

An Adaptive Concentric Clustering Scheme for Energy Efficient Routing in Wireless Sensor Networks

Yih-Chuan Lin

Department of Computer Science and Information
Engineering, National Formosa University, Yunlin, Taiwan
63201.
lyc@nfu.edu.tw

Kai-Wei Jiang

Institute of Electro-Optical and Materials Science, National
Formosa University, Yunlin, Taiwan 63201.
kid824@gmail.com

Abstract—This paper presents an adaptive concentric clustering scheme to construct a routing topology which can reduce the transmission distance between nodes in a wireless sensor network for gathering sensory data from a base station. Depending on the current location of base station, the proposed algorithm can divide the set of network nodes into several concentric clusters and establish individual routing chain within each cluster such that the transmission distance between neighboring nodes on the chain is reduced. In addition, the proposed algorithm can automatically select appropriate cluster heads for the concentric clusters according to the number of sensor nodes remained so far in each cluster, which can also shorten the transmission distance between neighboring cluster heads. By computer simulation, the results show that the proposed algorithm performs better than other similar protocols in terms of energy save and lifetime increase capabilities for wireless sensor networks with random deployment of sensor nodes.

Index Terms— wireless sensor networks; PEGASIS; concentric clustering; energy saving.

I. Introduction

Recent advances in wireless communications and micro-electro-mechanical systems (MEMS) have enabled the drastic progress of sensor networks technologies. By these small and low-cost sensory devices, which are equipped with sensing, computation and wireless communication capabilities, it becomes easier to construct a wireless sensor network (WSN) over physical spaces for monitoring environments, vehicles or

buildings etc. Most applications in wireless sensor network are targeting for gathering sensory data in real-time from some habitat environments, military battlefields, and humans for data managements and decision making for physical phenomena. Unlike other types of network, wireless sensor network has the capabilities of self-organization and automatic data collection of environment information. There are different developments on sensor networks in literature, such as the design of energy efficiency routing algorithm, sensor network coverage, internet security, object tracking and hardware architecture design. Interested readers can refer to [2][6] for more in-depth information.

Due to the natures of small size and simple design, these sensor nodes are constrained to have limited battery power, memory storage and computing power. Besides, replenishment of battery power resource should be a difficult job for a network composed of a large number of sensor nodes spread over a wide physical space. Energy-efficient routing protocol design is one of the most important problems in the networks of battery-powered wireless sensor nodes [11].

Intuitively, the amount of energy consumption does bear directly on radio transmission distance between sensor nodes. Many energy-efficient routing algorithms in literature for WSNs, such as Low Energy Adaptive Clustering Hierarchy (LEACH) [12], Power-Efficient GATHERing in Sensor Information Systems (PEGASIS) [9] and Enhanced PEGASIS (EP) [4], have used the principle of shortening radio transmission distance

during the discovery of routes for data forwards. In LEACH, the network uses random-clustering scheme to divide itself into a set of sub-networks (called clusters) and collect sensory data locally within each cluster. It can decrease both the frequency and distance that sensor nodes need to transmit to base station directly. But it has a critical problem that the clusters may possess significantly different number of nodes. The different cluster size problem will lead to some nodes spend more energy than the nodes in other clusters. Under such unbalance energy consumption situation, it will shorten the network lifetime. Moreover, it may also cause that no cluster head was selected due to the usage of random probability to determine the cluster head [3][8][12].

In PEGASIS, the network acts as a large cluster and all nodes in the network are formed into a form of chain for transmission. The transmission distance between neighboring nodes on the chain is much reduced and only one head-node is required to transmit data to the base station. For these reasons, PEGASIS can save more energy than LEACH. However, PEGASIS has the redundant data transmission problem, which would waste extra energy consumption for delivery in the circuitous routing chain to the base station [4]. Besides PEGASIS uses a greedy algorithm to construct the routing chain that would result in relative large transmission distance between some neighboring nodes on the chain. This situation would make these nodes died faster than other nodes and then shorten the network lifetime. Furthermore, the long transmission delay with which the data travels to the base station via the chain routing topology is another problem in PEGASIS [3][4][9].

Afterward, EP uses concentric clustering scheme to mitigate the redundant energy consumption of PEGASIS. From the base station, EP uses concentric clustering scheme to evenly divide the network nodes into several clusters with identical cluster size. Next, EP performs similar work as PEGASIS to construct chain in each cluster. The data transmission across clusters is performed in the order from the farthest cluster to the nearest cluster, and finally to the base station. With this mechanism of data transmission, EP [4] can

decrease redundant data transmission. However, EP has the same problem that transmission distance between nodes is too long whether in between cluster heads or general nodes within single cluster. One of the reasons is that the starting node in the chain construction and the selection rule of cluster heads are not suited for the network topology. EP uses greedy algorithm to perform chain construction, which starts at the farthest node from the base station in each cluster and progressively connects the remaining nodes one at a time using the greedy policy of shortest distance. Owing to the fixed selection rule, the starting node selected is not always at a good start point for chain construction within a cluster. Therefore, it may still result in relatively long transmission distances between neighboring nodes and redundant transmission phenomenon and thus makes these nodes dead rapidly.

In this paper, the main design concept is focused on the time-driven model [3][5][10]. We propose an algorithm “Boundary First PEGASIS” that can adapt the construction of routing topology by taking into account the position of base station and the current number of active nodes in each cluster. The approaches for the adaptability are to use vector projection approach which considering the location of base station, for constructing the network topology, and to design cluster head selection rule, which depend on the number of active nodes in the WSN, to construct the inter-head nodes data routes.

The system model on which the proposed algorithm is based can be represented as shown in Figure 1. When the sensor nodes have been deployed, the base station starts to broadcast an initial message to collect the individual status information of all nodes, and then all nodes reply responses respectively. Next the base station performs the cluster and chain formation and broadcasts the result to the network. Further, all nodes that receive the information from the base station can proceed their own initial setting to establish the initial routing topology, such as obtaining its cluster number and its order number on the chain for constructing communication path. After a certain period of time, the base station starts to broadcast the “*data collect*” command, and then

all nodes start to collect sensing data for the base station. This process can be regarded as a *round* (fixed time period) and the base station has to issue the data collect command every round of time. Once some nodes are died (the data collection process is failure), the base station can become aware of this situation through time out and then to broadcast Maintenance message. All nodes start to check whether its neighbor died or not and reconstruction the communication path. After that, all nodes proceed with data forwarding.

The remainder of this paper is organized as follows: Section 2 is devoted to the proposed algorithm. Section 3 provides the simulations results and evaluates the proposed algorithm. Finally, concludes are given in Section 4.

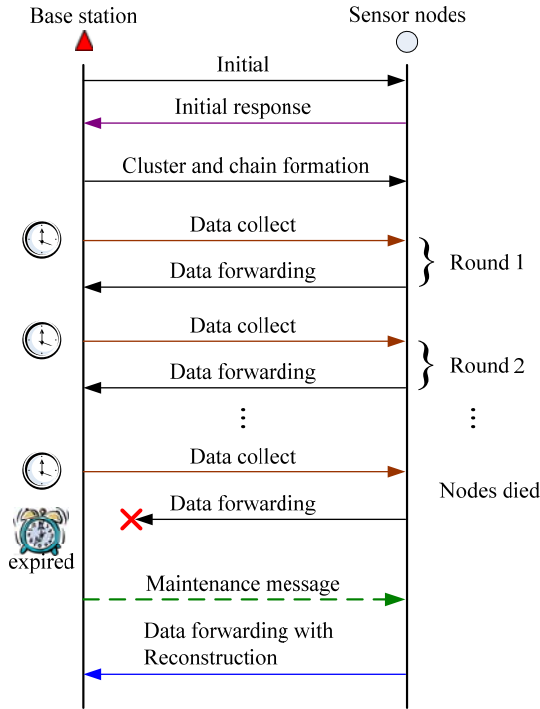


Figure 1 System operation process of BFP.

II. The Proposed Algorithm

The proposed algorithm has following assumptions:

- Sensor nodes know their own positions in the monitoring field.
- Sensor nodes can adjust its radio power level to communicate with other nodes or base station.
- The lower communication protocols of the WSN considered have no automatic acknowledge scheme. Reliable data transmission can be performed by the routing protocol itself if

require.

Figure 2 shows the flowchart of the proposed algorithm. It is named after PEGASIS to be Boundary First PEGASIS (BFP) due to its nature like PEGASIS. In this section, we describe the complete works that are needed to achieve the target requirements.

For the ease of explanation, Table 1 defines the notations that are used in this paper.

Let us first discuss that how the position of base station affects the routing topology to be constructed.

Table 1 Definition of notations

Notation	Definition
s_i	Sensor nodes in the network, $i = 1, 2, \dots, n$
S	Node set; $S = \{s_1, s_2, \dots, s_n\}$
BS	Base station
$d(s_i, BS)$	The distance between node s_i and BS
$C(s_i)$	Cluster number of s_i , $C(s_i) = 1, 2, \dots, c_{\max}$
CS_c	Cluster set; $CS_c = \{s \in S C(s) = c\}; c = 1, 2, \dots, c_{\max}$
$L_i(r)$	Cluster head in cluster i at the r -th round; the corresponding order number on the chain would be cluster head
$T_i(r)$	Total number of nodes in cluster i at the r -th round
$A(r)$	Total number of nodes in the network at the r -th round
$PN(r)$	Total number of clusters in the network at the r -th round
(x_M, y_M)	The location of center point (M) of the sensor field
(x_B, y_B)	The location of Base station
(x_{s_i}, y_{s_i})	The location of node s_i
$O_i(s)$	Order number of nodes on the chain in the cluster i ; $s \in CS_c$

A. Impact of the Location of Base station and Data Routing

The study of how to place base stations such that

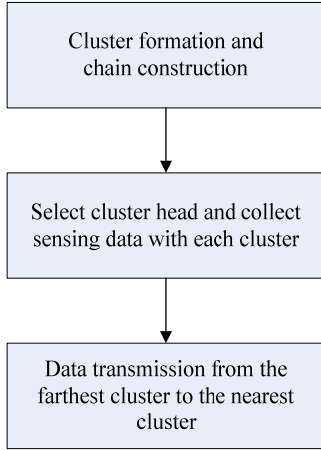


Figure 2 Flowchart of BFP.

the lifespan of the WSN can be maximized has ever been addressed and proved to be a NP-complete problem under some pre-specified conditions [1]. Researchers have used integer programming, computational geometry, local search and greedy heuristics to find sub-optimal solution [7].

The base station can be regarded as a member in the network, because all nodes have to transmit data to the base station. Apparently the energy consumption of nodes has indirect bearing on the location of base station; in other words the location of base station has impact on the data routing. We randomly deploy nodes to simulate EP and use different markers/colors to denote different clusters as show in Figure 3. To examine the network topology in cluster 2 (blue marker) in Figure 3 (a) and (b), we can see that under the same node distribution, once the location of base station is moved to a new place, the shape of cluster formation or network topology is changed correspondently. These variations can affect the performance of a routing algorithm. From the viewpoint of routing design, only uses concentric clustering is not enough. Our purpose is to construct a good network topology that can adapt with different locations of base station, as the example shown in Figure 4.

Specifically, for the example in Figure 5, we let all nodes in a cluster to project their position vectors \vec{s}_i onto a principal vector (\vec{v}). For each cluster of nodes, the principal vector is defined as follows,

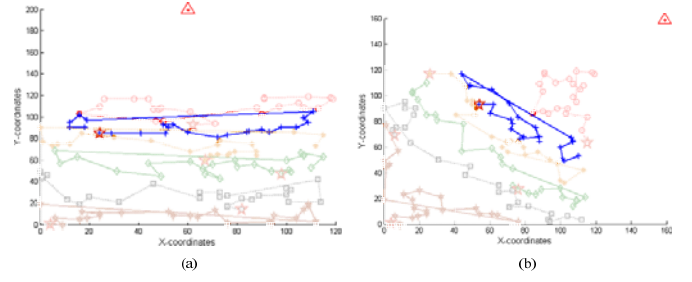


Figure 3 Illustration of the influence the location of base station on data routing

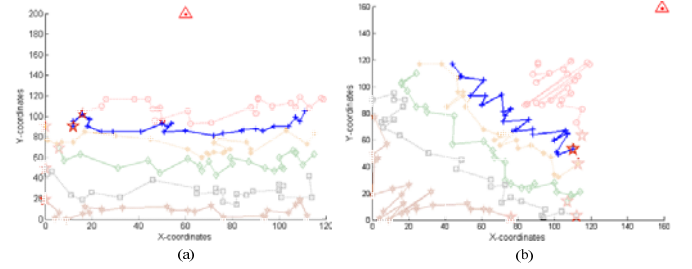


Figure 4 A Good network topology when confront with different environments.

$$\vec{v} = \begin{cases} (x_M, 0) & \text{if } x_a = 0 \\ (0, y_M) & \text{if } y_a = 0 \\ (-1, x_M/y_M) & \text{otherwise} \end{cases} \quad (1)$$

where x_a and y_a represent the x - and y -components of another vector \vec{a} that points to the base station from the position (x_M, y_M) , i.e.,

$$\vec{a} = (x_B - x_M, y_B - y_M) \quad (2)$$

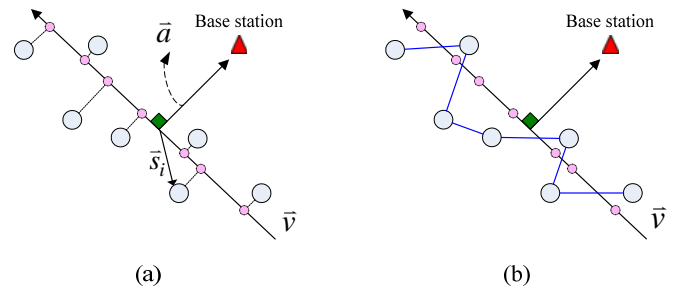


Figure 5 Using projection approach to construct the chain.

To construct the chain of nodes, the order of the nodes on the chain is arranged in consistent with the order of the corresponding projective values on the vector \bar{v} as shown in Figure 5 (b). Apparently, the chain construction process would changes with the positions of the base station; the network topology can be controlled to adapt to the different location of base station problem and can avoid the irregular topology.

B. The Cluster Head Selection Rule

Because cluster heads have to spend more energy than general nodes, therefore all nodes are taking turns to be cluster head and consume energy equally. A simple cluster head selection rule in EP which can achieve the purpose is

$$L_i(r) = r \bmod T_i(r) \quad (3)$$

It would perform well when the number of active nodes is same in each cluster, as shown in Figure 6. However, when nodes become inactive due to drained out of battery energy or other reasons, the cluster sizes among the clusters would become different and cause the different cluster size problem, which lead to a irregular forwarding path and thus more energy dissipation. The maximal transmission distance can be the length/width of the network field in the extreme case. Apparently it is serious energy consumption.

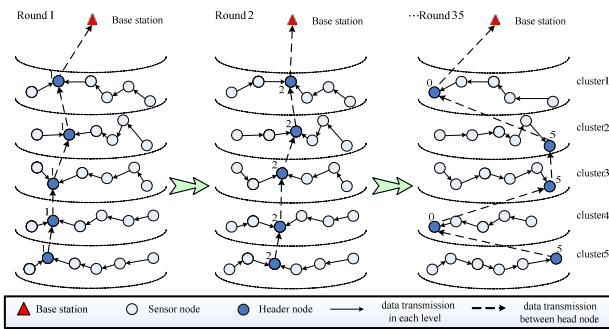


Figure 6 Illustration of the different cluster size problem.

Therefore, we propose a selection rule for cluster heads that can adapt to the different number of active nodes in clusters. This rule is called average-cluster-size based rule (ACSB for short) and (4) and (5) define its operation. In order to keep the rotation feature and maintain the

transmission consistently, we let the average of the cluster size over all the clusters to control the behavior of the selection rule. The purposes of ACSB are to reduce the transmission distance between cluster heads and let all nodes have chance to be cluster head.

$$L = r \bmod \left\lfloor \frac{A(r)}{PN(r)} \right\rfloor \quad (4)$$

$$L_i(r) = \begin{cases} L & L < T_i(r) \\ T_i(r) - 1 & L \geq T_i(r) \end{cases} \quad (5)$$

Eq. (5) is used to see if the number of node in the cluster is less than the order number of the next cluster head. In this case, the order number of cluster head $L_i(r)$ is set to $T_i(r)-1$. Let us see an example in Figure 7, in the $(r+1)$ -th round (Figure 7 (b)), cluster 2 can not select a suitable node to be cluster head because of the fewer number of nodes than the average cluster size. At this time, nodes can use equation (5) to select the last sensor node in the cluster as the cluster head in this round.

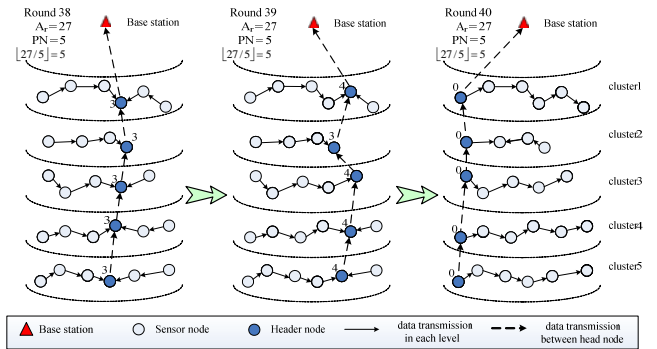


Figure 7 Illustration of the operation in ACSB.

C. BFP: Boundary First PEGASIS

Our WSN model requires doing the following tasks of BFP for constructing a routing topology.

- **Cluster and chain formation**

In this phase, the tasks about cluster and chain formation are computed by the base station. After the base station collects the individual status information of all nodes deployed in the monitoring space, the base station needs to do the clusters and chains construction task below:

Algorithm 1: Cluster formation

Input: Nodes s_1, s_2, \dots, s_n and c_{\max}

Output: $C(s_i); i = 1, 2, \dots, n$

1. Use the Euclidean distance formula to calculate:

$$d_1 = d(s_1, BS), d_2 = d(s_2, BS), \dots, d_n = d(s_n, BS)$$

and store them in a distance table.

2. Sort the distance table and classify it into c_{\max} cluster

sets which each cluster has $\left\lceil \frac{n}{c_{\max}} \right\rceil$ nodes.

3. For each cluster i , the base station needs to determine a pair of cluster boundary distance value: b_{i-1} and b_i . Hence for all cluster sets together, there exists a sequence: $\{b_0, b_1, \dots, b_{c_{\max}-1}, b_{c_{\max}}\}$.

These results can be tabulated as a cluster assignment table as like Table 2. Through base station to broadcast the cluster assignment table and the location information of the base station, each receiving node can realize its cluster number. Further, according to the location information of nodes, the base station can construct the relative localization and center point of the sensor field, as shown in Figure 8.

Table 2 Cluster assignment table

Cluster number	Distance between nodes and base station (Unit: M)
1	$b_0 \leq d(s_i, BS) < b_1$
2	$b_1 \leq d(s_i, BS) < b_2$
3	$b_2 \leq d(s_i, BS) < b_3$

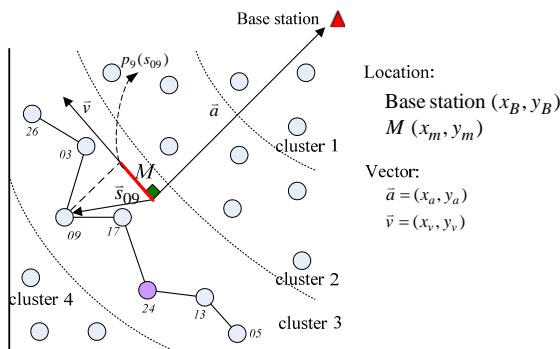


Figure 8 Illustration of projection approach to construct the chain.

For each cluster set CS_c where $CS_c = \{s \in S | C(s) = c\}$, We can use vector

projection approach to perform chain construction in each cluster which can avoid the problem in EP and shorten the transmission between nodes. The chain construction is also performed by the base station and does following steps through algorithm 2:

Algorithm 2: Chain construction

Input: All cluster sets $CS_c; c = 1, 2, \dots, c_{\max}$.

Output: $O_c(s), s \in CS_c, c = 0, 1, 2, \dots, c_{\max}$.

1. Calculate a spatial vector \vec{a} which connect the center point M with the base station and calculate the vertical vector \vec{v} which perpendicular to vector \vec{a} .

2. For each node in CS_c , the base station calculates a vector \vec{s}_i which connects the center point M to node s_i . Next, uses equation (6) to calculate the $p_i(s_i)$ which indicates the projection of \vec{s}_i onto \vec{v} .

$$p_i(s_i) = \frac{\vec{s}_i \cdot \vec{v}}{|\vec{v}|} \quad (6)$$

3. For the cluster set CS_c , collect all the projection values in the cluster as a projection set $P_c = \{p_1(s_1), p_2(s_2), \dots, p_{|CS_c|}(s_{|CS_c|})\}$.

4. Sort P_c and for each node in CS_c , assign an appropriate integer from $\{0, 1, 2, \dots, |CS_c| - 1\}$ to $O_c(s)$, $s \in CS_c$ such that $O_c(s_i) < O_c(s_j)$ if $p(s_i) < p(s_j)$ for any pair of different nodes $s_i, s_j \in CS_c; s_i \neq s_j$.

For example, as shown in Figure 8, let us look at node 09 in cluster 3: the base station calculates vector \vec{s}_{09} and $p(s_{09})$ to indicate the projection length (red line) of \vec{s}_{09} onto \vec{v} . Other nodes in the cluster are treated with the same operation. All results of operation can be tabulated as a table. Next, sort the table with column P_c as like shown in Table 3. The order of nodes in the Table 3 is the result about the order of nodes on the chain, as shown in Figure 8.

After these steps, the base station broadcasts the results about the sequence: $\{b_0, b_1, \dots, b_{c_{\max}-1}, b_{c_{\max}}\}$ and the sorted P_c to each cluster sequentially. When a node receives these information, it can set related parameters for constructing routing. These parameters are

needed to maintain by each node, as tabulated in Table 4. For each receiving node s_i in the cluster c uses Algorithm 3 to set related parameters.

Table 3 Sort the projection length record with column P_3

s_i	(x_{s_i}, y_{s_i})	P_3
05	(100,34)	-46.7
13	(94,50)	-31.1
24	(87,56)	-21.9
17	(61,68)	4.9
09	(51,85)	24.0
03	(41,85)	31.1
26	(46,107)	43.1

Table 4 Maintain parameters for a node

Variables	Definition
C_{num}	Cluster number
O_{num}	Order number on the chain
$neighbor_L$	Left-sided neighbor which its order number is $O_{num} - 1$
$neighbor_R$	Right-sided neighbor which its order number is $O_{num} + 1$
CZ	Cluster size of itself; initial value is $ P_c $
TN	Total number of nodes in the network; initial value is n
TC	Total number of clusters in the network; initial value is C_{max}
$L_i(r)$	Cluster head in cluster i at the r -th round; the corresponding order number on the chain would be cluster head. The first time to receive “data collect” command, $L_i(r) = 1$

Algorithm 3: Set related parameters for constructing routing

Input: $\{b_0, b_1, \dots, b_{c_{max}-1}, b_{c_{max}}\}$, sorted P_c and the location of base station

Output: Each node have to set parameters such as C_{num} , O_{num} , $neighbor_L$, $neighbor_R$, and CZ .

1. Sets its cluster number. If $b_{k-1} \leq d(s_i, BS) < b_k$, then sets $C_{num} = k$.

2. Sets O_{num} according to the index from $\{0, 1, 2, \dots, |P_c| - 1\}$ in the P_c . Then sets $CZ = |P_c|$.

3. Constructs/Sets communication route and collect the location information of neighbor nodes:

Switch (O_{num})

Case 0: with the right-side neighbor node (order number = $O_{num} + 1$). Sets $neighbor_R$.

Case CZ-1: with the left-side neighbor node (order number = $O_{num} - 1$). Sets $neighbor_L$.

Default: with the two-sided neighbor nodes (order number = $O_{num} + 1$ and $O_{num} - 1$). Sets $neighbor_R$ and $neighbor_L$.

This information has aids for maintaining an accuracy data transmission routes. In addition, each node can adjust a suitable power level while transmitting sensing data according to the location information of its neighbor nodes.

• Data forwarding

After base station broadcasts the “data collect” command which includes values in variables TN , TC and r . The data collection process is performed first by all clusters concurrently, and then cluster heads transmit their own data orderly from the farthest cluster to the nearest cluster, which is similar to EP. For the ease of discussion, the data forwarding can be divided into two parts: the process of data transmission only happened within the cluster, called *intra communication*, whereas the data transmission among clusters is called *inter communication*. The intra communication for each cluster performs Algorithm 4:

Algorithm 4: Intra communication

Input: Data collect command, TN , TC and r .

Output: Aggregation data

1. Update $T_i(r) = CZ$; $A(r) = TN$; $PN(r) = TC$

2. For each node in the c -th cluster, to use the cluster selection rule to calculate $L_c(r)$ in each round.

3. For each cluster i ,

If $L_c(r) \neq O_{num}$

Performs *general node data collect procedure*, as shown in Figure 9.

else

Performs *cluster head data collect procedure*, as shown in Figure 10.

End if


```

Switch ( $O_{num}$ )
Case 0:
  Initiates a counter:  $counter_{inter} = 0$ .
   $counter_{inter} = counter_{inter} + 1$ .
  Transmits  $counter_{inter}$  and sensing data to
   $neighbor\_R$ .
Case CZ-1:
  Initiates a counter:  $counter_{inter} = 0$ .
   $counter_{inter} = counter_{inter} + 1$ .
  Transmits  $counter_{inter}$  and sensing data to
   $neighbor\_L$ .
Default:
  If  $L_c(r) > O_{num}$ 
    Waits to receives  $counter_{inter}$  and sensing
    data from  $neighbor\_L$ .
     $counter_{inter} = counter_{inter} + 1$ .
    Then to perform data fusion and transmits
     $counter_{inter}$  and data to  $neighbor\_R$ .
  else
    Waits to receives  $counter_{inter}$  and sensing data
    from  $neighbor\_R$ .
     $counter_{inter} = counter_{inter} + 1$ .
    Then to perform data fusion and transmits
     $counter_{inter}$  and data to  $neighbor\_L$ .
  End if

```

Figure 9 General node-data collect procedure.

```

1. Switch ( $O_{num}$ )
Case 0:
  Receives  $counter_{inter}$  and sensing data from
   $neighbor\_R$ , and then to perform data fusion.
   $CZ\_temp = counter_{inter} + 1$ 
Case CZ-1:
  Receives  $counter_{inter}$  and sensing data from
   $neighbor\_L$ , and then to perform data fusion.
   $CZ\_temp = counter_{inter} + 1$ 
Default:
  Receives  $counter_{inter}$  and sensing data from
   $neighbor\_R$ .
   $CZ\_temp = counter_{inter}$ 
  Receives  $counter_{inter}$  and sensing data from
   $neighbor\_L$ 
   $CZ\_temp = CZ\_temp + counter_{inter}$ 
  Then performs data fusion.
2. If  $CZ \neq CZ\_temp$ 
   $CZ = CZ\_temp$ 
  Transmits  $CZ$  to its two-side neighbors
  End if

When a node receives  $CZ$ , it has to update  $CZ$ 
and transmits it to its neighbor. Therefore, each
node can maintain accurate  $CZ$  sequentially.

```

Figure 10 Cluster head-data collect procedure.

Through these steps, the cluster head can collect all sensing data in its cluster.

By Algorithm 5, cluster heads can relay all sensing data in the network to the base station and finish this round accordingly. Next all nodes wait to receive the data collect command and perform data collection. When the data collection process is failed (by timer expiration

in BS), it means the network has some nodes died. The base station can broadcast a Maintenance message to let all nodes to check whether its neighbor died or not and re-establish the broken link if necessary. For each receiving node in the c -th cluster, Algorithm 6 (shown on the next page) describes the process that need to be done for the case.

Algorithm 5: Inter communication

Input: Data collect command and r .

Output: Aggregation data

1. For each cluster head, its order number is $L_c(r)$

If $TC == 1$, then

Transmits data, TC and TN to the base station.

else if $C_{num} == TC$

Initiates two counters: $counter_{cluster} = 0$ and $counter_{total} = 0$.

Performs *inter-data collect procedure*, as shown in Figure 11.

else

Performs *inter-data collect procedure*.

End if

```

If  $C_{num} \neq TC$ 
  waits to receive  $counter_{cluster}$ ,  $counter_{total}$ 
  and data from the head node of cluster
   $C_{num} + 1$ .
End if

 $counter_{cluster} = counter_{cluster} + 1$ 
 $counter_{total} = counter_{total} + CZ$ 

Switch ( $C_{num}$ )
Case  $TC$ :
  Transmits  $counter_{cluster}$ ,  $counter_{total}$ 
  and data to the cluster head of  $C_{num} - 1$ 
Case 1:
   $TN = counter_{total}$ 
   $TC = counter_{cluster}$ 
  Performs data fusion and transmits data,  $TN$  and  $TC$ 
  to the base
  station
Default:
  Performs data fusion and transmits  $counter_{cluster}$ ,
   $counter_{total}$  and data to the cluster
  head of  $C_{num} - 1$ 

```

Figure 11 Inter-data collect procedure.

In Algorithm 6, the cluster number is needed to update for accurate data forwarding. Next the same principle of Step 2 in Cluster head-data collect procedure is used to inform other nodes in the cluster about the current cluster number. Afterward, the base station can collect data correctly until all

nodes died.

Algorithm 6: Topology maintenance

Input: Maintenance message.

Output: Update a accurate order number, $neighbor_R$ and $neighbor_L$.

1. To check its neighbor died or not.

Switch (O_{num})

Case 0:

Performs intra communication (Algorithm 4) and inter communication (Algorithm 5).

Case CZ-1:

Performs *maintain information procedure*, as shown in Figure 12.

Sets $neighbor_L$.

Performs intra communication and inter communication.

Default:

Performs *maintain information procedure*

Sets $neighbor_L$.

Sets a message: $INFO_value = x$ and then transmits it to inform its right neighbor. Through this step, a node can realize itself is the boundary node (right side) or not. Therefore, if the cluster head in this round is $L_c(r) = CZ - 1$, the new boundary node has to replace to be the cluster head.

Performs intra communication and inter communication.

2. When a node receiving the inform message, to update its order number = $O_{num} - INFO_value$.

If its neighbor died, existing x death nodes which their order number $< O_{num}$, then its order number is reconstructed as $O_{num} - x$. Constructs communication route and collects the location information of neighbor node with order number = $O_{num} - 1$.

Figure 12 Maintain information procedure.

III. Simulation Result

In our simulation, the formal radio mode: first order radio model [4][9][12] is used to evaluate the performance. We simulated in a 120m x 120m sensor field and randomly deployed 144 nodes (in the EP and BFP, we divided the network into six clusters which each cluster had 24 nodes). The initial energy for each node is 0.5 J and 50 nJ for the energy consumption of transmitter/receiver unit; 100 pJ for the energy consumption of transmitter amplifier. The packet length is fixed at 2000 bits and needs 5 nJ for the energy consumption of data aggregation. In each simulated algorithm, we

perform 1000 times and average the results of the outcomes. If the base station is located at the fixed position (60,200), the simulation results are shown in Figures 13, 14 and 15.

Figure 13 shows the total residual energy in each round, we can see that BFP has the best performance in this environment and it performs better than EP and PEGASIS. Besides, we can see that when BFP uses the ACSB rule, it can obtain more energy conservation than the one that uses the original rule.

Figure 14 shows the average transmission distance of each node. Because the head node in PEGASIS transmits data to base station directly, therefore the average transmission distance of nodes at each round to routing data to the base station is shorter than BFP and EP. On the contrary in BFP, cluster heads transmits data to base station through multi-hop manner. For this reason, it can preserve more energy as shown in Figure 13.

Furthermore, Figure 14 also shows that the average transmission distance in EP is longer than BFP; it means the intra communication in EP is longer than BFP. Besides the amplitude of range in EP is higher than BFP; this implies the inter communication in EP is longer than BFP. Moreover, in this figure we can see that the curve of ACSB is smooth than the original rule. Apparently ACSB can adapt to the active nodes to shorten the transmission distance between cluster heads. Therefore ACSB can preserve more energy.

To compare with PEGASIS, we use equation (7) to examine the improvement rate of BFP and EP in each round.

$$rate(r) = \frac{resd_{B/E}(r) - resd_P(r)}{resd_P(r)} \times 100 \quad (7)$$

where $resd_{B/E}(r)$ represents the total residual energy in BFP/EP at r -th round; $resd_P(r)$ represents the total residual energy in PEGASIS at r -th round. Figure 15 shows the result. As we see the improvement rate in BFP has about 43% in 1800 round; ACSB has about 48% in 1800 round. These simulation results have verified the facts that BFP can solve the problem in EP and that ACSB cluster head selection rules can outperform the selection rule in PEGASIS.

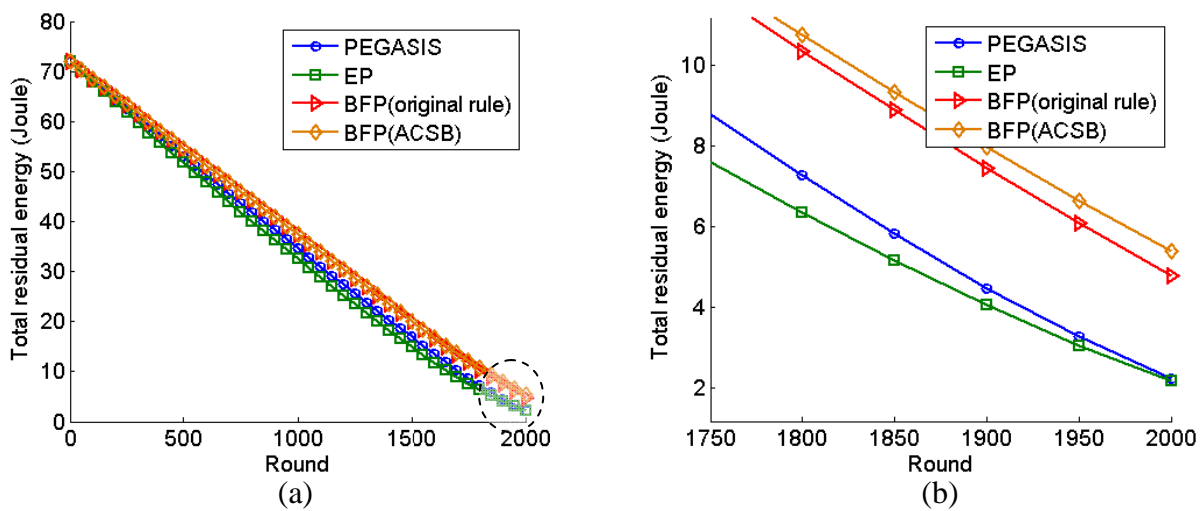


Figure 13 Total residual energy of each node. We magnify partial (a) to (b).

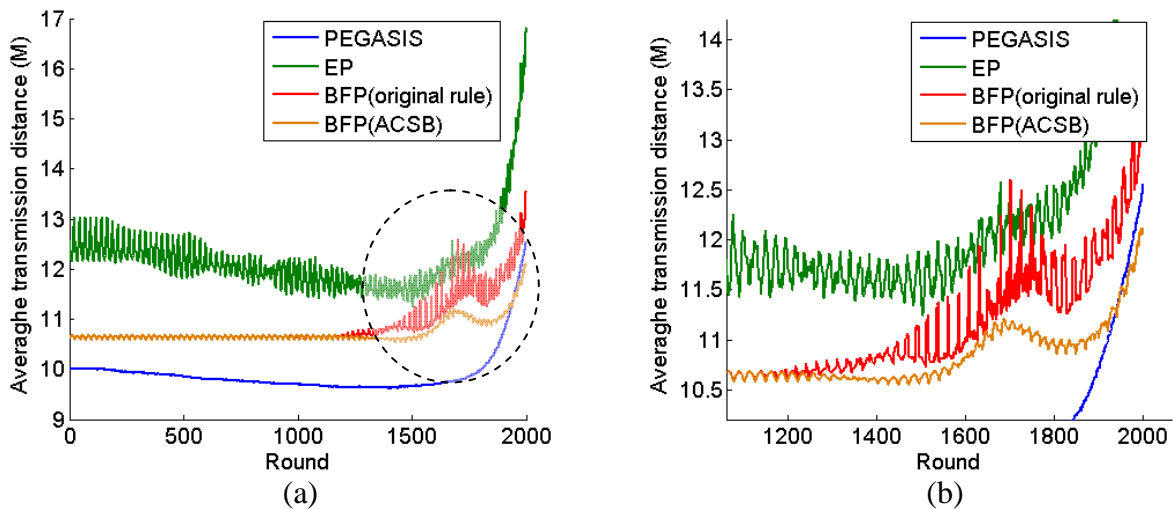


Figure 14 Average transmission distance of each node. We magnify partial (a) to (b).

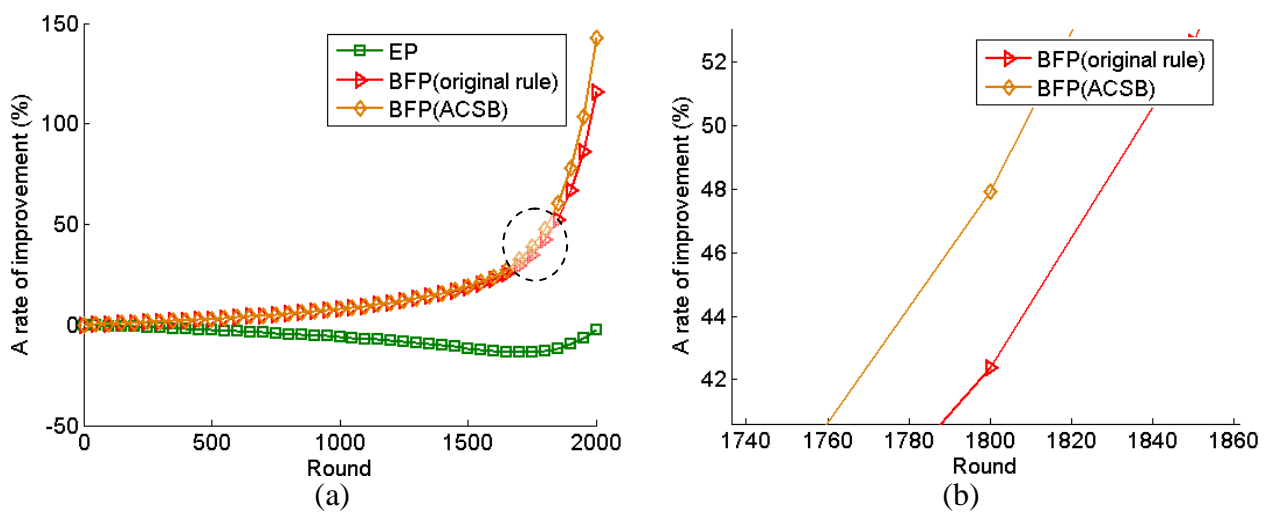


Figure 15 Compare with PEGASIS, a rate of improvement in EP and BFP. We magnify partial (a) to (b).

Next, we randomly test 100 different positions of base station in the considered space to gather sensory data for a fixed deployment of 144 sensor nodes and measure the network lifetimes (in terms of round). As like shown in Figure 16, these positions are restricted within in the red-square area. In such simulation environment, Figure 17 and 18 shows the first node die and all nodes die respectively, the horizontal coordinate represents the distance between different positions of base station and the center point of sensor filed. In the most cases, the performance of the proposed algorithm can perform better than EP and PEGASIS. However, according to our observation, the performance of the proposed algorithm decreases when the position is located within the sensor field.

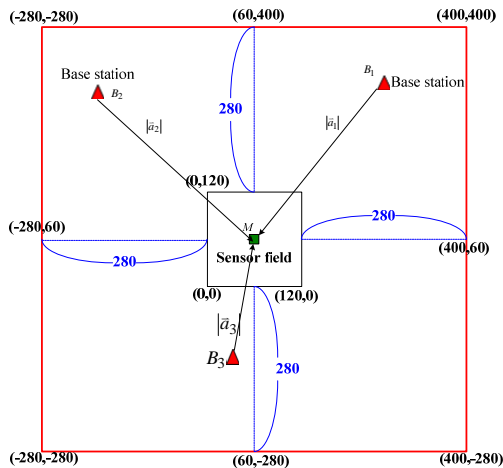


Figure 16 Illustration of random deploy different position of base station.

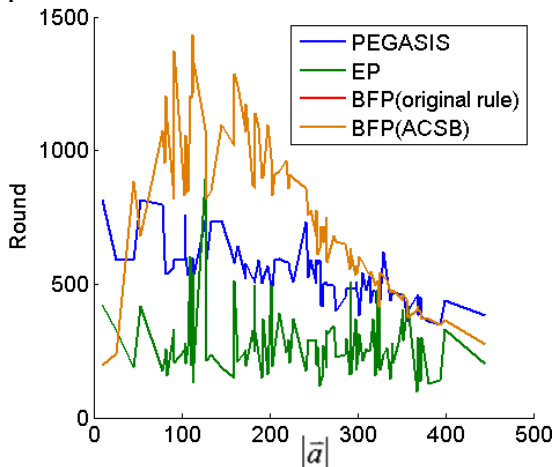


Figure 17 The network lifetime when the first node die. The curves of the proposed algorithm are overlapped. Therefore, only three curves are displayed in this figure.

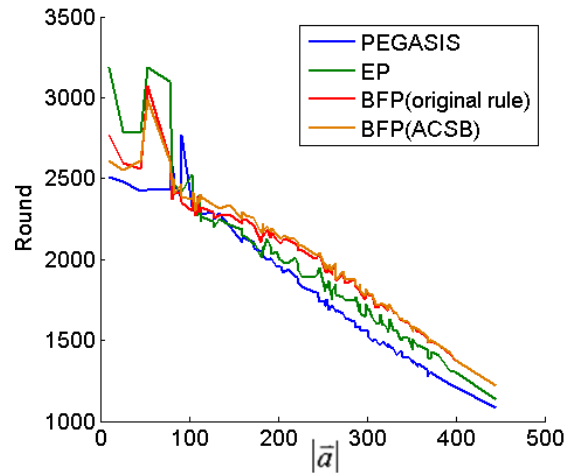


Figure 18 The network lifetime when the all nodes die. In order to display a clear result, so the improvement on the ACSB cluster head selection rule is not obvious (the curve of width is thick).

IV. Conclusion

In this paper, we propose schemes that can adapt the construction of WSN routing topology to the positions of base station and consumes energy conservatively when aggregating sensory data orderly within each cluster and across cluster heads. For different base station's positions, the proposed algorithm can shorten the distance and control the network topology through the vector projection approach. When the numbers of active nodes among the clusters are different, the proposed head-node selection rule can adapt to this dynamic phenomenon and further prevent the extended crisscross transmission phenomenon. By simulation, the proposed schemes can be energy-efficient and be adaptive when confront with different position of base station.

Unavoidably, the transmission mechanism of concentric clustering scheme has a common problem, that is, nodes which are within the nearest cluster to the base station would exhausted more quickly due to their heavy multi-hop relays. A simple way to solve this problem is to change the location of base station periodically. In such situation, the proposed algorithm can be used adaptively for different base station placements.

REFERENCE

[1] A. Efrat, S. Har-Peled and J.S.B. Mitchell, "Approximation algorithms for two optimal

- location problems in sensor networks,” 2005 2nd International Conference on Broadband Networks, vol. 1, pp.714-723, October 2005
- [2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “A Survey on Sensor Networks,” *IEEE Communications Magazine*, vol. 40 no. 8, pp.102-114, August 2002.
- [3] J. N. Al-Karaki, A. E. Kamal, “Routing Techniques in Wireless Sensor Networks: A Survey,” *IEEE Wireless Communications*, vol. 11 no. 6, pp.6-28, December 2004.
- [4] S. M. Jung, Y. J. Han and T. M. Chung, “The Concentric Clustering Scheme for Efficient Energy Consumption in the PEGASIS,” *The ninth International Conference on Advanced Communication Technology*, vol. 1, pp.260-265, February 2007.
- [5] K. Holger, A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, Chichester, England: Wiley, 2005.
- [6] K. Romer, F. Mattern, “The Design Space of Wireless Sensor Networks,” *IEEE Wireless Communication*, vol. 11 no. 6, pp.54-61, December 2004.
- [7] K. Akkaya, M. Younis, and M. Youssef, “Positioning of Base Stations in Wireless Sensor Networks,” *IEEE Communications Magazine*, vol. 45, pp.96-102, April .2007.
- [8] C. M. Liu, C. H. Lee and L. C. Wang, “Distributed Clustering Algorithms for Data-gathering in Wireless Mobile Sensor Networks,” *Journal of Parallel and Distributed Computing*, vol. 67 no. 11, pp.1187-1120, November 2007.
- [9] S. Lindsey and C. S. Raghavendra, “PEGASIS: Power-Efficient GATHERing in Sensor Information Systems,” *IEEE Aerospace Conference Proceedings 2002*, vol. 3, pp.1125-1130, 2002.
- [10] S. Tilak, Nael B. Abu-Ghazaleh and W. Heinzelman, “A Taxonomy of Wireless Micro-sensor Network Models,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6 no. 2, pp.28-36, April 2002.
- [11] S. Mahfoudh and P. Minet, “Survey of Energy Efficient Strategies in Wireless Ad Hoc and Sensor Networks,” *IEEE Seventh International Conference on Networking*, pp.1-7, April 2008.
- [12] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, “Energy-efficient Communication Protocol for Wireless Microsensor Networks,” *IEEE Proceedings of the Annual Hawaii International Conference on System Sciences*, pp.1-10, January 2000.