A Low Energy Intelligent Clustering Protocol for Wireless Sensor Networks

Qiao Li[1], Lingguo Cui[1], Baihai Zhang[1] and Zhun Fan[2]
[1]School of Automation, Beijing Institute of Technology, Beijing-100081, China
liqq007@gmail.com; cuilingguo@bit.edu.cn; smczhang@bit.edu.cn
[2]Department of Mechanical Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark
zhfa@man.dtu.dk

***Abstract*-LEACH (low-energy adaptive clustering hierarchy) is a well-known self-organizing, adaptive clustering protocol of wireless sensor networks. However it has some shortcomings when it faces such problems as the cluster construction and energy management. In this paper, LEICP (low energy intelligent clustering protocol), an improvement of the LEACH protocol is proposed to overcome the shortcomings of LEACH. LEICP aims at balancing the energy consumption in every cluster and prolonging the network lifetime. A fitness function is defined to balance the energy consumption in every cluster according to the residual energy and positions of nodes. In every round the node called auxiliary cluster-head calculates the position of the cluster-head using Bacterial Foraging Optimization Algorithm (BFOA). After aggregating the data received, the cluster-head node decides whether to choose another cluster-head as the next hop for delivering the messages or to send the data to the base station directly, using Dijkstra algorithm to compute an optimal path. The performance of LEICP is compared with that of LEACH. Simulation results demonstrate that LEICP can prolong the lifetime of the sensor network by about 62.28% compared with LEACH and acquire uniform number of cluster-heads and messages in the network.**

I. INTRODUCTION

The wireless sensor network (WSN) is a class of wireless networks in which sensor nodes collect, process and transmit data acquired from the physical environment to an external base station [1]. Sensor nodes are equipped with the regular power. Energy is consumed during computation and communication among the nodes. Although there are many important aspects which need to be taken into consideration when we are dealing with the overall network design problem, energy efficiency should be considered as the key design objective among them, since a sensor node can only be equipped with a limited energy supply in all application scenarios. Sensor node lifetime shows a very strong dependency on battery lifetime. That is the reason why we have a focus on the issue of energy efficiency when considering the routing protocols of the WSNs.

LEACH (low-energy adaptive clustering hierarchy) is a well-known self-organizing, adaptive clustering protocol [2], [3]. In LEACH, all the nodes organize themselves into local clusters according to a certain procedure, with a number of nodes acting as the cluster-heads and other nodes acting as the members of the clusters. Because the cluster-head consumes more energy than the member node, LEACH includes randomized rotation of the cluster-head positions in order not to drain the energy of a particular sensor. In a cluster, the cluster-head performs local data fusion to "compress" the amount of data being collected from the members of the cluster and transmits the processed data to the base station in order to reduce energy consumption and enhance the lifetime of the whole sensor network. LEACH runs many rounds in the lifetime of the network, and each round contains a cluster formation phase and a cluster steady phase. In the cluster formation phase, a number of nodes are selected to act as the local cluster-heads with a certain probability. Then these cluster-heads broadcast their status to other nodes near them in the network. Each sensor node chooses the nearest one as its own cluster-head. Once all the nodes are organized into clusters, each cluster-head creates a TDMA schedule for all the members in its cluster. In the cluster steady phase, the member node collects the data messages from the physical environment and sends them to its cluster-head, which aggregates the data and then transmits them to the base station directly. In the next round, the network performs the cluster-head selection procedure again. Time synchronization is a key factor in the network and the cluster-head must be awake to receive all the data from the members in its cluster. Once the cluster-head receives all the data, it performs data aggregation to enhance the useful signal and reduce the uncorrelated noises. To reduce energy dissipation, each member node uses power control to set the amount of transmission power based on the received strength of the cluster-head advertisement. Furthermore, the radio of each member node is turned off until its allocated transmission time slot. The following equation gives the definition of probability with which a cluster-head is selected:

$$T_n = \begin{cases} \dfrac{P}{1 - P \cdot (r \bmod \dfrac{1}{P})} & if \;\; n \in G \\ 0 & otherwise \end{cases} \tag{1}$$

where n is the current node, P is the desired percentage of cluster-heads among all nodes, r is the current round, and G is

the set of nodes that have not been cluster-heads in the last $1/P$ rounds. In the cluster formation phase, each node in the network decides by itself whether or not to become a cluster-head for the current round with a certain probability. This decision is made by the node choosing a random number between 0 and 1. If the number is less than the threshold T_n, the node becomes a cluster-head for the current round.

However LEACH has some shortcomings when it faces such problems as cluster construction and energy management. LEACH doesn't fully consider the distribution situation of nodes when it chooses the cluster-heads. The number of nodes in every cluster is not uniform. The nodes far from the cluster-heads will consume much more energy when they are communicating with their cluster-heads. Some cluster-heads, which will communicate with the base station directly, are distributed in the network unevenly. They will drain their energy quickly if they are far from the base station or own large number of members. During the cluster-head election procedure, residual energy and positions of nodes are not fully taken into consideration.

Many improvements of LEACH have been studied in recent years [4], [5], [6]. The improvements can be arranged into two categories. One category focuses on changing the cluster-head selection procedure, and the other aims at avoiding direct communication between the cluster-head and the base station, and using multi-hop among the cluster-heads is a general choice for many studies in this line. In addition, most improvements consider the residual energy of the nodes. How to balance the energy consumption in the whole network is an attractive problem for most researchers.

The rest of the paper is organized as follows. The Bacterial Foraging Optimization Algorithm (BFOA) is first introduced in Section Ⅱ. The network model and radio model are given in Section Ⅲ. Our proposed protocol, low energy intelligent clustering protocol (LEICP), is introduced in details in Section Ⅳ. Our proposed protocol is compared with LEACH by simulations in Section Ⅴ. The paper concludes in Section Ⅵ.

II. BACTERIAL FORAGING OPTIMIZATION ALGORITHM(BFOA)

Some computational methodologies have been used to improve the present network routing protocols [5], [6]. Up to now many biologically inspired computational methodologies have received much attention. Bacterial Foraging Optimization Algorithm (BFOA) [7], [8] is a well-known computational methodology which is based on the study of the bacterial foraging behaviors. The complex but organized activities exhibited in bacterial foraging patterns could inspire a new solution for optimization problems. The underlying mechanism of the surviving of bacteria, especially *E. coli* in a complex environment has been reported by researchers in the area of biological sciences. Inspired from these phenomena, BFOA was developed as an optimization algorithm by K. M. Passino [7], [8], in which the self-adaptability of individuals in the group searching activities has attracted a great deal of interests.

E. coli is a common type of bacteria. An *E. coli* bacterium alternates between running and tumbling. If it swims up nutrient gradient the *E. coli* will swim longer. If it swims down nutrient gradient the *E. coli* will search again to avoid unfavorable environments. Events can occur such that all the bacteria in a region are killed or a group is dispersed into a new part of the environment. Elimination and dispersal events have the effect of possibly destroying chemotactic progress, but they also have the effect of assisting to place bacteria near good food sources. When the bacteria are moving, they can release the attractant aspartate to congregate into groups and move as concentric patterns of groups with high bacterial density.

If the basic goal is to find the minimum of

$$J(\theta),\ \theta \in R^P$$

θ is the position of a bacterium, and $J(\theta)$ represents an attractant-repellant profile ($J < 0$, $J = 0$, and $J > 0$ represent the presence of nutrients, a neutral medium, and the presence of noxious substances, respectively).

$$P(j, k, l) = \{\theta^i(j, k, l) / i = 1, 2, \ldots S\}$$

represents the positions of each member in the population of the S bacteria at the jth chemotactic step, kth reproduction step, and lth elimination-dispersal event. $J(i, j, l)$ denotes the cost at the location of the ith bacterium $\theta^i(j, k, l) \in R^P$.

N_c is the length of the lifetime of the bacteria as measured by the number of chemotactic steps. The tumble step can be represented as follows:

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i)\phi(j) \tag{2}$$

$\phi(j)$ is generated as a unit length random direction. $C(i) > 0$ is the size of the step taken in the random direction specified by the tumble. Another chemotactic step of size $C(i)$ in this same direction will be taken if the cost $J(i, j+1, k, l)$ at $\theta^i(j+1, k, l)$ is better than at $\theta^i(j, k, l)$. N_s is the maximum number of chemotactic steps.

The function $J_{cc}^i(\theta)$ is used to model the cell-to-cell swarming step.

$$J_{cc}(\theta) = \sum_{i=1}^{S} J_{cc}^i \tag{3}$$

$$= \sum_{i=1}^{S}[-d_{attract}\exp(-w_{attract}\sum_{j=1}^{P}(\theta_j - \theta_j^i)^2)] + \sum_{i=1}^{S}[-h_{repellant}\exp(-w_{repellant}\sum_{j=1}^{P}(\theta_j - \theta_j^i)^2)] \tag{4}$$

where $d_{attract}$ is the depth of the attractant released by the cell. $w_{attract}$ is a measure of the width of the attractant signal. $h_{repellant} = d_{attract}$, which is the height of the repellant effect. $w_{repellant}$ is a measure of the width of the repellant. $\theta = [\theta_1, \ldots \theta_p]^T$ is a point on the optimization domain, which can have P dimension.

The goal is to find the minimization of

$$J(i, j, k, l) + J_{cc}(\theta^i(j, k, l)).$$

The bacteria will try to find nutrients, avoid noxious substances, and at the same time try to move toward other bacteria, but not too close to them. The $J_{cc}(\theta^i(j, k, l))$ function

dynamically deforms the search landscape to represent the desire to swarm. After N_c chemotactic steps, a reproduction step is taken. N_{re} is the number of reproduction steps. In the reproduction steps healthiest bacteria split, the same number of unhealthy ones are killed. N_{ed} is the number of elimination-dispersal steps with probability p_{ed}.

Up to now, BFOA has successfully been applied to solve various real word problems like optimal controller design [9], transmission loss reduction [10] and so on. Some key steps of BFOA have been further studied recently. A. Abraham et al. analyze the reproduction operator of BFOA [11]. S. Dasgupta et al. analyze adaptive computational chemotaxis in BFOA [12]. B. Niu et al. study a lifecycle model of BFOA [13].

III. The Network Model and the Radio Model

A. The Network Model

The network model we adopted in our research has the following features:

All the nodes are distributed randomly in a rectangle field and all the nodes are immobile.

All the nodes have the same transmitting range and the same energy level at the initial stage. The transmitting range of each node can cover the detection field.

After the node drains its energy, it dies, and it can't act as a cluster-head or a cluster member any more.

Every node knows its own position and residual energy. It knows these two attributes of its neighbors through exchanging status information.

The base station is immobile, and it has enough energy.

B. The Radio Model

The radio model utilized in LEICP is similar to that in LEACH [2], [3]. We assume an energy loss due to channel transmission. The radios can perform power control and hence use the minimum energy required to reach the intended recipients. Due to attenuation with distance, one energy loss model is used for relatively short distances and the other energy loss model is used for longer distances. The energy consumed by the radio in transmitting l bits data over a distance d is given as follows:

$$\begin{aligned} E_{TX}(l,d) &= l \cdot E_{elec} + l \cdot \varepsilon_{FS} d^2, \quad if \quad d < d_0 \\ &= l \cdot E_{elec} + l \cdot \varepsilon_{TR} d^4, \quad if \quad d \geq d_0 \end{aligned} \tag{5}$$

where E_{elec} is the energy dissipated per bit to run the transmitter or the receiver circuit. ε_{FS} and ε_{TR} depend on the transmitter amplifier model we use, and $d_0 = \sqrt{\varepsilon_{FS} / \varepsilon_{TR}}$ is the threshold transmission distance. To receive l bits data, the radio consumes

$$E_{RX}(l) = l \cdot E_{elec} \tag{6}$$

In our study, the parameters are set as: E_{elec} = 50nJ/bit, ε_{FS} = 10pJ/bit/m^2 and ε_{TR} =0.0013pJ/bit/m^4. The data fusion model used in our research assumes that the cluster-head would process $(n/2)\cdot l$ bits of data at the end of one frame (n is the number of cluster members). The energy cost for data aggregation is set as E_{DA} =5nJ/bit.

IV. LEICP: The Low Energy Intelligent Clustering Protocol

LEICP aims at balancing the energy consumption among the nodes in every cluster and reducing the energy dissipation of the cluster-heads. Like LEACH, LEICP runs many rounds in the lifetime of the network. The optimum number of clusters can be ascertained according to the method introduced in [3]. There is a cluster-head in every cluster. LEICP includes three phases during the working process: a cluster formation phase, a cluster-head adjustment phase and a steady phase.

In the cluster formation phase, every node sends its position information to the base station. A certain number of nodes are elected to act as the auxiliary cluster-heads with a certain probability. According to the number of auxiliary cluster-heads, the network is divided into the same number of clusters evenly. If no node dies, the number of auxiliary cluster-heads is fixed, so is the number of members in every cluster. The base station uses greedy algorithm to construct the cluster network. However, it is difficult to ensure that all nodes have close neighbors as this problem is similar to the traveling salesman problem. To construct the cluster network, the base station starts with the furthest node from it as the first auxiliary cluster-head, which chooses the nearest fixed number of nodes as its member nodes. Then the base station chooses the furthest node from it as the second auxiliary cluster-head from the rest nodes, which chooses the nearest fixed number of nodes as its members. This process continues until all the auxiliary cluster-heads and members are chosen out. The base station sends notifications to all the nodes in the network. According to this procedure, if no sensor node dies in the network, this cluster construction is stable. The network owns the same auxiliary cluster-heads and every auxiliary cluster-head owns the same members. But the auxiliary cluster-heads are not the final cluster-heads, and BFOA algorithm is used for the adjustment. The algorithm is employed by every node. Auxiliary cluster-heads then decide the final cluster-heads by BFOA after every member node sends its position and residual energy information to its auxiliary cluster-head. After the cluster formation phase, this computation process is performed in the second phase, the cluster-head adjustment phase.

In the cluster-head adjustment phase, every member node sends its position and residual energy information to its auxiliary cluster-head. Each auxiliary cluster-head computes the position of the final cluster-head, and sends notifications to the final cluster-head and other members in the cluster. In this phase, the BFOA algorithm is used by the auxiliary cluster-head for the adjustment. A fitness function is proposed for balancing energy consumption in the cluster. The final cluster-head collects node status information from their members and sets up a TDMA schedule for all the members. Every member of the

cluster is allocated a time slot. In the cluster steady phase, each member node sends data messages in its time slot at the idle state of a frame. In order to avoid collisions during communication, a kind of CSMA model is set up.

In the steady phase, the final cluster-head collects the data acquired from the physical environment by all the members in its cluster and aggregates the data. Instead of transmitting the processed data to the base station directly, every cluster-head decides whether to choose another cluster-head as the next hop or not. Each cluster-head broadcasts its residual energy and position information to other cluster-heads. Then the cluster-head chooses the next hop using Dijkstra algorithm [14]. According to the result of computation, it decides whether to transmit the data to the base station directly or to send them to the next hop. The cluster-head allows the radio devices of each member to be turned off at all times, except during the transmission time. Data are sent from the cluster-head to the base station or next hop using a fixed spreading code and CSMA model. When a cluster-head has data to send to the base station (at the end of its frame), it must sense the channel to see if anyone else is transmitting data, using the base station spreading code. If so, the cluster-head waits to transmit the data. Otherwise, the cluster-head sends the data using the base station spreading code. The members communicate with their cluster-head using CSMA model to avoiding collisions. When one node drains its energy during its working time, it sends a death notification to the base station. Then the base station sends initialization notifications to all the nodes in the network. After receiving the initialization notifications, the network enters the cluster formation phase at the end of one round. If no node dies in the network, the network enters the cluster-head adjustment phase at the end of one round. In this phase, the cluster-head of last round will act as the auxiliary cluster-head, and every member node sends its position and residual energy information to it. After computation using BFOA by the auxiliary cluster-head, the new cluster-head in this round can be chosen out.

As the working hours goes by, the sensor network enters one working phase after another. All the nodes are constructed into a cluster network. The base station collects the physical environment information from the nodes distributed in the specific area, after processing the data it transmits the compressed data to clients.

Fig. 1 shows the time line of LEICP operation.

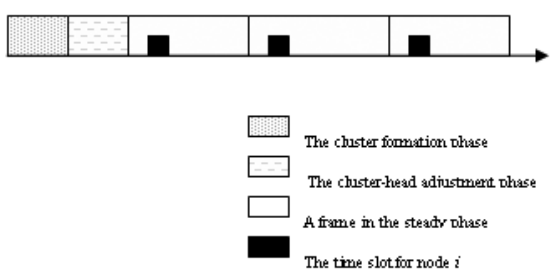

Fig. 1. Time line showing LEICP operation

The second phase is very important among the three. The auxiliary cluster-head uses BFOA algorithm to compute the position of the final cluster-head in this phase. Then it sends notifications to all nodes in the cluster. Aiming at balancing the energy consumption among all the nodes and reducing energy dissipation of the cluster-head in the cluster a fitness function is proposed as follows:

$$f(i) = \eta \cdot e_i + \frac{\lambda}{n-1} \cdot \sum_{k=1,k\neq i}^{n} \frac{2}{\pi} \cdot e_k \cdot \arctan\left|d_{ik} - d_{ave}\right| \quad (7)$$

$$d_{ave} = \frac{1}{n(n-1)} \cdot \sum_{i=1}^{n} \sum_{k=1,k\neq i}^{n} d_{ik},$$

where $\eta + \lambda = 1$, $\eta, \lambda \in [0,1]$. n is the number of nodes in the cluster. e_i is the residual energy of the ith node.

This function takes the residual energy and positions of nodes into consideration. It tries to balance the energy consumption in the cluster. According to this fitness function, the final cluster-head should be in such a position that the farther the member is from it, the more residual energy the member should own. The character of

$$\frac{2}{\pi} \cdot e_k \cdot \arctan\left|d_{ik} - d_{ave}\right|$$

provides us this guarantee. Where e_k is the residual energy of the kth node, d_{ave} is the average distance in a cluster and d_{ik} is the distance between the ith node and the kth node. When $d_{ik} > d_{ave}$, the larger d_{ik} is than d_{ave}, the larger the value of the function is. When $d_{ik} < d_{ave}$, the smaller d_{ik} is than d_{ave}, the larger the value of the function is.

Let

$$u(d_{ik}) = \frac{2}{\pi} \cdot e_k \cdot \arctan\left|d_{ik} - d_{ave}\right| . \quad (8)$$

Supposing d_{ave}=29.2226, e_k=0.25 and d_{ik} varies from 8.8299 to 45.2225, Fig. 2 shows the character of $u(d_{ik})$.

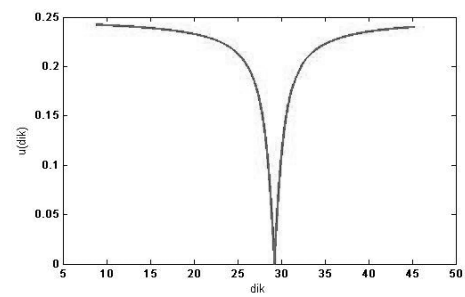

Fig. 2. $u(d_{ik})$

In the cluster-head adjustment phase of every round, every node's residual energy value can be regarded as a fixed value and d_{ik} is a variable. During the chemotactic steps, reproduction steps and elimination-dispersal events of bacteria (nodes), d_{ik} is changing.

Let

$$u(e_k) = \frac{2}{\pi} \cdot e_k \cdot \arctan\left|d_{ik} - d_{ave}\right| \quad (9)$$

Supposing d_{ave}=29.2226, d_{ik} =45.2225 and e_k varies from 0 to 0.25, $u(e_k)$ reflects a linearity character.

Supposing d_{ave}=29.2226, d_{ik}=8.8299 and e_k varies from 0 to 0.25, $u(e_k)$ reflects a similar characteristic to that when d_{ik} = 45.2225.

According to the analysis we can know that if $d_{ik} > d_{ave}$ or $d_{ik} < d_{ave}$, the larger the value of e_k is, the larger the result of $u(e_k)$ is. If the distance value between the *i*th node and the *k*th node is a fixed value, the value of fitness function increases when the value of e_k increases.

In our research, we consider a sensor node a bacterium and a cluster a bacteria colony. Although all the nodes are immobile, in order to find the most suitable node to act as the final cluster-head we suppose that every bacterium could experience Run and Tumble, adjusting its position in the chemotactic processes. For every node,

$$J(i) = f(i). \tag{10}$$

The second part of $f(i)$,

$$\frac{\lambda}{n-1} \cdot \sum_{k=1,k\neq i}^{n} \frac{2}{\pi} \cdot e_k \cdot \arctan\left|d_{ik} - d_{ave}\right|$$

implies a cell-to-cell swarming process. The problem of finding the final cluster-head can be transformed into solving a maximum value problem.

After a number of iteration loops the best bacterium with the most suitable position can be found. Then the most suitable position is mapped into one of the real positions of the nodes in the cluster. The node in this corresponding position will be selected to be the final cluster-head. How is the most suitable position mapped into one of the real position of the nodes in the cluster? An equation is given as follows:

$$d_{\min} = \min\{\left\|P^b - P_1\right\|_2, \left\|P^b - P_2\right\|_2 \cdots \left\|P^b - P_i\right\|_2 \cdots \left\|P^b - P_n\right\|_2\} \tag{11}$$

where $P = \{P_1, P_2 \cdots P_i \cdots P_n\}$ is the real position of *n* node in a cluster, $P_i = \{X_i, Y_i\}$. $P^b = \{X^b, Y^b\}$, which is the most suitable position after the adjustment using BFOA. The real position of a certain node with $d_{\min}$ will be chosen to be the position of the final cluster-head, which means that the nearest node from the P^b in the cluster will act as the final cluster-head.

After the cluster-heads aggregate the data they communicate with each other in the steady phase of every round. Each one broadcasts its position and compressed data information to other cluster-heads. After receiving the information, each cluster-head chooses the optimal route for transferring its data. Dijkstra algorithm is a shortest path optimization algorithm, which is used by cluster-heads to choose the next hop.

Fig. 3 shows the paths for data transmission of cluster-heads.

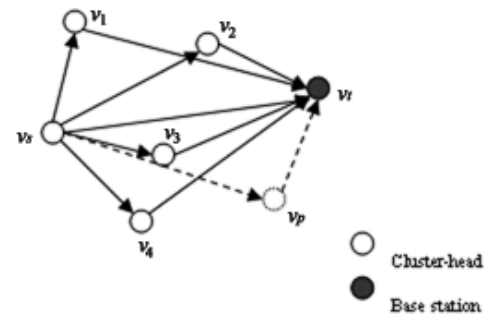

Fig. 3. Paths for data transmission of cluster-heads

$\{v_s, v_1, v_2 \ldots v_p\}$ is the set of cluster-heads of the sensor network. Node v_t is the base station. Node v_s is the source node. Node v_s can choose another cluster-head as the next hop or send its data to Node v_t directly.

We present the data transmission network by a directed weighted graph $D=(V, A)$, where V is a nonempty set of nodes (vertexes) and A is a set of arcs. v_i and v_j are two nodes in the graph. For the arc $a=(v_i, v_j)$, $w(a) = w_{ij}$, which represents the weight of a. Here, w_{ij} represents the wasting energy of node v_i. If node v_i transfers data to node v_j,

$$\begin{aligned} w_{ij} &= l \cdot E_{elec} + l \cdot \varepsilon_{FS} d_{ij}^{\ 2}, \quad if \quad d_{ij} < d_0 \\ &= l \cdot E_{elec} + l \cdot \varepsilon_{TR} d_{ij}^{\ 4}, \quad if \quad d_{ij} \geq d_0 \end{aligned} \tag{12}$$

where d_{ij} is the distance between v_i and v_j . If v_j is the second hop node chosen by another cluster-head, v_t is the base station,

$$\begin{aligned} w_{jt} &= 2 \cdot l \cdot E_{elec} + l \cdot \varepsilon_{FS} d_{jt}^{\ 2}, \quad if \quad d_{jt} < d_0 \\ &= 2 \cdot l \cdot E_{elec} + l \cdot \varepsilon_{TR} d_{jt}^{\ 4}, \quad if \quad d_{jt} \geq d_0 \end{aligned} \tag{13}$$

because it includes a receiving consumption.

L is a path from node v_s to node v_t. The weight of L, $w(L)$, is the sum of all the weights of arcs in L. The shortest path weight, also called distance, from node v_s to node v_t, denoted $d(v_s, v_t)$ or d_{st}, is the minimum weight of all possible directed paths with origin v_s and destination v_t.

In the specific steps of Dijkstra algorithm, we use Q and T to represent the Q sign and T sign of some node. S_i represents the set of nodes with Q sign in the *i* step. In order to get the shortest path and paths from node v_s to other nodes we set the λ value for each node. When the algorithm comes to the end, $\lambda(v)=m$, it represents the front node of v is v_m in the shortest path from v_s to v. if $\lambda(v)=M$, it represents there is no path from v_s to v in D. if $\lambda(v)=0$, which represents $v= v_s$.

For a directed weighted graph $D=(V, A)$, the computation process is given in details.

At the initial stage (i=0), $S_0=\{ v_s \}$, $Q(v_s)=0$, $\lambda(v_s)=0$. To every v, if $v\neq v_s$, let $T(v) = +\infty$, $\lambda(v)=M$, and $k=s$.

Step 1: if $S_i=V$, the algorithm ends and to every $v\in S_i$ $d(v_s, v)= Q(v)$. Otherwise, go to Step 2.

Step 2: Consider every v_j , for which $(v_k, v_j) \in A$ and $v_j \notin S_i$.

If $T(v_j) > Q(v_k)+w_{kj}$, adjust $T(v_j)$ to be $Q(v_k)+w_{kj}$ and $\lambda(v_j)$ to be k. Otherwise, go to Step 3.

Step 3: Let $T(v_{ji}) = \min_{v_j \notin S_i}\{T(v_j)\}$.

If $T(v_{ji})<+\infty$, change the T sign of v_{ji} to the Q sign, $Q(v_{ji})=T(v_{ji})$. Let $S_{i+1}=S_i \cup \{ v_{ji} \}$, $k=ji$, $i=i+1$ and go to Step 1. Otherwise, the algorithm ends. Now for every $v \in S_i$, $d(v_s, v)= Q(v)$ and to $v \notin S_i$, $d(v_s, v)=T(v)$.

Every cluster-head can get the optimization path and choose another one for the next hop or send its data to the base station directly. This process is different from LEACH, in which the cluster-heads collect the data information from their members and send the processed data information to the base station directly. For the unevenly distribution of cluster-heads, some cluster-heads are far from the base station or own too many members. They will consume too much energy during the messages transmission. For LEICP, the cluster-head can choose another cluster-head as the next hop. Will it save energy or not if the cluster-head chooses any one to act as the next hop? Even though the transceiver of the node consumes energy during the communication, Dijkstra algorithm provides us a way to find the optimal path. The simulation in the following section results will show that the network with LEICP saves much energy.

V. Simulation Analysis

A. Simulation Parameters Setting

In our study, OMNET++ is used for simulation. Twenty nodes are distributed randomly in a 150m×150m rectangle field. Each node can communicate with any other node in this field. The base station is set at the (100, 100) point of the rectangle field.

Initial energy level of each node is 0.25J. We suppose that in every frame every node sends l=1000bits data to its cluster-head in its time slot. The probability of being cluster-head is 25%. That means that no more than 5 cluster-heads will be selected in one round among the 20 nodes.

In our research, when calculating the fitness value $J_i(j,r)$ for the ith bacterium at the jth chemotactic step in the rth iteration loop, η is 0.5 and λ is 0.5. When using the BFOA algorithm for adjusting the cluster-head position, C is 1.0 m. N_c=5, N_s=3, N_{re}=2 and N_{ed}=2. In the elimination-dispersal steps, the probability p_{ed} is 0.15.

B. Simulation Results

Simulations are performed for LEICP, and the results are compared with those of LEACH.

Fig. 4 shows the residual energy status of the sensor network in working period. We only consider the energy consumption of data transmissions in the network. The lower line shows the residual energy status of network with LEACH; the upper line shows that with LEICP. The residual energy of the network decreases for transmitting data messages. The cluster-heads consume energy more rapidly than the member nodes. We ignore the energy consumption of the base station in simulation.

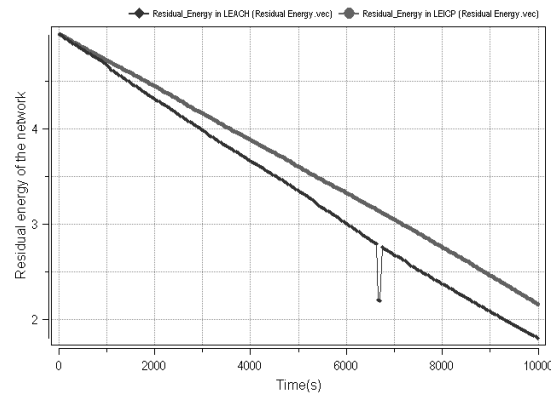

Fig. 4. The residual energy of the network

The cluster-head collects residual energy information of its cluster and sends it to the next hop or the base station directly. As for LEACH, the number of cluster-heads is not uniform in different rounds. If one cluster-head owns too many members, in one frame some members can't send their data and residual energy information to their cluster-heads in their allocated time slots. Fig. 4 shows this phenomenon. We see a spike pulse at 6625.8s in the figure. The base station can't compute the residual energy of the whole network because some cluster-head owns too many members for the random character of cluster-head choosing equation (1). At 6745s, the first node in the network with LEACH dies. As for LEICP, the number of cluster-heads is uniform. It uses BFOA and Dijkstra algorithm to adjust the positions of cluster-heads, balance energy consumption of every cluster and reduce the energy consumption of the cluster-heads. Unlike LEACH, the cluster-head chooses an optimal path for transferring data in LEICP. The energy consumption of the network with LEICP is reduced greatly. The simulation result shows that LEICP can acquire more residual energy than LEACH.

At the time t_s, if e_m is the residual energy of network with LEACH and e_n is that with LEICP, let $p_s = (e_n - e_m)/ e_m$. Fig. 5 shows the variety of p_s. As the working hours of the sensor network goes by, LEICP saves more energy than LEACH. At 10000s in the simulation, LEICP saves energy by about 20% compared with LEACH.

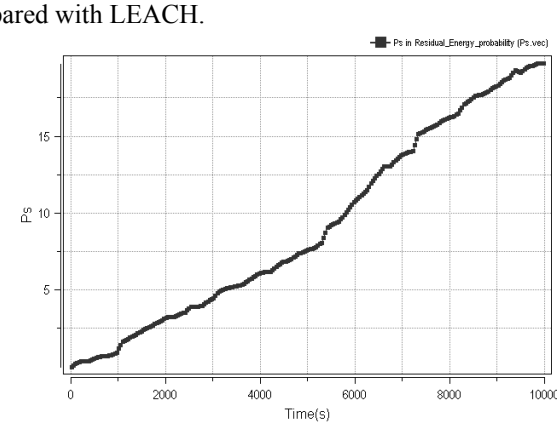

Fig. 5. p_s

After a node drains its energy it dies, and it can't communicate with other nodes any more. We run the simulation

for 8000s, 10000s, 12000s, 15000s, and 20000s respectively, and find that LEICP can prolong the lifetime of the sensor network greatly. Fig. 6 shows the simulation result of 15000s. The upper line shows the dead node number of the network with LEACH; the lower line shows that with LEICP. LEICP balances the energy consumption of the cluster and reduces the energy consumption of cluster-heads. The simulation result shows that LEICP saves much energy in working hours. At 6745s, the first node in the network with LEACH dies. The time for the first dead node coming forth is deferred to 10946s in the network with LEICP. LEICP prolongs the lifetime of the sensor network by about 62.28% compared with LEACH in this run. When the simulation ends there are 8 dead nodes in the network with LEACH, while there are 3 dead nodes in the network with LEICP. The energy consumption of network with LEICP is reduced greatly, which can provide a satisfactory result.

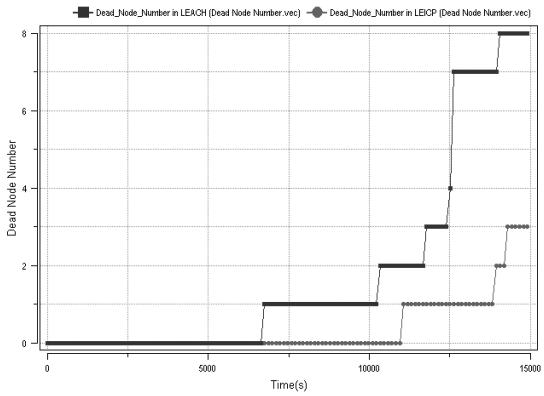

Fig. 6. The dead node number of the network

Fig. 7 shows the number of messages sent in the network with LEACH. During the working period of the sensor network, there are lots of messages sent by the nodes. We let the base station compute the number of messages sent in the network. Because the number of cluster-heads is not uniform in the network with LEACH, the number of messages sent in the network varies significantly in every round. At 6745s, the first node dies. The total number of messages decreases. So, the figure can be divided into two parts at 6745s. The average number of messages of the front part is larger than that of the latter part. We can see this change from the Fig. 7.

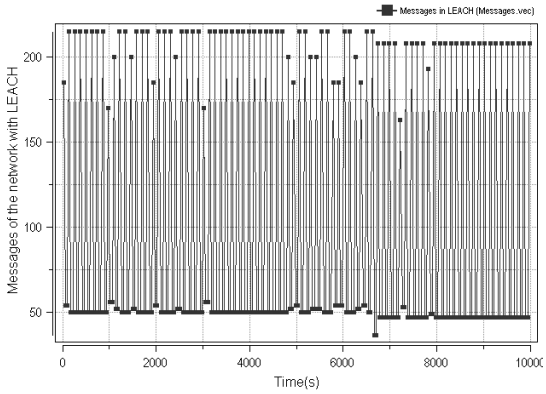

Fig. 7. Messages of the network with LEACH

Fig. 8 shows the number of messages sent in the network with LEICP. According to the cluster-head selection procedure of LEICP, the network owns uniform number of cluster-heads in every round if no node dies. The positions of the cluster-heads are varying in the cluster during every round, but clusters are fixed. The number of messages sent in the network with LEICP is more uniform than that with LEACH. At the initial state in the cluster formation phase of the first round, every node sends its status information to the base station, and the base station computes the auxiliary cluster-heads and the clusters. Then it sends notifications to all the nodes in the network. So, the number of messages of the first round is larger than that of the latter rounds before the first node dies.

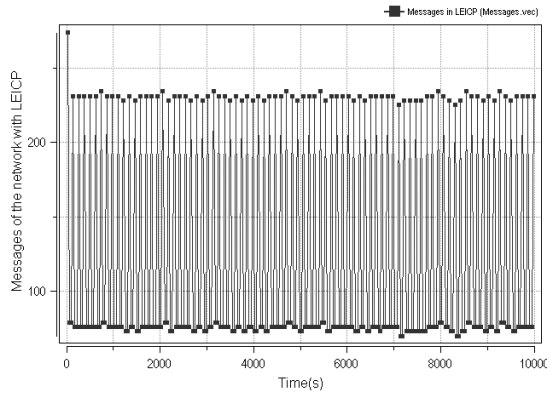

Fig. 8. Messages of the network with LEICP

As for the topology of the sensor network, if one node is selected to be a cluster-head of the network its in-degree reflects the number of its member nodes. Fig. 9 shows the in-degree of node 2 in the network with LEACH and that with LEICP. As for LEICP, the number of cluster-heads in every round is uniform if no node dies according to the cluster-head selection procedure. In addition, the positions of clusters and number of members in every cluster are fixed if no node dies. So, we can see that the in-degree of the node in the network with LEICP is more uniform than that with LEACH.

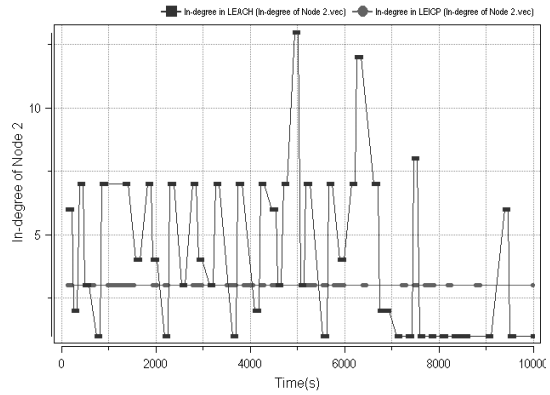

Fig. 9. In-degree of Node 2

Now we consider the influence on the sensor network in small scale from the variation of BFOA parameters (N_c, N_s, N_{re}, N_{ed}, p_{ed}, C).

When one parameter of BFOA varies and other parameters remain invariable, we consider its influence on the residual

energy of the network. The simulation results show that the variation of the parameter has little influence on the results. When its value increases, the residual energy of the sensor network keeps the same level during the working period. For example, p_{ed} = 0.15, 0.25, 0.35, 0.45, 0.55, 0.65 respectively and other parameters remain invariable, Fig. 10 shows the influence on the residual energy of network from the variation of p_{ed}.

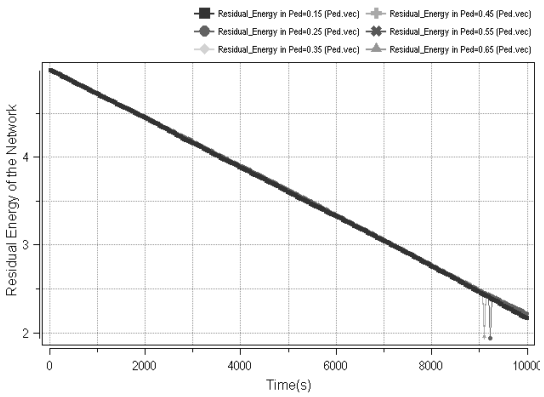

Fig. 10. The residual energy of the network (p_{ed} varies)

From Fig. 10 we can see that the variation of BFOA parameters has little influence on the sensor network of small scale. For the small-scale network, there is small number of nodes in every cluster. The variation of BFOA parameters has little influence on the selection of final cluster-head in every cluster. So it has little influence on the working process of the wireless sensor network. For the large-scale wireless sensor network, there is large number of members in every cluster and the cluster-head will consume more energy during its work time. If parts of nodes' distances from the sink are larger than their communication radius, we need to select various routing strategies, which will be investigated in future study.

VI. Conclusions and Future Work

In this paper, a low energy intelligent clustering protocol named LEICP is proposed. LEICP aims at adjusting the positions and reducing the energy dissipation of the cluster-heads. It uses BFOA and Dijkstra algorithm to balance the energy consumption in the cluster and reduce the energy consumption of the whole wireless sensor network. It avoids constructing the cluster network in every round, reducing the number of messages sent to the base station in the sensor network. LEICP prolongs the lifetime of the sensor network by about 62.28% compared with LEACH and saves much energy in the lifetime of the sensor network. The number of messages in the network with LEICP is more uniform than that with LEACH. The cluster-head can own uniform in-degree in the network with LEICP. The variation of BFOA parameters has little influence on the sensor network of small scale. In the future research, we will pay more attention to the influence of the BFOA parameters on the large-scale wireless sensor network.

References

[1] I. F. Akyilldiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag. USA,* vol. 40, pp.102-114, August 2002.

[2] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences. USA*, vol. 2, pp. 1-10, January 2000.

[3] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wirel. Commun. USA,* vol. 1, pp. 660-670, October 2000.

[4] W. Y. Zhang, Z. Z. Liang, Z. G. Hou, and M. Tan, "A power efficient routing protocol for wireless sensor network," *2007 IEEE/ACS International Conference on Computer Systems and Applications. Jordan*, pp. 20-25, May 2007.

[5] N. M. Abdul Latiff, C. C. Tsimenidis, and B. S. Sharif, "Energy-aware clustering for wireless sensor networks using particles swarm optimization," *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications. Greece*, pp. 2713-2717, September 2007.

[6] Y. Liang, H. B. Yu, and P. Zeng, "Optimization on routing protocol of wireless sensor network based on clustering using PSO," *Control and Decision. China,* vol. 21, pp. 453-456, 461, April 2006.

[7] K. M. Passino, "Distributed optimization and control using only a germ of intelligence," in *Proceedings of the 2000 IEEE International Symposium on Intelligent Control. Held jointly with the 8th IEEE Mediterranean Conference on Control and Automation. Greece*, P5-P13, July 2000.

[8] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst Mag. USA*, vol. 22, pp. 52-67, June 2002.

[9] Y. Liu, and K. M. Passino, "Biomimicry of social foraging bacteria for distributed optimization: models, principles and emergent behaviors," *J. Optim. Theory Appl. USA*, vol. 155, pp. 603-28, December 2002.

[10] M. Tripathy, S. Mishra, L. L. Lai, and Q. P.Zhang, "Transmission loss reduction based on FACTS and bacterial foraging algorithm," *Parallel Problem Solving from Nature. Springer Berlin/ Heidelberg*. vol. 4193, pp. 222-231, October 2006.

[11] A. Abraham, A. Biswas, S. Dasgupta, and S. Das, "Analysis of reproduction operator in bacterial foraging optimization algorithm," *IEEE Congr. Evol. Comput., CEC. China*, pp. 1476-1483, January 2008.

[12] S. Dasgupta, A. Biswas, A. Abraham, and S. Das, "Adaptive computational chemotaxis in bacterial foraging algorithm," *Proc. - CISIS, Int. Conf. Complex, Intelligent Softw. Intensive Syst. Spain*, pp. 64-71, March 2008.

[13] B. Niu, Y. L. Zhu, X. X. He, H. Shen, and Q. H. Wu, "A lifecycle model for simulating bacterial evolution," *Neurocomput. Netherlands*, vol. 72, pp. 142-148, December 2008.

[14] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*. vol. 1, pp. 269-271, 1959.