Study on Aggregation Tree Construction Based on Grid

Jian Shu
Nanchang HangKong University,China
Email: shujian@jxjt.gov.cn

Yebin Chen, Linlan Liu, Sheng Zhang and Jun Li
Nanchang HangKong University,China
Email : chenhb46@163.com, linda_cn68@yahoo.com, zwxzs168@126.com, tianylijun@163.com

***Abstract*—Data fusion is one of the key techniques in Wireless sensor network (WSN). Data fusion can reduce quantity of data transmission in the network, extend network lifetime by reducing energy consumption, and improve efficiency of bandwidth utilization. At present, researches on data fusion in WSN mainly fall into the following aspects: aggregation tree construction and data processing. This paper proposes an aggregation tree construction method based on grid named ATCBG which makes some improvements over GROUP. The results of simulation experiment show that the average energy consumption of ATCBG is evidently lower than GROUP, and the lifetime of the network is much longer than GROUP before the emergence of node death.**

***Index Terms*—Wireless Sensor Network, data fusion, aggregation tree**

I. INTRODUCTION

In Wireless Sensor Network (WSN), nodes are usually battery-powered. When the energy of the battery decreased below a certain threshold, the node will lose its ability and be died. The efficiency and reliability will be decreased with the increment of dead nodes [1]. In practice, a large number of nodes are usually deployed randomly by airplane in particular area, some of which are not available for human. So it is impossible to supplement energy by changing battery for nodes. The lifetime of WSN is greatly depends on energy. How to use the limited energy effectively is one of the goals of designing WSN.

The energy consumption modules of the node include wireless communication module, processor module and sensing module. With advances in integrated circuit techniques, energy consumption in processor and sensing module becomes lower and lower. For sensor nodes, energy is mainly consumed by wireless communication module [2].

WSN is a data-centric network. What the user cares for is the area where the event occurs, but not caring about which node detected the event. In WSN, the node is prone to failure especially in harsh environment. To make the network have the ability of fault tolerance, a large number of redundant nodes should be deployed which makes data collected by nodes redundant. Compared with only sending fused data to the user, sending all the raw data to the user can't get more information. And the sending of all the raw data will lead to more data transmitting in the network, so it will increase data collision and more energy consumption.

Data fusion can be implemented by using the computing and storage resource of the node, which can reduce data transmission, improve efficiency of bandwidth utilization, and extend network lifetime by reducing energy consumption. But traditional data fusion algorithms can't be applied in WSN directly for the constraint of the computing and storage resource. Furthermore, WSN is application dependent network, so it is difficult to figure out a data fusion technique that suits for all applications.

In WSN, there are following data fusion techniques: aggregation tree related techniques and data processing according to actual applications. What the aggregation tree related data fusion techniques care for is how to design the route for transmitting data to make the data be fused efficiently, and so the amount of data transmission is reduced.

This paper studies on the aggregation tree construction technique (aggregation tree construction based on grid, ATCBG) which aims at fusing the periodical data collected by the whole network nodes in event-driven WSN. The ATCBG is an improvement over GROUP [3, 4].

The remainder of this paper is organized as follows: Section II introduces some studies on aggregation tree construction, and section III describes the design of ATCBG. The simulation result will be shown in section IV. Section V is the conclusion of the paper.

II. REALTED WORKS

The aggregation tree construction techniques vary with the applications of WSN. Aggregation tree construction techniques introduced in this section is classified into the

This paper is sponsored by National Nature Science Foundation (No.60773055), Jiangxi Nature Science Foundation (No.2007GZS1501, 2009JX02563), Nanchang Hangkong University Postgraduate Innovation Foundation (No.YC2008015, YC2008017).

doi:10.4304/jcp.5.6.852-860

following categories: aggregation tree in query based WSN, aggregation tree for event data in event-driven WSN, and aggregation tree construction for periodical data in large scale WSN.

A. Aggregation Tree Construction in Query Based WSN

In query based WSN, nodes will not report information to the user initiatively. The node will transmit data to the user only when the data detected matches with the interest information required by the user. One of the representative aggregation tree algorithms in query based WSN is directed diffusion [5] (DD). The main idea of DD is that the node data is named by value-attribute pair and all the network information is for named data. DD can be divided into three phases: periodical interest propagation, gradients set up and path reinforcement.

B. Aggregation Tree for Event Data in Event-driven WSN

In event-driven WSN, nodes report data to the sink only when the defined event occurs and is detected by nodes. There are some characteristics of event-driven WSN, and one of which is that the amount of the nodes that detected the event (it can be called source node) is usually small. The source nodes are concentrated on in event area. Only some of the nodes are considered for constructing aggregation tree for event data in event-driven WSN. The aggregation tree construction techniques in this kind of application can be divided as follows: sub-optimal techniques, residual energy based techniques, center aggregation techniques, etc.

Some related studies prove that for any deployed WSN, data centric routing with least transmission times can be described as a problem--constructing a minimum Steiner tree, which is an NP-hard problem. Literature [6] mentioned three sub-optimal algorithms: GIT (Greedy Increment Tree) [7], SPT (Shortest Paths Tree), and CNS (Center at Nearest Source).

Min Ding, Xiuzhen Cheng and Guoliang Xue propose an aggregation tree construction algorithm based on residual energy EADAT [8] (Energy Aware Distributed Aggregation Tree). In EADAT, an aggregation tree is constructed and maintained. In order to save energy by reducing data transmission, the leaf nodes of the network turn off the radio module. When electing the aggregation nodes, the delay is calculated according to its residual energy. The nodes with more residual energy will delay less time to forward tree construction packet. So the more residual energy of the node, the bigger probability it becomes a non-leaf node. The non-leaf nodes will not turn off its radio because it may forward data packet comes from other nodes.

Guobin Liu, Yugeng Sun, and Ting Yang [9] propose center aggregation algorithm. The main idea of the algorithm is that the aggregation node is found in the center of the event area. The aggregation node fuses all received data. The aggregation node that lies in the center of the event area makes data transmission delay shortest. Time is very important for the event information in event-driven WSN. The center aggregation algorithms can be divided into the following four phases: gradient set up, center node finding, aggregation tree construction, and data transmission.

For the case that all nodes are far from sink, LEACH [10], PEGASIS [11] and hierarchical PEGASIS [12] are designed. Hierarchical PEGASIS makes some improvements over PEGASIS, and PEGASIS makes some improvements over LEACH.

C. Aggregation Tree Construction for Periodical Data in Large Scale WSN

Periodical data not only exist in time-driven WSN, but also in event-driven WSN. Time-driven WSN is another kind of WSN. In time-driven WSN, the node reports data periodically. In event-driven WSN, some information such as self-health information will be reported periodically. The common feature of all the periodical data is that the number of the nodes that needs to transmit data is very large. Nearly all nodes in the network transmit data at the same duration. For these applications, the aggregation tree with cluster-tree structure is commonly adopted, such as SCT [13] (Semantic/Spatial Correlation-aware Tree) and GROUP [3, 4] (Grid-clustering Routing Protocol), etc.

In SCT, the network is divided into many ring-sector sub-areas, as shown in Fig.1. For each sub-area, an aggregation node is selected. All the other nodes take the corresponding aggregation node which lies in the same sub-area as its parent (aggregation node). They can communicate with the corresponding aggregation node through single hop. All the aggregation nodes form the backbone of the network. The ideal aggregation node lies in the geometric center of the lower arc bounding of the sub-area. But the center of the lower arc bounding may not existing node, so the actual aggregation node is near the ideal location.

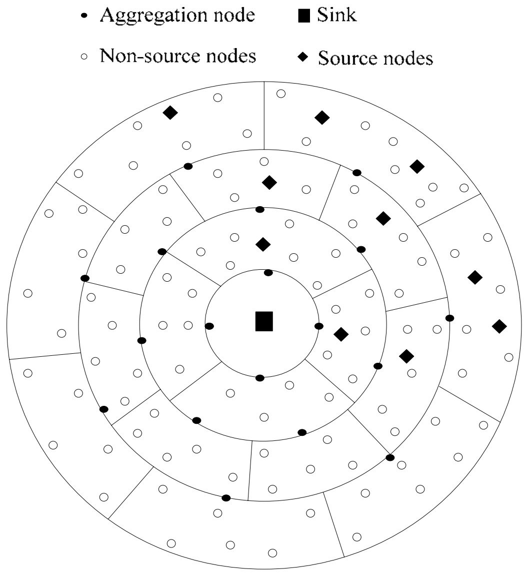

Figure 1. network structure of SCT[13]

To balance energy consumption, two methods are proposed in SCT. One is that changing the orientation of the sectors which can be implemented by changing the angle of polar coordinates. Another is varying the locations of the rings which can be implemented by changing the size of the most inner sub-area. The sizes of rings will not be changed except the first and last.

GROUP is designed based on the following network model:

- All sensor nodes in the network are homogeneous, stationary, and aware of their own locations.
- The number of sinks may vary over time. And all sink nodes can communicate with each other through tethered network or satellite.
- Each sensor node is able to adjust its wireless transceiver's power consumption.

The main idea of GROUP is that the cluster heads which are distributed approximately as grid are elected periodically or according to requirement. The election of the cluster head is initiated by one of the sink nodes. The sink who initiates the election of cluster head is primary sink. The primary sink can be selected based on location. The primary sink is closer to the center of network than other sinks. The cluster head election message is propagated in the network gradually until all the cluster heads have been elected. The cluster head is the node who is nearest to the cross of the grid, which is formed by taking a grid seed node as one of the crosses as shown in Fig.2. The grid seed is elected by primary sink.

The whole process of aggregation tree construction can be divided into following steps: primary sink node selection, grid seed election, cluster head (aggregation node) election, and cluster member participation.

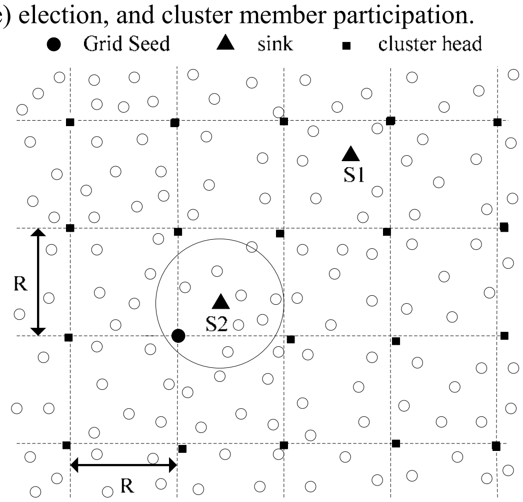

Figure 2. network structure of GROUP [3,4]

The main purpose of selecting a primary sink node is that selecting a sink node to initiate the tree construction and make sure the aggregation tree is sole.

After primary sink has been elected, the primary sink broadcasts grid seed election message and start a timer. The message contains a value between 0 and 1. If the nodes within the competition area have received the message, a random value between 0 and 1 will be generated. If the value generated is smaller than the received one, the node will send answer message to the primary sink. When the timer is expired, the primary sink will select a node which has more residual energy as grid seed. If no node answers the election message, the primary sink will resend the election message with a bigger value.

The grid is set up by taking the grid seed as one of the crosses. The other cluster heads are elected at the cross of the grid. But there may be no node at the cross, the node near to the cross will compete for the cluster head. The maximum distance between competition node and the cross of the grid is set to 0.4R in literature [4], where R is the cell size. The node nearest to the cross will have larger probability becoming cluster head. After the cluster head has been elected, other nodes will join the cluster.

In GROUP, cluster heads are distributed approximately as grid. The cluster is not divided by grid. To balance energy consumption, the aggregation tree must be reconstructed periodically or according to requirement. When reconstructing, if the grid seed elected is the same as the one in last turn, the reconstruction will be stopped. The data will be transmitted and fused by using the aggregation tree which is constructed in last turn.

The aggregation tree of aware data in query based WSN and in event-driven WSN can't be used in large-scale WSN for periodical data aggregation. The reconstruction of aggregation tree for SCT and GROUP to balance energy consumption will cause large amount of energy consumption. Furthermore, when electing aggregation node, only the distance to the ideal location is considered in both algorithms, without considering residual energy. There may be a problem that the node nearest to the ideal location may have little residual energy which makes the nodes died quickly.

For the problem mentioned above, an aggregation tree construction algorithm which is an improvement over GROUP is proposed in this paper.

III. Aggregation Tree Construction Based on Grid

The aggregation tree construction algorithm (ATCBG) proposed in this paper makes some improvements over GROUP which is designed based on the following assumptions:

- There is only one sink node in the network, and the sink is static.
- All the nodes are stationary and aware of their own locations which can be implemented by using positioning techniques or GPS.
- The nodes can adjust its transceiver's power according to the actual distance.

The main idea of ATCBG is that aggregation tree is constructed by taking the sink as the center of a grid. The size of the cell is R. The whole network is divided into grids. Each grid forms a cluster. The cluster head is elected by considering residual energy, distance to the center of the grid and other factors. The cluster head take responsible for data fusion. All the cluster heads form a tree-structure. The aggregation tree constructed in this paper is a cluster-tree structure.

A. *Aggregation Tree Construction*

The aggregation tree construction is initiated by sink. Sink first broadcasts tree construction message. The transmitting radius of broadcasting tree construction message is L_1. The structure of tree construction message is shown in Fig. 3.

Type	SID	DID	SX	SY	BSX	BSY	FCH	GRIDWidth

Figure 3. message structure of tree construction

Where, SID is node ID of source node (sending node), DID is destination address, here is broadcast address, SX and SY are position information of source node, BSX and BSY are the position information of sink, FCH is node ID of its parent (if the packet is sent by sink, this field will be filled with NULL), and GRIDWIDTH is cell size.

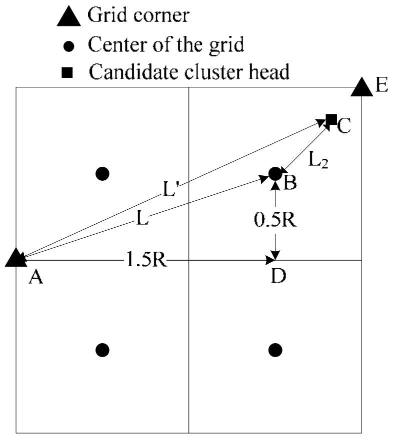

Figure 4. analysis of tree construction message broadcasting radius

To ensure that all the nodes in competition area can receive the tree construction message, the transmitting radius of sending tree construction message should satisfy formula (1), the analysis is shown in Fig. 4.

$$L_1 \geq L' \quad (1)$$

From the triangle theory that the sum of the length of any two edges of a triangle always exceeds the length of the third edge. It can be obtained that $L' \leq L + L_2$, where the equation will be established only when A, B, and C are in a line. Here, L_2 is the distance to the center of corresponding grid. The value of L can be calculated by Pythagorean Theorem as in formula (2).

$$\begin{aligned} L &= \sqrt{AD^2 + BD^2} \\ &= \sqrt{(1.5R)^2 + (0.5R)^2} = \sqrt{2.5}R \end{aligned} \quad (2)$$

The transmitting radius of the node is usually limited, so the size of the cell is also constrained. It is assumed that the maximal transmitting radius of the node is D, so there should be $L_1 \leq D$. The maximal value of L_1 is $\sqrt{5}R$ which is the distance between A and E in Fig.4. So the relation between cell size and maximal transmitting radius should satisfy formula (3)

$$D \geq \sqrt{5}R \quad (3)$$

Node N who receives tree construction message first calculates the grid coordinates (X_{Nr}, Y_{Nr}) according to the received packet information and itself position information (X_N, Y_N). The grid coordinates is calculated as formula (4).

$$\begin{cases} X_{Nr} = -\left\lfloor \dfrac{|X_N - X_S + \varepsilon - R/2|}{R} \right\rfloor & if\ X_N \leq X_S \\ X_{Nr} = \left\lfloor \dfrac{X_N - X_S - \varepsilon + R/2}{R} \right\rfloor & if\ X_N > X_S \\ Y_{Nr} = -\left\lfloor \dfrac{|Y_N - Y_S + \varepsilon - R/2|}{R} \right\rfloor & if\ Y_N \leq Y_S \\ Y_{Nr} = \left\lfloor \dfrac{Y_N - Y_S - \varepsilon + R/2}{R} \right\rfloor & if\ Y_N > Y_S \end{cases} \quad (4)$$

Where X_s and Y_s are the position information of sink, ε is an infinitesimal value used for processing the grid edge. The ε can be the minimum value that the node can process. $\lfloor X \rfloor$ is the down integral operation. It is assumed that the coordinates of grid which sink lies in is (0, 0). Fig.5 is an example of network area divided by grids.

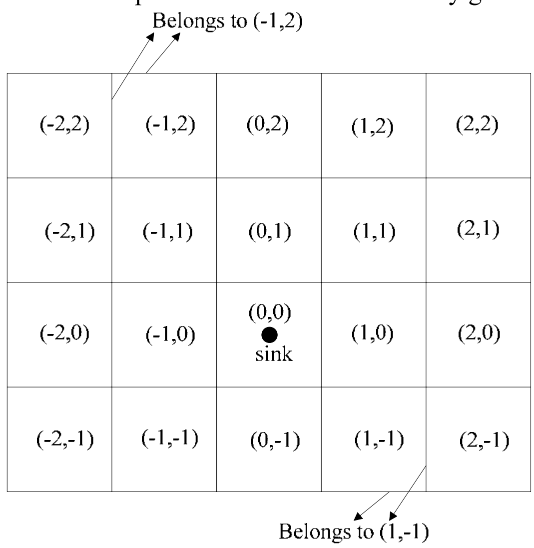

Figure 5. an example of network divided by grids

Definition 1: a neighbor grid of grid C is defined as upper grid, down grid, left grid, and right grid of grid C. Namely, if $abs(R_{DX} - R_{CX}) + abs(R_{DY} - R_{CY}) = 1$, and then grid D is the neighbor grid of C, where R_{CX} and R_{CY} are grid coordinates of grid C, R_{DX} and R_{DY} are grid coordinates of grid D.

The node A received tree construction message will take various processes according to its grid coordinates and the sending node B's grid coordinates information after calculated the corresponding grid coordinates. The process is introduced as follows:

1) If node A and node B come from the same grid, A will join the aggregation tree by taking B as the cluster head, and then record the information of the cluster head.

2) If node A and node B are from neighbor grids, A first check whether it has already joined the tree or not. If A has already joined the tree, A will check whether B takes A as its parent node. If B takes A as the parent node, A will take B as the child node. If A has not joined the tree, then it will compete for cluster head.

The algorithm that node A receives tree construction message from node B is described as follows:

```
Algorithm: ProcessTreeConMsg(msg)
if(A.grid==B.grid)            //A and B in the same grid
{
    A.clusterHead=B;          //A take B as cluster head
    A.isJoinTree=1;           //mark itself has join the tree
    A.recordCHInfo(B);        // record information of CH
}
if(abs(A.grid-B.grid)==1)     //A and B from neighbor grids
{
   if(!A.isJoinTree)          // A has not join Tree
      A.competeForCH();
   else                       //A has joined the tree
   {
      if( B.father==A)        //B take A as father
      {
         A.child=B;
         A.recordChildInfo(B);//record information of child
      }
   }
}
```

It is assumed that during competition for cluster head, the time difference of receiving the packet is negligible. The process of node A competing for the cluster head is presented as follows:

When node A receives the tree construction message, A first check whether it is competing for cluster head. If A is in competition, it will first cancel the competition and then restart competition by selecting a better candidate parent node according to the information of competition. If A is not in competition, A competes for the cluster head by taking the sender as its candidate parent.

Competition is implemented mainly by timer scheme. The competition nodes set delay of the timer according to the following information: residual energy of the node, distance between the node and candidate parent node, distance between the node and the furthest corner of the grid, and furthest distance between the node and the center of neighbor grids.

When the timer expires, node A will broadcast competition message. The competition message should be received by all the other candidate nodes in the same grid. So the transmitting radius is set to $\sqrt{2}R$. The message includes delay of the timer. The receiving node B uses the received competition message to check if the delay of its own is larger than A. If the delay is larger than A, B will cancel the competition, or B will broadcast the competition message. The node that broadcasts competition message last will become cluster head, and will broadcast tree construction message with transmitting radius L_1. The process will be iterated until all the nodes in the network join the tree.

The delay of the timer is composed by three parts. It can be calculated by formula (5).

$$T = T_1 + T_2 + T_3 \qquad (5)$$

Where, T_1 and T_2 make the head node closer to the center of the grid, T_3 makes the distance between the node and its parent shorter. T_1, T_2, and T_3 are calculated as formula (6), (7) and (8) respectively.

$$T_1 = \begin{cases} \dfrac{E_{elec} + \varepsilon_{fs} d_c^2}{E_{ini}} & if\ d_c < d_0 \\ \dfrac{E_{elec} + \varepsilon_{amp} d_c^4}{E_{ini}} & if\ d_c \ge d_0 \end{cases} \qquad (6)$$

$$T_2 = \begin{cases} \dfrac{E_{elec} + \varepsilon_{fs} d_{nc}^2}{E_{ini}} & if\ d_{nc} < d_0 \\ \dfrac{E_{elec} + \varepsilon_{amp} d_{nc}^4}{E_{ini}} & if\ d_{nc} \ge d_0 \end{cases} \qquad (7)$$

$$T_3 = \begin{cases} \dfrac{E_{elec} + \varepsilon_{fs} d_f^2}{E_{res}} & if\ d_f < d_0 \\ \dfrac{E_{elec} + \varepsilon_{amp} d_f^4}{E_{res}} & if\ d_f \ge d_0 \end{cases} \qquad (8)$$

Where, d_c is the distance to the furthest corner of the grid, d_f is the distance to the parent node, d_{nc} is the furthest distance to the center of neighbor grid, E_{res} is the residual energy of the node, E_{ini} is the initial energy of the node. In energy consumption model, when the transmitting distance is shorter than d_0 , it adopts free space model. Otherwise, it adopts Multi-channel attenuation model. E_{elec} is energy consumed by transmit circuit for transmitting or receiving one bit data. ε_{fs} and ε_{amp} represent the energy for amplifier in free space model and Multi-channel attenuation model respectively.

After all aggregation nodes have been elected, the cluster head is distributed as shown in Fig.6.

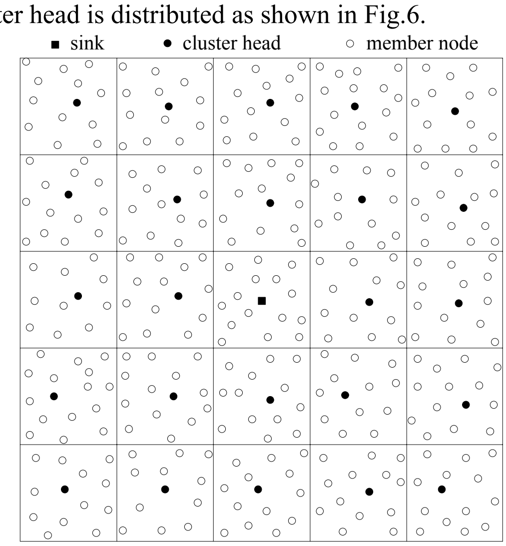

Figure 6. An example of cluster head distribution

B. Cluster Head Replacing Scheme

The cluster head will consume more energy due to receiving and fusing all the data from its member nodes and child nodes. To avoid premature death of the cluster head, the cluster head must be replaced after a certain time. The cluster head is replaced by reconstructing which is adopted by SCT and GROUP. In SCT and

GROUP, there is a problem in replacing cluster head by reconstructing the aggregation tree, that is, it consumes much energy. ATCBG does not adopt that way. ATCBG replaces the cluster head when its residual energy is below half of the energy with which it was electing for cluster head. The energy introduced here is the work energy, namely the node can't work only when the residual energy is 0. When the residual energy below the threshold, the cluster head sends replacing cluster head message for replacing the cluster head. The processing is introduced as follows:

After sending data, the head node checks if the residual energy is below $E_{cluster}/2$ or not, where $E_{cluster}$ is the energy when the node becomes cluster head. If the energy is below the threshold, it sends replacing cluster head message. The message includes message type, source node ID and the grid coordinates information in which the node lies. The message structure is shown in Fig.7.

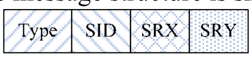

Figure 7. message structure of replacing cluster head

The replacing cluster head message should be received by all the nodes in the same grid, including the parent and children. So the transmitting radius L_3 should satisfy formula (9).

$$L_3 = MAX(d_{UC}, d_{DC}, d_{Corner}) \tag{9}$$

Where d_{UC} and d_{DC} are the furthest distance to the parent node and to child nodes respectively, d_{Corner} is the distance to the furthest corner of the grid.

After receiving the replacing cluster head message from node A, the received node B processes as follows:

1) If node A and node B are from the same grid, B first changes into the state that have not join the tree, and then compete for new cluster head.

2) If node A and node B are from neighbor grids, the process is shown as follows:

- If B takes A as parent node, B will delete the routing information of A.

- If B takes A as the child node, B will delete the routing information of node A.

C. Data Transmission

The cluster member nodes first send the collected data to the corresponding cluster head. The cluster head fuses the data after receiving all the data from member nodes and its child nodes. And then cluster head sends the fused data to its parent. The process continues until the data is sent to the sink. When transmitting data, the nodes adjust its own transmission power according to the actual distance.

D. Failure Recovery

If the node had not received data from its child node for a certain time period, it assumes that the child node is failed, and then it broadcasts the information to the corresponding grid for competing a new cluster head.

IV. SIMULATION

A. Network Model

OMNet++ is adopted for the simulation. Energy consumption model adopts the commonly used one, such as the energy consumption model in literature [14] and [15]. The transmitting energy consumption model is shown in formula (10). The receiving energy consumption model is shown in formula (11). The data fusion energy consumption is shown in formula (12). Some experiment parameters are set as shown in table I.

$$\begin{cases} E_{trans} = lE_{elec} + l\varepsilon_{fs} d^2 & if\ d < d_0 \\ E_{trans} = lE_{elec} + l\varepsilon_{amp} d^4 & if\ d \ge d_0 \end{cases} \tag{10}$$

$$E_{rece} = lE_{elec} \tag{11}$$

$$E_{fusion} = signal * \varepsilon_{fusion} \tag{12}$$

Where E_{elec}, ε_{fs}, and ε_{amp} have been introduced in the paper, l is the length of packet, signal is the whole data length for data fusion, and ε_{fusion} is the energy consumed by fusing one bit data.

TABLE I.
SOME PARAMETERS IN ENERGY CONSUMPTION MODEL

Parameter	Value
Threshold distance(d_0)(m)	87
E_{elec}(nJ/bit)	50
ε_{fs} (pJ/bit/m^2)	10
ε_{amp} (pJ/bit/m^4)	0.0013
ε_{fusion} (nJ/bit)	5
packetSize (byte)	100
Initial energy (J)	0.5

The ATCBG is implemented in OMNet++. One sink in the center of the field, and 300 nodes are randomly distributed in an area of 1200m×1000m which is the same as literature [3] and [4]. It is shown in Fig.8.

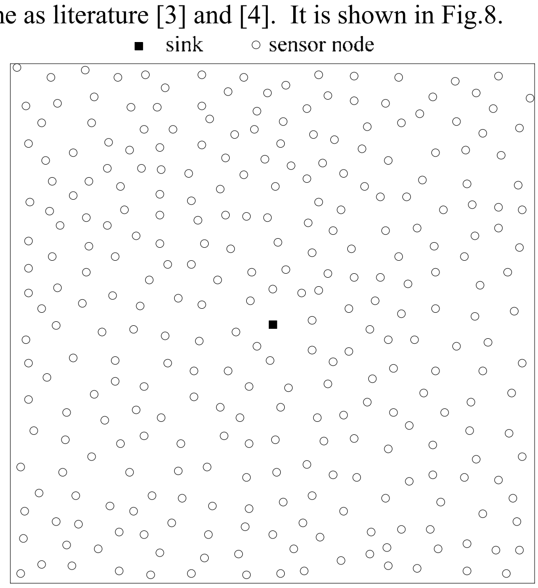

Figure 8. node distribution

The length of all the packets is defined as 100 bytes. It is assumed that data is perfectly aggregated [9] in cluster head. This means that no matter how many packets the cluster head has received, after data fusion, only one packet is sent to its parent.

B. *Impact of Parameter on ATCBG*

In ATCBG, the only parameter is the cell size R. To evaluate the impact of cell size on ATCBG, a lot of experiments have been done. During the experiment, the network field and the nodes number remain the same. The cell size changed from 100 meters to 220 meters with step of 20 meters. The impact of cell size is evaluated by considering the average energy consumption of sending 100 data packets. Table II shows the average energy consumption of sending 100 data packets.

TABLE II.
AVERAGE ENERGY CONSUMPTION OF SENDING 100 DATA PACKETS

R(M)	100	120	140	160	180	200	220
E(uJ)	14634	18375	20305	26702	30943	40336	45217

From the table, it can be seen that with the increment of cell size, the average energy consumption is increasing. Considering from the aspect of energy consumption and lifetime of the network, the cell size will be selected to make the average energy consumption lowest and make the lifetime of the network longest. But smaller size of the cell will increase the hops of the network and so increase the data transmission delay. In WSN, real-time data usually is a basic requirement. The outdated data will take no use to the user. The cell size should be chosen by balancing the lifetime and delay according to actual application. It can be seen from table II that with the increment of R, average energy consumption is increasing. The energy is proportional to d^4 when d is larger than d_0. When the cell size increased from 180 meters to 200 meters, the average energy consumption increased more quickly. So, 180 meters is adopted to compare the performance with GROUP in this paper.

To evaluate the impact of node density on ATCBG, experiment with different node density has been done. The node number is changed from 250 to 450 with step of 50. Fig.9 shows average energy consumption with various node densities.

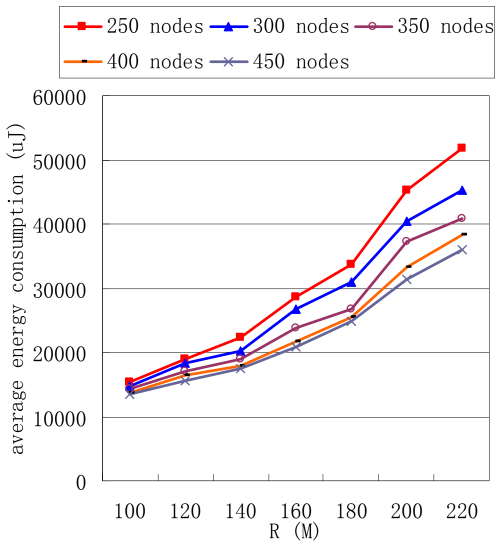

Figure 9. average energy consumption with various node densities

It can be seen form Fig.9 that the impact of node density on ATCBG is very small. The average energy consumption will be little lower. This occurs because with the density increasing, it will be more probable that the node lies in the center of the grid.

C. *Comparison With Group*

Two metrics are chosen to analyze the performance of ATCBG compared with GROUP:

- Average energy consumption of sending a certain amount of data packets.
- Total packets sent by one node before emergence of first node death.

In GROUP, the cell size is set to 200 meters by considering the energy consumption and delay. The comparison is done for the cell size being 180 meters and 200 meters respectively. In GROUP, the corresponding parameters are set as in literature [4]. The cluster head's maximum distance to the cross is 0.4R, and the member node's maximum distance to the cluster head is 0.75R. But in simulation experiment, there are some nodes can't join the tree when the member's maximum distance to the cluster head is 0.75R. In order to make sure that all the nodes can join the tree, 1.1R is set for the member's maximum distance to the cluster head. The analysis is shown as in Fig. 10.

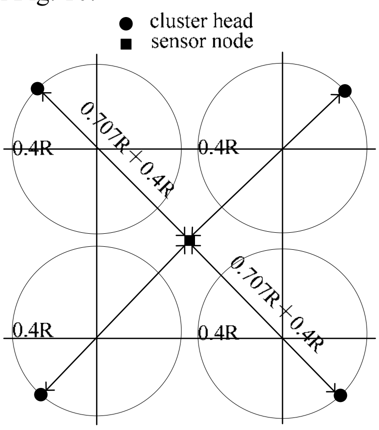

Figure 10. member's maximum distance to cluster head

1) *Comparison of average energy consumption*

Fig.11 and Fig.12 are the comparison of average energy consumption of sending 40 packets in various node densities when the cell size is set to 180 meters and 200 meters respectively. In the experiment, the tree reconstruction interval in GROUP is set to 10 packets, 20 packets, 30 packets, and 40 packets. In the Fig.11 and Fig.12, RI-X means that the reconstruction interval is X packets, the X corresponding to 10, 20, 30 or 40 packets.

From the figure, it can be seen that with the increment of node density, the average energy consumption of sending 40 packets in ATCBG is decreased slowly. This is because it is more probable that the aggregation node is closer to the center of the grid with higher node density.

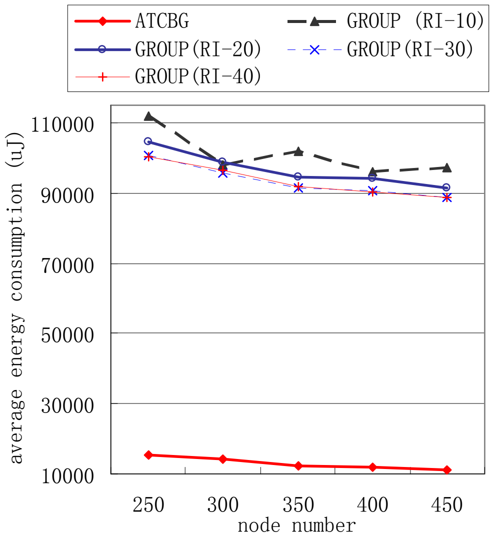

Figure 11. average energy consumption using ATCBG vs. GROUP(with different reconstruction intervals) (R=180m)

In GROUP, the average energy consumption of sending 40 packets is decreased, when reconstruction interval is set to 20 packets, 30 packets, and 40 packets, no matter the cell size is 180 meters or 200 meters. It can be explained as follows: first, it is more probable to find grid seed in shorter time for tree reconstruction with higher node density. Second, the number of cluster heads remains the same, as long as network area and cell size are fixed. All the increased nodes can be treated as cluster member who consumes less energy. Finally, the node number in competition area is larger with higher node density. But, the average energy consumption is fluctuant when the reconstruction interval is set to 10 packets. The fluctuation is more serious when the cell size is 180 meters as shown in Fig.11. This occurs due to the tree reconstruction mechanism in GROUP. The grid seed selected for reconstructing will have more probability the same as the one in last turn. In this case, the reconstruction will be stopped. So lesser energy will be consumed

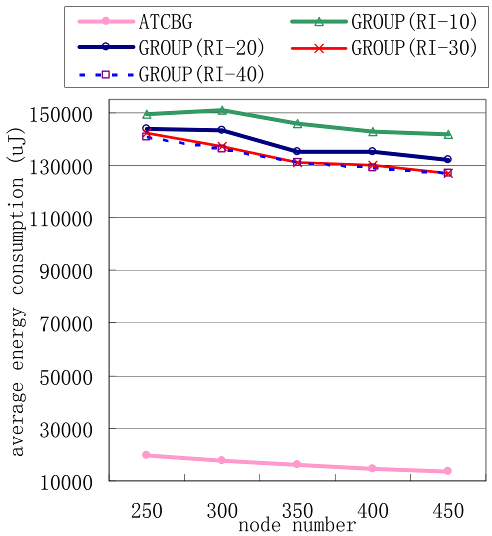

Figure 12. average energy consumption using ATCBG vs. GROUP(with different reconstruction intervals) (R=200m)

2) Comparison of packets sent before emergence of first node death

The number of packets has been compared in the experiment, which have been sent from one node before the emergence of first node death. But the fist node dead not only related to the algorithm, but also related to the nodes distribution in the cluster.

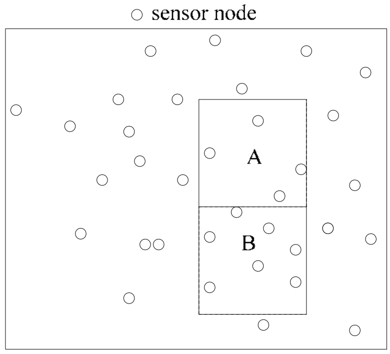

Figure 13. analysis of fist dead node related to node distribution

Nodes in cluster A will die earlier than the nodes in cluster B as shown in Fig.13. This is because more candidate nodes can be selected as cluster head when replacing the cluster head in cluster B. All nodes are randomly distributed in the particular network area. The time of first node died will varies with various node densities which will cause different node distribution.

The maximum and minimum packets have been sent from one node before emergence of first node death for various node densities is compared in the experiment.

Table III and table IV are the comparison of maximum and minimum packets number sent in the network for various node densities before the emergence of first node death. In the experiment, the reconstruction interval in GROUP is set to 10 packets, 20 packets, 30 packets, and 40 packets respectively.

TABLE III.
COMPARISON OF PACKETS NUMBER SENT BY ONE NODE BEFORE FIST NODE DEATH (R=180 M)

	ATCBG	GROUP (different reconstruction interval)			
		10pkts	20pkts	30pkts	40pkts
MAX	590	98	108	98	103
MIN	199	81	93	72	72

TABLE IV.
COMPARISON OF PACKETS NUMBER SENT BY ONE NODE BEFORE FIST NODE DEATH (R=200 M)

	ATCBG	GROUP (different reconstruction interval)			
		10pkts	20pkts	30pkts	40pkts
MAX	389	69	67	73	70
MIN	193	46	47	46	46

It can be seen from the two tables that the minimum packets number sent before first node death in ATCBG is larger than the maximum of GROUP in all occasions.

V. Conclusion And Future Work

ATCBG proposed in this paper is an improvement over GROUP. ATCBG outperforms GROUP from the following aspects: First, when selecting cluster head, only the distance to the cross of the grid is considered in

GROUP, without considering the residual energy. So, the node closer to the cross and with little residual energy may become cluster head, and it leads to the head dying soon. ATCBG proposed in this paper not only consider the distance to the center of the grid, but also consider the residual energy. Second, aggregation tree reconstruct periodically or according to the requirement is adopted to avoid cluster head premature death. But reconstruct the aggregation tree will consume large energy. To avoid the cluster head premature death, the cluster head whose residual energy below half of energy when it become cluster head will be replaced in ATCBG. Finally, 1.5R is taken as the transmitting radius of cluster head no matter how close to the parent, and the transmitting radius for member nodes is little shorter. ATCBG will adjust its transmitting radius according to the actual distance to save more energy.

In future work, the periodically data for self health information, such as residual energy or the link quality, will be fused based on the aggregation tree constructed in this paper.

REFERENCES

[1] Min Xiang, Weiren Shi, Changjiang Jiang and Zhiyong Luo, "Energy Saving Algorithm Based on Cluster Head Prediction for Wireless Sensor Networks," Computer Engineering, vol.34, pp. 27~29,2008.

[2] Fangmin Li, Wenjun Xu, Xinhua Liu, "Power control for wireless sensor networks," Journal of Software, vol.19, pp. 716~732, 2008.

[3] Liyang Yu, Neng Wang, Wei Zhang, Chunlei Zheng, "GROUP: a Grid-clustering Routing Protocol for Wireless Sensor Networks," In proceedings of 2006 IEEE International Conference on Wireless Communications, Networkding and Mobile Computing (WiCOM 2006), Wuhan, China, 2006, pp.1~5.

[4] Liyang Yu, "Study on grid-clustering routing protocol and data aggregation for wireless sensor networks," East China Normal University, 2008.

[5] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," In Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'2000), Boston, Massachusetts, August 2000.

[6] Krishnamachari B，Estrin D，Wicker S, "Modeling data-centric routing in wireless sensor networks," In Proceedings of IEEE INFOCOM，2002.

[7] Shinji MIKAMI，Takafumi AONISHI，Hironori YOSHINO，Chikara OHTA，Hiroshi KAWAGUCHI，Masahiko YOSHIMOTO, "Aggregation Efficiency-Aware Greedy Incremental Tree Routing for Wireless Sensor Networks," IEICE Transactions on Communications, 2006，E89-B (10), pp.2741~2751.

[8] Ding M, Cheng X, Xue G, "Aggregation Tree Construction in Sensor Networks," In proceeding of the IEEE 58th Vehicular Technology Conference. 2003, vol.4, pp.2168 ~2172.

[9] Guobin Liu, Yugeng Sun, Ting Yang, "Research on Center Aggregation Algorithm in Wireless Sensor Networks," Chinese Journal of Sensors and Actuators, 2006, vol.19, pp.872~877.

[10] Heinzelman W，Chandrakasan A，Balakrishnan H, "Energy-efficient Communication Protocol for Wireless Microsensor NetWorks," In Proceeding of 33rd Annual Hawaii Int'l Conf. on System Sciences. Maui：IEEE Computer Society，2000, pp.3005~3014.

[11] Lindsey S，Raghavendra C, "PEGASIS：Power-efficient Gathering in Sensor Information System," In Proceeding of the IEEE Aerospace Conference. Montana：IEEE Aerospace and Electronic Systems Society，2002, pp.1125~1130.

[12] Lindsey S，Raghavendra C，Sivalingam K, "Data gathering in sensor networks using the energy * delay metric," Proceeding of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing'01，2001, pp.23~27.

[13] Yujie Zhu, Ramanuja Vedantham, Seung-Jong Park and Raghupathy Sivakumar, "A Scalable Correlation Aware Aggregation Strategy for Wireless Sensor Networks," Proceedings of the first International conference on wireless internet, 2005, pp.122~129.

[14] Yingchi Mao, Guofu Feng, Lijun Chen, Daoxu Chen, "A location-independent connected coverage protocol for wireless sensor networks," Journal of software, vol 18, No.7,July 2007, pp.1672~1684

[15] Heinzelman WB, Chandrakasan AP and Balakrishnan H, "An Application Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Transactions on Wireless Commu- nications, 2002.vol.1, pp. 660~670.

Jian Shu was born on May 25,1964, in Nanchang, Jiangxi Province, China. He received a Ms. in Computer Networks from Northwestern Polythecnical University in 1990.

He is currently a professor at the school of software, Nanchang Hangkong University, China. His research interests include wireless sensor network, embeded system and software enginering.

Yebin Chen was born on March 30,1985, in Nanchang, Jiangxi Province, China. He is a postgraduate student of Nanchang Hangkong University. His research interest is wireless sensor networks.

Linlan Liu was born on March 22,1968, in Nanchang, Jiangxi Province, China. She received a Bs. in Computer Aplication from National University of Defence Technology in 1988.

She is currently a professor at the school of information engineering, Nanchang Hangkong University, China. Her research interests include Software Engineering and Distributed System.

Sheng Zhang was born on December 21, 1968, in Huanggang, Hubei Province, China. He received the Ph.D. from Institute of Geodesy and Geophysics, Chinese Academy of Sciences, China, in 2006.

He is currently an associate professor at the school of information engineering, Nanchang Hangkong University, China. His research interests include data mining, artificial intelligence, Web information processing, GIS, and wireless sensor networks.

Jun Li was born on November 19,1985, in Ji'an, Jiangxi Province, China. He is a postgraduate student of Nanchang Hangkong University. His research interest is wireless sensor networks.