can be a leader if and only if it belongs to more than one group, i.e., in Figure2, the sensor 7 can't be a leader since it belongs to only one group.

Table: 2 the groups and Leaders of network (Figure 2)

Node	Groups	Members	Leaders
1	1	(1,2,3,4)	4,3
2	1	(1,2,3,4)	3,4
3	1,2	(1,2,3,4),(3,4,9)	3,4,9
4	1,2,3	(1,2,3,4),(3,4,9),(4,8,9)	3,4,8,9
5	8,7	(5,7),(5,10,11)	10,11,5
6	4	(6,8)	8
7	8	(5,7)	5

8	3,4	(4,8,9),(6,8)	4,8,9
9	2,3,5	(3,4,9),(4,8,9),(9,10)	3,4,8,9,10
10	5,6,7	(9,10),(10,11,12),(5,10,11)	5,9,10,11
11	6,7	(10,11,12),(5,10,11)	5,10,11
12	6	(10,11,12)	10,11

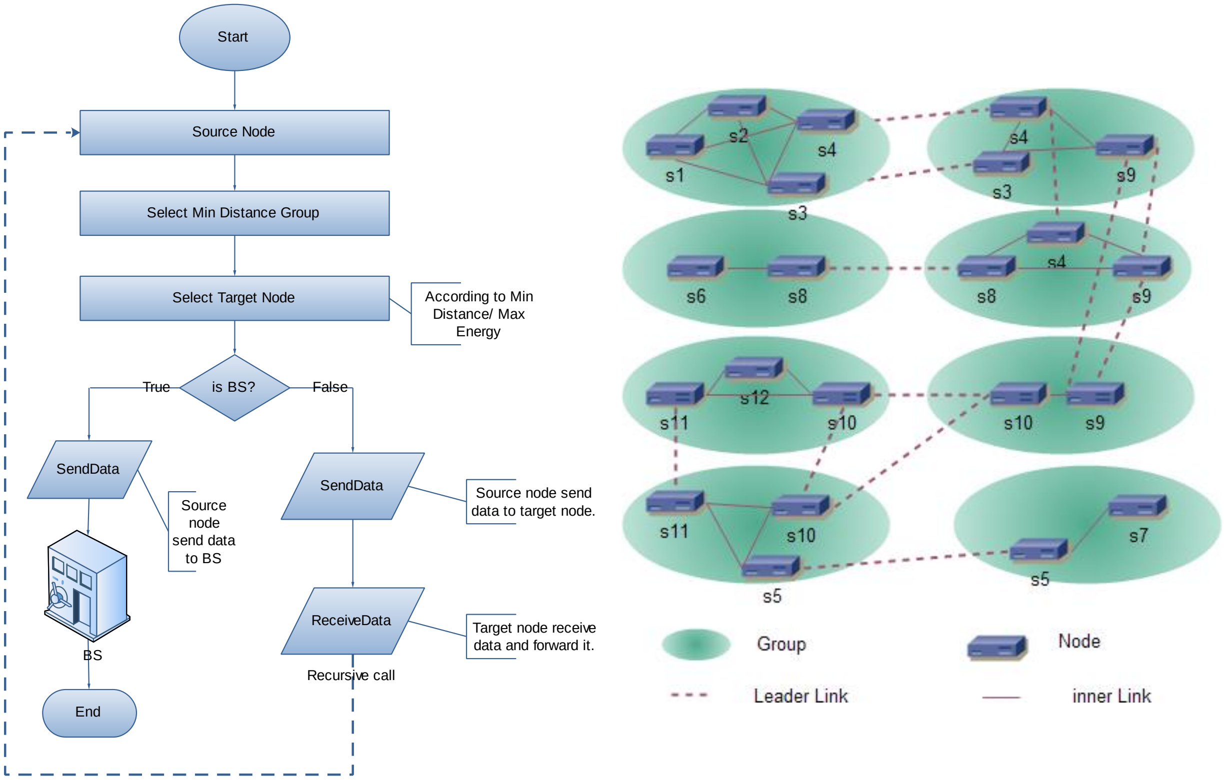
4. Data Routing

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 choose 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 maintain the next hop (Target node), the leader of the selected group will be transmit the data directly to Target node. After the operation of receiving data is finished, the target node find its minimum distance group and next and transmit the data recursively until the data reach the sink node. The flow chart of routing process is illustrated in Figure 4.

Algorithm 2: (Data Routing Algorithm)
Input: s_i is the Source Node
Output: send data to BS directly, or forward data via multi-hop path.

```
DataRouting( $s_i$, $\boldsymbol{Message\ msg}$)
{
   TargetGroup ← $GetMinDistanceGroupFor(s_i)$;
   $s_t$ ← MinDistanceSensor(TargetGroup); // $s_t$ is the target node
   $if(s_t$.ID≠SinkNode.ID)
    {
      $s_i.SendData(s_t)$;
      $s_t$.ReceiveData($s_i$);
      DataRouting( $s_t$, $\boldsymbol{msg}$);
    }
   $else$
    {
      $s_i.SendData$(SinkNode);
      SinkNode.ReceiveData($s_i$);
```

```
    }
}
```



(a)Flow chart of Data Routing (b) Logic structure of the groups in Figure2.

Figure 4: Grouped based Data Routing

Energy Saving during Data Routing

The grouping model provided an efficient energy management during communication. Consider the group $G^4 = \{s_1, s_2, s_3, s_4\}$ 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 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 Figure4 (b). When a *none-leader* node has a data to send, it just select one of the *leaders* in the group and then transmit 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 node (i.e., the nearest *leader* win) and
3- The current state of *leader* (i.e., busy or free).

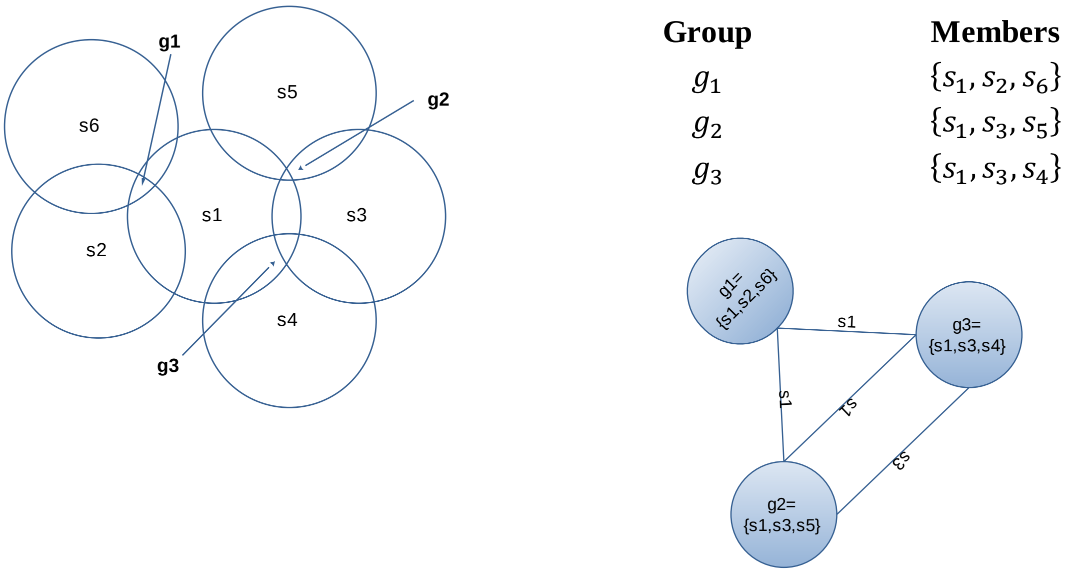
5. Object Tracking and Energy Balancing

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

Building the Notification Tree

The main objective of *Notification Tree* is to distribute the energy dissipation among the nodes of $\mathrm{G}^*(\mathrm{s_j})$ as evenly as possible such that each node in $\mathrm{G}^*(\mathrm{s_j})$ get notified and consume the same amount of energy. Each node $\mathrm{s_j}$ in the network has a *Notification Tree* $NT(s_j)$ *and its root is* $\mathrm{s_j}$. For the node $\mathrm{s_j}$, consider the Graph $G_j = (V_j, E_j)$, where $V_j = \{s_x | s_x \in \mathrm{G}^*(\mathrm{s_j})\}$ and $E_j \subseteq [V_j]^2$ as shown in Figure 5. The operation $s_x \in \mathrm{G}^*(\mathrm{s_j})$ is true if $s_x \in G^i$ and $G^i \in G^*(s_j)$. Consider $\mathrm{s_j}$ as *articulation vertex or a cut vertex* of G_j such that $G_j - s_j = \{G_{j,0}, G_{j,1} \dots G_{j,i}\}$ where $G_{j,i}$ is a component or subgraph of G_j.

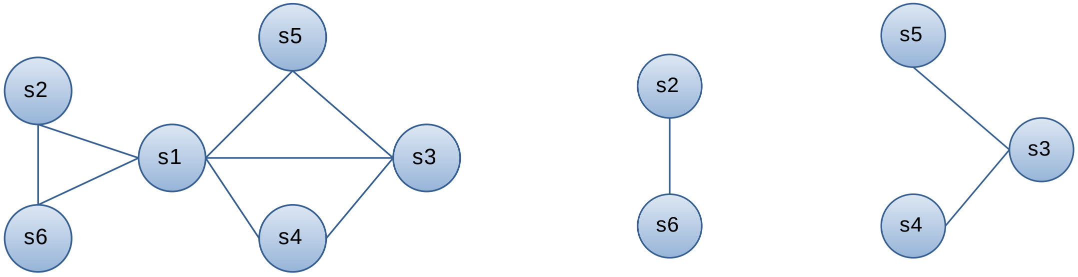
When there is a notification message to be send from s_j to its associated groups $G^*(s_j)$, s_j will send $|G_j - s_j|$ messages. That is to say, it will send a message to each component of $G_j - s_j$. Each $G_{j,i} \in G_j - s_j$ will build a spanning tree $T_{j,i}$ which will be attached to the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



Group	Members
g_1	$\{s_1, s_2, s_6\}$
g_2	$\{s_1, s_3, s_5\}$
g_3	$\{s_1, s_3, s_4\}$

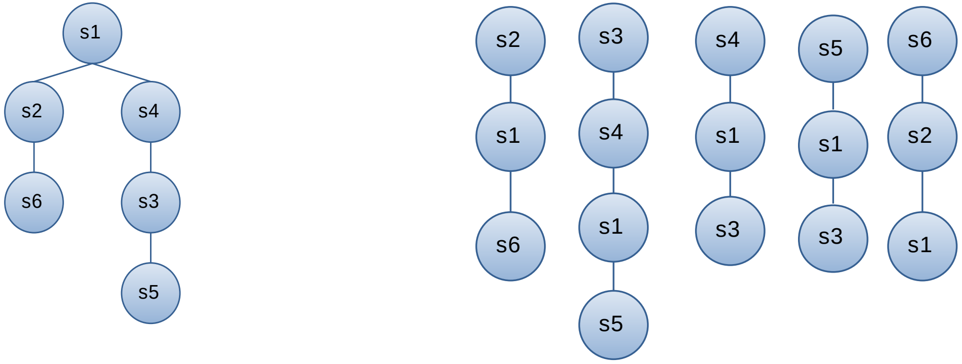
(a): Random 5 Nodes

(b): Groups $G^*(s_1)$



(c): Undirect Graph G_1

(d): Components of $G_1 - s_1$



(e): $\mathrm{NT}(\mathrm{s}_1)$

(f): $\mathrm{NT}(\mathrm{s}_2), \mathrm{NT}(\mathrm{s}_3), \mathrm{NT}(\mathrm{s}_4), \mathrm{NT}(\mathrm{s}_5)$ and $\mathrm{NT}(\mathrm{s}_4)$

Figure 5: Notification Tree.

Algorithm 3:Build the Notification Tree

Input : $\boldsymbol{s_j}$

Output: $\boldsymbol{NT(s_j)}$

```
$Root \leftarrow s_j;$
$G_j \leftarrow (V_j, E_j)$ // $V_j = \{s_x | s_x \in G^*(s_j)\}$ and $E_j \subseteq [V_j]^2$
$G_{j,i} \leftarrow G_j - s_j$ //$\{G_{j,0}, G_{j,1} \ldots G_{j,i}\}$ subgraphs
```

```
    $for(int\ i \leftarrow 0; i < |\ G_{j,i}|; i++)$
    {
      $T_{j,i} \leftarrow BuidSpanningTree(\ G_{j,i})$ // build the spanning tree for each subgraph.
      $Root.childern.add(\ T_{j,i})$
    }
```

Building the Spanning of subgraph

Each subgraph or component $G_{j,i} \in G_j - s_j$ will build a spanning tree $T_{j,i}$ with taking consideration that any node sensor in $T_{j,i}$ will consume approximately same energy for transfer same the notification message as others. Assume that the vertex set of subgraph $G_{j,i}$ is $V_{j,i}$. The process of building the spanning tree can be managed in the following steps:

1- Select a vertex $v_{min} \in V_{j,i}$ with the minimum degree $\delta(\ G_{j,i})$.

2- Assume that $v_{min} = \{s_0, s_1 \dots s_x\}$. Select a node $s_b \in v_{min}(b \leq x, s_b$ is not a leader) to be the root of $T_{j,i}$ and then link them i.e., if we select s_0 to be the root then the Link operation will be as: s_0 is the parent of s_1, s_1 will be the parent of $s_2 \dots$ and s_{x-1} will be the parent of s_x. $L_k = \{s_0 \rightarrow s_1 \rightarrow \cdots s_x\}$.

3- Select a leader node s_l from v_{min} and then link it to the tail of L_k:
$L_i = \{s_0 \rightarrow s_1 \rightarrow \cdots s_x \rightarrow s_l\}$. If L_k is empty then l_i will be the root node.

4- Mark v_{min} as visited vertex. Then select v_z as an incident of v_{min} such that $l_i \in v_z$ and $l_i \in v_{min}$.

5- Repeat the steps from 2-4 for v_z until all nodes are marked.

Algorithm 4: Build Spanning Tree $T_{j,i}$ for the *subgraph* $G_{j,i} \in G_j - s_j$
Input : $G_{j,i}$
Output: $T_{j,i}$

```
$minDegreeVertex \leftarrow SelectMinDegreeVertext(\ G_{j,i})$;
 $T_{j,i} \leftarrow null$;
 $T_{j,i}.Sensor \leftarrow subRootSensor(minDegreeVertex,\ G_{j,i})$;
$if\ (\ T_{j,i}.ID \neq -1)$// **the link list is Empty**
   $LinkNoneLeaderSensors(\ T_{j,i}, minDegreeVertex,\ G_{j,i})$;
 $LeaderSensor \leftarrow selectLeaderSensor(minDegreeVertex, LinkList)$;
$if\ (LeaderSensor \neq null)$
$LinkLeaderSensor(\ T_{j,i}, LeaderSensor)$;
$minDegreeVertex.wasVisited \leftarrow true$;
$incidentVertex \leftarrow getIncidenVertext(LeaderSensor,\ G_{j,i})$;
$while\ (incidentVertex \neq null)$
{
 $LinkNoneLeaderSensors(\ T_{j,i}, incidentVertex,\ G_{j,i})$;
 $LEADERsensor \leftarrow selectLeaderSensor(incidentVertex,\ T_{j,i})$;
  $if\ (LEADERsensor \neq null)$
```

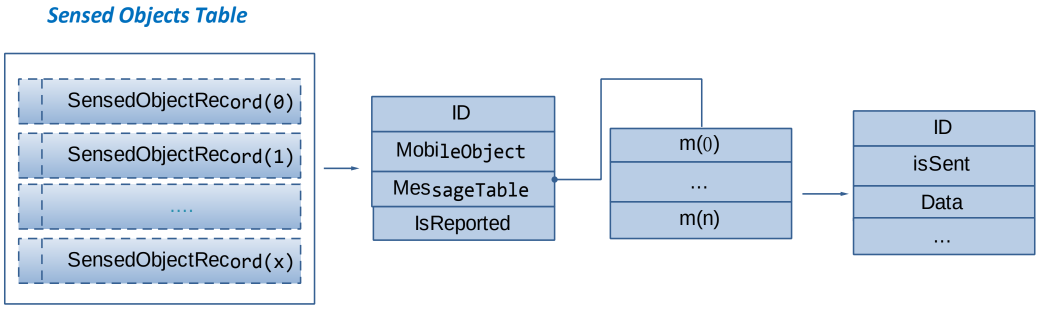
```
        $LinkLeaderSensor(\ T_{j,i}, LEADERsensor);$
        $incidentVertex.wasVisited \leftarrow true;$
       $incidentVertex \leftarrow getIncidenVertext(LEADERsensor,\ G_{j,i});$
    }
```

6. Avoiding Data Redundancy Mechanism

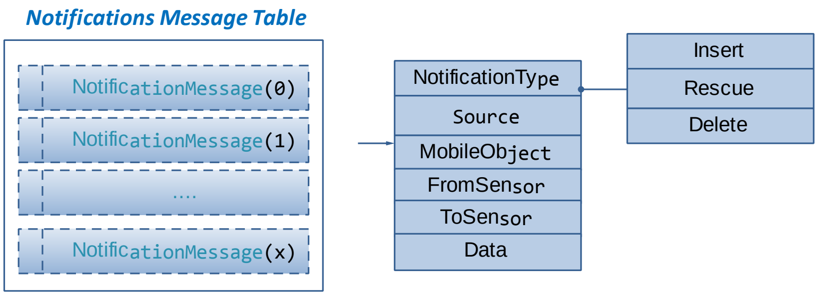
An energy-efficient avoiding reporting of redundant data helps in reducing the consumed power during communications. In what follows, we will mostly focus on tracking of object such that each object is allocate to one proxy node.

For $s_j \in S$ we assume that it has $Sensed\ Object\ Table\ SOT(s_j)$ table to save the information of sensed object (i.e., current position or image .etc.), and it has a $Notification\ Messages\ Table\ NMT(s_j)$ to store the notification send/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 sensing o_i until s_j send a $Delete$ notification message indicating that o_i got out 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 that intended to report the sensed data to base station is running after 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 record for o_i in $NMT(s_j) = \{\mathrm{N}_{i,j}, \mathrm{N}_{i,0}, \mathrm{N}_{i,1} \ldots \mathrm{N}_{i,x}\}\ x \neq j$, $\mathrm{N}_{i,x}$ is a notification messages for object o_i which detected by node s_x. In this case s_j will be the proxy node if it satisfied the election requirements i.e., maximum ID, residual energy, the current state…etc. if s_j loss this round or did not elected as a proxy node it will remove o_i record from $SOT(s_j)$ and remove $\mathrm{N}_{i,j}$ notification message from $NMT(s_j)$. Worthily noted 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 reporting model explain in section 4.



(a)Sensed Objects Table



(b) Notification Message Table

Figure 6: Basic Storage Tables

The 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.

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

Input: A mobile object o_i get in the range of s_j

```
$\textbf{int}\ x \leftarrow 0;$ // message ID
$\textbf{if}\ (isWithinMyRange(o_i))$
{
    $\textbf{if}\ (o_i \notin NMT(s_j))$ // no notification from other nodes
    {
        $m_x \leftarrow new\ \textbf{Message}();$ // sensed data object
        $m_x.MessageContent \leftarrow "data";$
        $m_x.isSent \leftarrow false;$
        $\textbf{if}\ (o_i \notin SOT(s_j))$ // the object has no SOT record
        {
            $r_i \leftarrow new\ \textbf{SensedObjRecord}();$ // create new record in SOT
            $r_i.MobileObject \leftarrow o_i;$
            $r_i.MessageTable.Add(m_x);$
            $SOT(s_j).\textbf{Add}(r_i);$
            $INM \leftarrow new\ \textbf{NotificationMessage}();$ // create insert notification
            $INM.MobileObject \leftarrow o_i;$
            $INM.NotificationType \leftarrow Notification.Insert;$
            $INM.Source \leftarrow s_j;$
            $INM.Data \leftarrow "data";$
            $NMT(s_j).Add(INM);$
            $\textbf{BroadcastInsertNotification}(INM, NT(s_j));$ // send the notification via notification tree
            $s_p \leftarrow SelectProxyNode(o_i);$
            $\textbf{if}(s_p \neq null)$
            {
```

```
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.     }// $if$ (! $isInMyNMT(mobj)$)
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**($DNM \neq null$)
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.   }
```

In line #26, the function $s_p.StartTracking()$ instruct the proxy node s_p to start the reporting process of 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 $r_i = \{m_0, m_0, m_0 \ldots m_x\}$ where m_i $(i \leq x)$ is the current sensed data(see Figure 5(a)). The proxy node s_p report the records in $SOT(s_p)$ sequentially. The records in $SOT(s_p)$ will be delated after reporting process is finished.

7. Results and Discussion

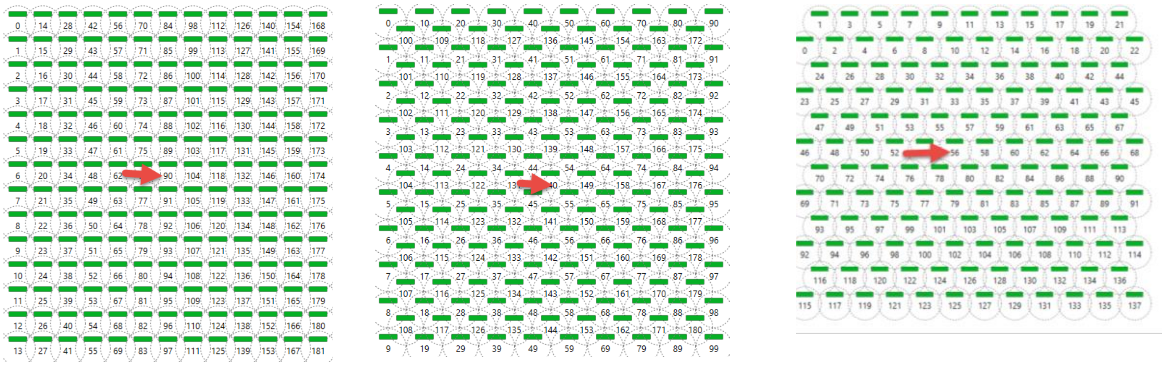
In this work, we assumed that the radio channel is symmetric such that the energy required to transmit a message of k bits from node x to node y is the same as energy required to transmit the same size message from node y to node x for a given signal to noise ratio (SNR). We assume a simple model called (First Order Radio Model). This model discussed in [14-17]. For this model, the energy dissipation to run the radio (E_{elec}) is 50 nJ/bit. E_{elec} is depends on the factors such as the digital coding, modulation, filtering, and spreading of the signal. The Free space model of transmitter amplifier is $\varepsilon_{fs} = 10\ pJ/bit/m^2$ and the Multi-path model of transmitter amplifier is

$\varepsilon_{mp} = 0.0013\ pJ/bit/m^4$. Both of Free space and Multi-path models are depending on the distance to the receiver and the acceptable bit-error rate. If the distance is less than a threshold $t_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}$ (meters), the free space (fs) model is used; otherwise, the multi path (mp) model is used, as shown in Equations (3, 4). A sensor node will consume $E_{Tx}(d, k)$ of energy to transmit $k\ bits$ size message over distance d, and consume $E_{Rx}(d, k)$ of energy to receive transmit $k\ bits$ size message.

$$E_{Tx}(d,k) = \begin{cases} (k.E_{elec}) + \left(k.\varepsilon_{fs}.d^2\right) & d < t_0 \\ (k.E_{elec}) + \left(k.\varepsilon_{mp}.d^4\right) & d \geq t_0 \end{cases} \tag{3}$$

$$E_{Rx}(k) = k.E_{elec} \tag{4}$$

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



(a) Grid coverage scheme 1[29]

(b) Grid coverage scheme 2[29]

(c) Zigzag Coverage scheme[28]

Figure 7: The network topologies used to simulate the performance of GHS. The arrow indicated the position of base station (sink node). The green progresspar indicates the residual energy. Here all nodes are full charged 0.5 Joule.

Grid Based Coverage

For Grid based, we have selected two algorithms provided in [29]. Grid Square Coverage version (1) and Grid Square Coverage version (2). The Grid Square Coverage (1) algorithms has 78% coverage efficiency while the Grid Square Coverage (2) has 73%.

1) Grid Coverage version (1)

Table3 shows the parameters we used to evaluate the performance of GHS in grid coverage version (1). The sensors deployed as show in Figure 7 (a).

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
ε_{amp}	0.001 pJ/bit/m^2
BS location	Inside
Network Topology	Grid Coverage 1

The evaluation results for using GHS on grid coverage version (1) are show in Figure 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 round). It indicates that the closed nodes to base station will die out first due to network load. As the communication rang of far node is limited, thus, 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 is4, which means more channels for leaders to forwards 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 send from a node A to node B is either consumes 76288 nJ or 63744 nJ .

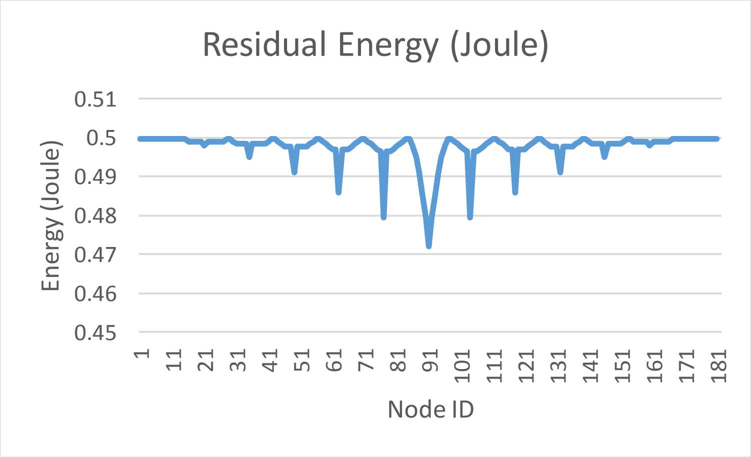
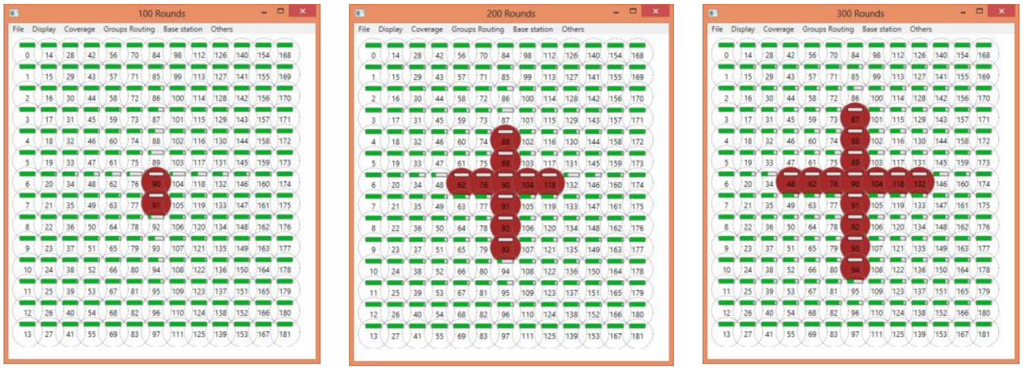


Figure 8: The residual energy of nodes after five rounds. We assume that the battery of each node is 0.5J. Network Topology is Grid version (1).



(a) 1.09% of nodes die out after 100 rounds.

(b) 5.4% of nodes die out after 200 rounds.

(c) 7.6% of nodes die out after 300 rounds.

Figure 9: The dead nodes (***brown***) after 100,200,300 rounds using Grid Coverage 1 topology. The initial energy is 0.5J/node with 50 nJ/bit dissipation to run the radio (E_{elec}). In each round, each sensor sends a packet 1024 bits. The green progresspar indicates the residual energy.

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).

Table 4: Simulation Parameters

Parameter	**Value**
Network size	$500 \times 500\ m^2$
Nodes	181
Radius	25 m
Data length	1024 bits
Initial energy	0.5 Joule
E_{elec}	50 nJ/bit
ε_{amp}	0.001 pJ/bit/m^2
BS location	Inside
Network Topology	Grid Coverage 2

In Figure 10, the residual energy (in joule) is explained after each node of 181 nodes sent five message of size 1024 bits (five round). Figure 11 explained the die out node after 100,200 and 300 round. In this coverage scheme, the number of nodes in each group is 2, which means there are two channels for leaders to forwards 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 send from a node A to node B is either consumes 62996.48 nJ or 64020.48 nJ .

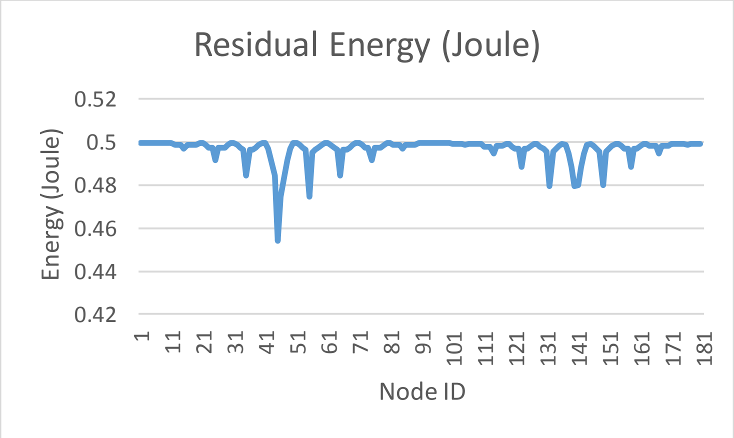
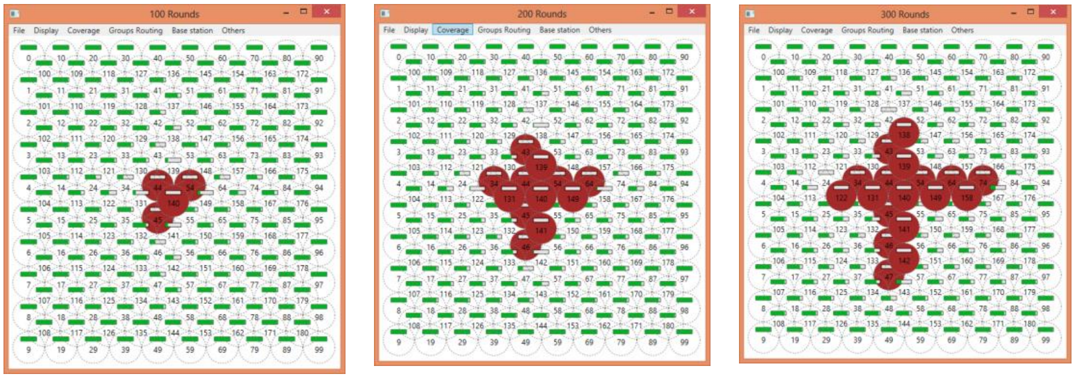


Figure 10: The residual energy of nodes after five rounds. We assume that the battery of each node is 0.5J. Network Topology is Grid version (2).



(a) 2.2% of nodes die out after 100 rounds. (b) 6.6% of nodes die out after 200 rounds. (c) 8.8% of nodes die out after 300 rounds.

Figure 11: The dead nodes (brown) after 100,200,300 rounds using Grid Coverage 2 topology. The initial energy is 0.5J/node with 50 nJ/bit dissipation to run the radio (E_{elec}). In each round, each sensor sends a packet 1024 bits. The green progresspar indicates the residual energy.

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 91%, as well as, it expands and covers the whole interest area with minimum number of nodes, while it generates a very small of coverage redundancy [28].Table 5 shows the parameters we used to evaluate the performance of GHS on zigzag coverage sachem [17]. The network topology for Zigzag shown in Figure 5(c).

Table 5: Simulation Parameters

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

In Figure 12, the residual energy (in joule) is explained after each node of 138 nodes sent five message of size 1024 bits (five round). It indicates that the closed nodes to base station will die out first due to network load. Due to limited communications rang of far node, thus, the close nodes to base station will act as routers for forwarding packets that transmitted form far nodes. Figure13 explained the amount of energy consumed by each node for the same number of round. In addition, Figure 14 explained the die out node after 100,200 and 300 round. In this coverage scheme, the number of nodes in each group is 3, which means there are three channels for leaders to forwards data. The average number of nodes in each path (the number of hops) is 4.6 when Radius is 25 m. The min distance for each group is 43 m and the max distance is 44 m, that is to say for any message ($1024\ bits$) to be send from a node A to node B is either consumes $70605.856nJ$ or $71069.7216\ nJ$.

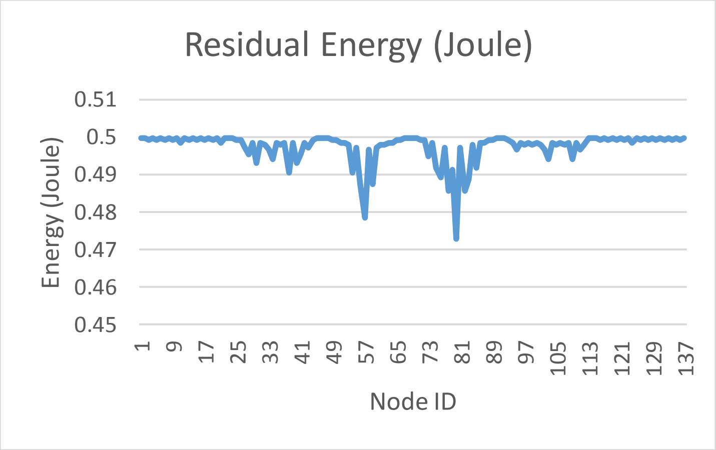


Figure 12: The residual energy of nodes after five rounds. We assume that the battery of each node is 0.5J. Network Topology is Zigzag**.**

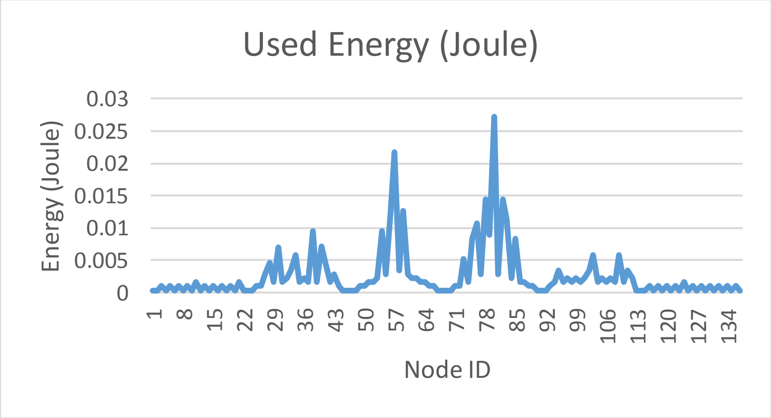
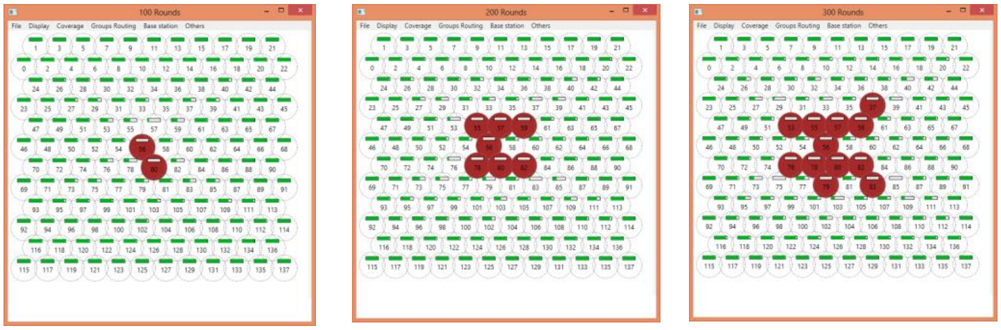


Figure 13: The consumed energy of nodes after five rounds. We assume that the battery of each node is 0.5J. Network Topology is Zigzag.



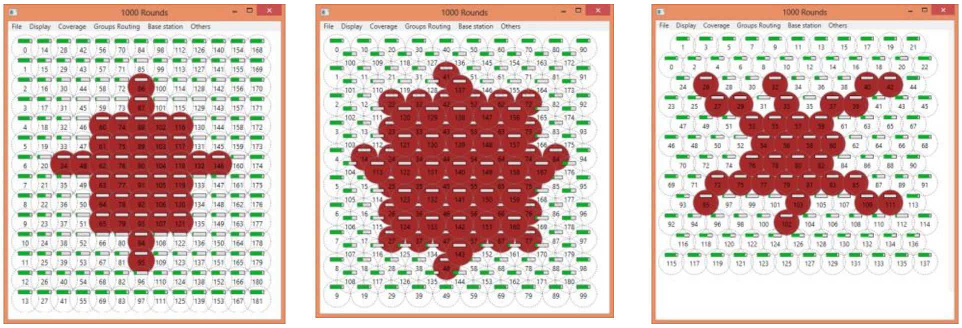
(a)1.4% of nodes die out after 100 rounds.

(b) 5% of nodes die out after 200 rounds.

(c)8.6% of nodes die out after 300 rounds.

Figure 14: The dead nodes (brown) after 100,200,300 rounds using Zigzag topology. The initial energy is 0.5J/node with 50 nJ/bit dissipation to run the radio

(E_{elec}). In each round, each sensor sends a packet 1024 bits. The green progresspar indicates the residual energy.



(a) Grid coverage (1). 20.87% die out nodes.

(b) Grid coverage(2). 38.12% die out nodes.

(c)Zigzage Coverage. 23.18% die nodes.

Figure 15: The dead nodes (brown)of three coverage schemes after 1000 rounds of GHS, the Battery of each sensor is 0.5 J. in each round each sensor send a packet 1024 bits. The energy dissipation to run the radio (E_{elec}) is 50 nJ/bit. The green progresspar indicates the residual energy.

Table 6: Comparison between GHS and others well know structure over different topologies and different initial energy.

Energy (J/node)	Coverage Protocol	Nodes	Structure	Round first node dies
0. 5	Grid	182	SC	67
			DC	80
			PTP	51
			GHS	93
	Zigzag	138	SC	77
			DC	86
			PTP	85
			GHS	107
	Random	100	SC	117
			DC	125
			PTP	121
			GHS	137
1.0	Grid	182	SC	130
			DC	163
			PTP	102
			GHS	186
	Zigzag	138	SC	140
			DC	170

			PTP	173
			GHS	190
	Random	100	SC	234
			DC	254
			PTP	243
			GHS	274

Notes:

- PTP= Peer-to-peer, DC= Dynamic Clustering and SC= Static Clustering.
- One round means that all nodes in network take a turn to send a message i.e., one round in Grid means 182 message.

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 sensor contains multiple number nodes and a multiple number of leaders, as well as, the node can belong to more than group. We assumed that the interested 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, as well as it maximizes the lifespan of nodes by minimizing energy consumption, balancing energy and generating a little redundant data.

For further research, we plan to extend our structure by studding the influence of nodes quick movement in the field. As well as we will study the object tracking deeply using this structure. The researchers are very welcome to develop energy efficient data routing algorithms and objects tracking algorithms using *GHS*.

Acknowledgment

The authors would like to thank the anonymous reviewers for the helpful comments and suggestions.

References

[1] X. Bai, Z. Yun, D. Xuan, T. Lai and W. Jia, “Optimal Patterns for Four-Connectivity and Full Coverage in Wireless Sensor Networks,” IEEE Transactions on Mobile Computing, 2008.

[2] M. Cardei, J. Wu, N. Lu and M. O. Pervaiz, “Maximum Network Lifetime with Adjustable Range,” IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob’05), 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, 2008, p. 3601. http://dx.doi.org/10.3390/s8063601.

[4] Nor Azlina Ab. Aziz, Kamarulzaman Ab. Aziz, And Wan Zakiah Wan Ismail “Coverage Strategies For Wireless Sensor Networks” World Academy Of Science, Engineering And Technology 26 2009.

[5] S. C.-H. Huang, S. Y. Chang, H.-C. Wu and P.-J. Wan, “Analysis and Design of Novel Randomized Broadcast Algorithm for Scalable Wireless Networks in the Interference Channels,” IEEE Transactions on Wireless Communications, Vol. 9, No. 7, 2010, pp. 2206-2215.

[6] J. Wu and S. Yang, “Coverage and Connectivity in Sensor Networks with Adjustable Ranges”, International Workshop on Mobile and Wireless Networking (MWN), Aug. 2004.

[7] M. Cardei, J. Wu, N. Lu, M.O. Pervaiz, “Maximum Network Lifetime with Adjustable Range”, IEEE Intl. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob'05), Aug. 2005.

[8] M. Cardei, M.T. Thai, Y. Li, and W. Wu, “Energy-efficient target coverage in wireless sensor networks”, In Proc. of IEEE Infocom, 2005.

[9] Ghiasi, Soheil, Ankur Srivastava, Xiaojian Yang, and Majid Sarrafzadeh. "Optimal energy aware clustering in sensor networks." Sensors 2, no. 7 (2002): 258-269.

[10] ABBASI, Ameer Ahmed; YOUNIS, Mohamed. A survey on clustering algorithms for wireless sensor networks. Computer communications, 2007, 30.14: 2826-2841.

[11] V. Kawadia, P.R. Kumar, Power control and clustering in Ad Hoc networks, in: Proceedings of IEEE INFOCOM, San Francisco, CA, March 2003.

[12] A.D. Amis, R. Prakash, T.H.P. Vuong, D.T. Huynh, Max-Min Dcluster formation in wireless Ad Hoc networks, in: Proceedings of IEEE INFOCOM, March 2000.

[13] A.B. McDonald, T. Znati, A mobility based framework for adaptive clustering in wireless ad-hoc networks, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1466–1487.

[14] Heinzelman, W.; Chandrakasan, A.; Balakrishnan, H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii, HI, USA, 2000; pp. 1–10.

[15] W. Heinzelman , A. P. Chandrakasan and H. Balakrishnan "An Application Specific Protocol Architecture for Wireless Microsensor Networks", IEEE Trans. Wireless Commun., 2002, vol 1, issue 4.pp. 660 – 670, DOI:10.1109/TWC.2002.804190.

[16] Akl, Robert, and Uttara Sawant. "Grid-based coordinated routing in wireless sensor networks." Consumer Communications and Networking Conference. CCNC. 2007. DOI: 10.1109/CCNC.2007.174.

[17] Lindsey, S.; Raghavendra, C. PEGASIS: Power-Efficient GAthering in Sensor Information Systems. In Proceedings of the IEEE Aerospace Conference, Los Angeles, MT, USA, 2002;pp. 1125–1130.

[18] Sheikhpour, Razieh, Sam Jabbehdari, and Ahmad Khadem-Zadeh. "Comparison of energy efficient clustering protocols in heterogeneous wireless sensor networks." International Journal of Advanced Science and Technology36 (2011): 27-40.

[19] Chen, Jong-Shin, et al. "Efficient cluster head selection methods for wireless sensor networks." journal of networks 5.8 (2010): 964-970.

[20] Kulik, J.; Heinzelman, W.R.; Balakrishnan, H. Negotiation-based protocols for disseminating information in wireless sensor networks. In Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99), Seattle, WA, USA, 15–19 August 1999; pp. 169-185.

[21] Intanagonwiwat, C.; Govindan, R.; Estrin, D. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, USA, November 2000; pp. 56-67.

[22] Rodoplu, V.; Ming, T.H. Minimum energy mobile wireless networks. IEEE J. Sel. Areas Comm. 1999, 17, 1333-1344.

[23] Li, L.; Halpern, J.Y. Minimum energy mobile wireless networks revisited. In Proceedings of IEEE International Conference on Communications (ICC'01), Helsinki, Finland, 11–14 June 2001; pp. 278-283.

[24] Wang, J.; Kim, J.-U.; Shu, L.; Niu, Y.; Lee, S. A Distance-Based Energy Aware Routing Algorithm for Wireless Sensor Networks. Sensors 2010, 10, 9493-9511.

[25] Razieh Sheikhpour, Razieh Sheikhpour, and Sam Jabbehdari Sam Jabbehdari. "A two-level cluster based routing protocol for wireless sensor networks."International Journal of Advanced Science and Technology 45 (2012): 19-30.

[26] Derogarian, Fardin, João Canas Ferreira, and Vítor Grade Tavares. "A Routing Protocol for WSN Based on the Implementation of Source Routing for Minimum Cost Forwarding Method." SENSORCOMM 2011, The Fifth International Conference on Sensor Technologies and Applications. 2011.

[27] Tang, Feilong, et al. "A chain-cluster based routing algorithm for wireless sensor networks." journal of intelligent manufacturing 23.4 (2012): 1305-1313.

[28] Hawbani, A., & Wang, X. "Zigzag Coverage Scheme Algorithm & Analysis for Wireless Sensor Networks". Network Protocols And Algorithms **2013**, 5 (4), 19-38 DOI: http://dx.doi.org/10.5296/npa.v5i4.4688.

[29] A.Hawbani, X.Wang, N. Husaini, S. Karmoshi "Grid Coverage Algorithm & Analyzing for wireless sensor networks". Network Protocols And Algorithms **2014**, vol.6 issue 4. DOI: http://dx.doi.org/10.5296/npa.v6i4.6449