Energy-Efficient Clustering Method for Data Gathering in Sensor Networks

Junpei Kamimura, Naoki Wakamiya, Masayuki Murata
Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
{kamimura, wakamiya, murata} @ist.osaka-u.ac.jp

Abstract

By deploying wireless sensor nodes and composing a sensor network, one can remotely obtain information about the behavior, conditions, and positions of entities in a region. Since sensor nodes operate on batteries, energy-efficient mechanisms for gathering sensor data are indispensable to prolong the lifetime of a sensor network as long as possible. A sensor node consumes energy: observing its surroundings, transmitting data, and receiving data. Cluster-based data gathering mechanisms have been proposed based on a model where energy consumption in data transmission is proportional to the square of the radius of the radio signal. In clustering sensor nodes, we need to consider that a cluster-head consumes more energy than the others when receiving data from cluster members, fusing data to reduce the size, and sending the aggregated data to a base station. In this paper, we proposed a novel clustering mechanism where clusters are organized in a distributed and energy-efficient way through local communication among neighboring sensor nodes. Through simulation experiments, we showed that our mechanism can gather data from more than 80% of the sensor nodes longer than LEACH by over 25%.

1 Introduction

With recent advancements and developments in Micro Electro Mechanical System (MEMS) technologies, low-cost and low-power consumption wireless micro sensor nodes have become available. A sensor node has one or more sensors, a general purpose processor with limited computing power and memory, a radio transceiver, that operates on batteries. By deploying sensor nodes and composing a sensor network, one can remotely obtain information about behavior, conditions, and the position of entities in the region [1]. A sensor network consists of hundreds or thousands of wireless sensor nodes distributed in a region in uncontrolled and unorganized ways. Sensor data obtained at sensor nodes are sent to a base station through wireless communications (Fig. 1). A base station summarizes collected data and presents them to a user or sends them to a remote host. Since sensor nodes derive power from disposable batteries, an energy-efficient data gathering mechanism is indispensable to observe the region as long as possible.

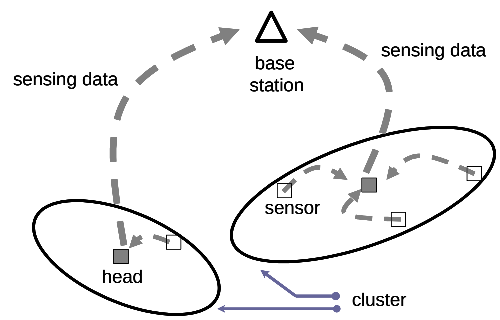

Figure 1. Cluster-based data gathering in sensor networks

A sensor node consumes its energy in monitoring its environment and receiving and sending radio signals. The amount of energy consumed in a radio transmission is proportional to the square of the range of the radio signal propagation [2]. Since the distance from sensor node to sensor node is shorter than from sensor node to the base station, it is energy-inefficient for all sensor nodes to send their data directly to a distant base station. Therefore, cluster-based data gathering mechanisms effectively save energy [2-6]. In cluster-based mechanisms, groups of neighboring sensor nodes form clusters. In each cluster, one representative node called a cluster-head gathers sensor data from its members and sends the collected data to a base station. In LEACH [2], which is well-known and widely referred to, a pre-determined percentage of sensor nodes become cluster-heads which advertise their candidacy to the rest of the sensor nodes. Hearing advertisements, each sensor node chooses the closest cluster-head and registers itself as a cluster member. Eventually clusters are formed. Cluster members send their sensor data to a cluster-head which combines all n-bit data into single n-bit data and sends them to a base station. Since a cluster-head expends more energy than its members in advertising and receiving, fusing, and emitting data to a base station, LEACH rotates the role of cluster-head among sensor nodes. As a result, energy consumption is equalized among sensor nodes, extending the life of the sensor network.

LEACH's clustering algorithm assumes that sensor nodes are homogeneous and equal. In reality, however, their battery capacities are different, and the amount of energy consumed in gathering data also differs among cluster-heads, depending on the number of cluster members and their positions in the region. Energy consumption also differs among cluster members due to the distance to a cluster-head. Some sensor nodes might also be deployed later for denser observations. Consequently, residual energy is different among sensor nodes. In addition, the optimum percentage of cluster-heads has to be determined in advance, considering the topology of a sensor network. Therefore, LEACH cannot adapt to such changes in sensor networks as the addition, removal, and transfer of sensor nodes, although the percentage of cluster-heads considerably affects the efficiency of data gathering. Finally, for organized clusters to cover an entire sensor network, each cluster-head must broadcast its own advertisement to all the other nodes, another inefficient use of energy. To tackle the problem, in [5], they proposed two variations of LEACH. LEACH-C (LEACH-centralized) is a centralized protocol, which takes into account the residual energy in choosing sensor nodes for cluster-heads. The other is a distributed but less efficient implementation of LEACH-C.

In sensor networks consisting of hundreds or thousands of sensor nodes, it is impractical to employ a centralized mechanism to organize clusters. Such a mechanism is useful where sensor nodes autonomously form appropriate clusters through local communications. In biology, ants and other social insects construct clusters, i.e., colonies, parties, and cemeteries in self-organizing ways [7, 8]. Taking inspiration from such biological systems, much research has been done in the fields of data clustering and graph partitioning [9-11]. [9] proposed an algorithm, ANTCLUST, based on an ant model of colonial closure, to solve clustering problems. Ants recognize each other by exchanging a chemical substance. If they are similar, the ant is welcomed and treated as a member of the same nest. In ANTCLUST, two randomly chosen objects meet. Based on their similarity, a cluster is created, merged, or deleted. By repeating meetings, an appropriate set of clusters is eventually formed so that similar objects are accommodated in the same cluster.

In this paper, based on ANTCLUST, we propose a novel clustering method that organizes energy-efficient clusters through local interactions among neighboring sensor nodes. In our method, sensor nodes with more residual energy independently become cluster-heads. Sensor nodes meet through local radio communications where information about clusters is advertised. Each sensor node with less residual energy chooses a cluster based on the residual energy of the cluster-head, distance to the cluster-head, and an estimation of cluster size. Energy-efficient clusters are eventually formed that extend the life of the sensor network.

The paper consists of the following sections. Section 2 explains the hypotheses of sensor networks considered in this paper. Section 3 introduces ANTCLUST, a clustering algorithm on which our method is based. In Section 4, we propose a new clustering method for energy-efficient data gathering in sensor networks. Results of simulation experiments are given in Section 5. Finally, Section 6 concludes the paper and describes future research.

2 Sensor Network

In this paper, we propose a method for cluster-based data gathering mechanisms in sensor networks. We consider an application where sensor data are gathered from all sensor nodes to a base station at regular intervals and/or on demand. Sensor nodes operate on energy-limited, irreplaceable batteries. The capacity of batteries can differ among sensor nodes. Sensor nodes have a wireless transmitter and receiver. The range of radio signals can be adjusted. In addition, we assume that sensor nodes can aggregate or fuse multiple data into single-sized data [12]. To avoid installation cost and the need for careful planning, sensor nodes are deployed in the region to monitor freely. Sensor nodes stop due to a loss of battery power, move from one place to another, and are deployed later. Sensor nodes can determine their absolute or relative geometrical positions by using Global Positioning System (GPS) or a position detection system [13-15]. However, the number of applications where these assumptions hold is limited. In our next work, we consider to adopt our method to sensor networks where some of these assumptions are not valid.

We use the same energy consumption model as [3] in whose model a sensor node consumes E_{elec} (nJ/bit) in transmitter or receiver circuitry and ε_{amp} (pJ/bit/m^2) in transmitter amplifier. A sensor node expends energy $E_{T_x}(k,d)$ or $E_{R_x}(k)$ in transmitting or receiving a k-bit message to or from distance d, given by the following equations:

$$E_{T_x}(k,d) = E_{elec} \times k + \varepsilon_{amp} \times k \times d^2 \quad (1)$$

$$E_{R_x}(k) = E_{elec} \times k. \quad (2)$$

A sensor node also consumes E_{fuse} (nJ/bit/message) in aggregating multiple sensor data into one.

3 ANTCLUST

Ants synthesize a chemical substance called colony odor which differs by individuals, species, and environment; they spread it on their cuticles [16, 17]. When two ants meet, they recognize whether they belong to the same nest by exchanging and comparing these chemical substances, which is updated at each meeting. After spending some time in the nest and repeatedly meeting other ants, a young ant can prepare an appropriate chemical substance to recognize its mates.

ANTCLUST is a clustering algorithm which applies a colonial closure model and regards an object as an ant and a cluster as a nest [9]. A similarity $Sim(i, j) = [0, 1]$ is defined between a pair of objects i and j. Each object i has a cluster identifier, $Label_i$, an acceptance threshold of similarity, $Template_i$, an estimator of cluster size, $M_i = [0, 1]$, and an estimator, $M_i^+ = [0, 1]$, which measures how well the object is accepted in the cluster. They are initialized as $Label_i = 0$, $M = 0$, and $M^+ = 0$. $Template_i$ is defined through a learning phase where object i experiences random meetings.

$$Template_i \leftarrow \frac{\overline{Sim(i,\cdot)} + Max(Sim(i,\cdot))}{2}. \tag{3}$$

$\overline{Sim(x,\cdot)}$ and $Max(Sim(x,\cdot))$ represent the average and the maximum value of similarity between object x and all object that object x has met, respectively. In ANTCLUST, two randomly chosen objects meet. Based on their similarity, threshold values, and clusters, they create, merge, or delete clusters. By repeatedly conducting random meetings, clusters are appropriately organized so that objects in the same cluster become more similar with one another than those in different clusters.

We consider here the case when two objects i and j meet. First, two objects i and j decide whether they accept their counterpart according to similarity $Sim(i, j)$ and threshold values $Template_i$ and $Template_j$.

$$\begin{aligned} Acceptance(i,j) \Leftrightarrow\ & (Sim(i,j) > Template_i) \\ & \wedge (Sim(i,j) > Template_j). \end{aligned} \tag{4}$$

Then $Template_i$ and $Template_j$ are updated by Eq. (3).

Next, their $Label$s are compared. When neither of them belongs to any cluster and they accept each other, a new cluster is created.

$$\begin{aligned} & Label_i \leftarrow Label_{NEW},\ Labael_j \leftarrow Label_{NEW} \\ & \text{if } (Label_i = Label_j = 0) \wedge \big(Acceptance(i,j) = True\big). \end{aligned} \tag{5}$$

If one of two objects, say object i, does not belong to any cluster, and if they accept each other, object i joins the cluster of the other.

$$\begin{aligned} & Label_i \leftarrow Label_j, \\ & \text{if } (Label_i = 0) \wedge (Label_j \neq 0) \\ & \quad \wedge \big(Acceptance(i,j) = True\big). \end{aligned} \tag{6}$$

When two objects belong to the same cluster and they accept each other, they increase their size estimate of their cluster:

$$\begin{aligned} & M_i \leftarrow (1-\alpha)M_i + \alpha,\ M_j \leftarrow (1-\alpha)M_j + \alpha, \\ & M_i^+ \leftarrow (1-\alpha)M_i^+ + \alpha,\ M_j^+ \leftarrow (1-\alpha)M_j^+ + \alpha, \\ & \quad \text{if } (Label_i = Label_j) \wedge (Label_i \neq 0) \\ & \qquad \wedge (Label_j \neq 0) \wedge \big(Acceptance(i,j) = True\big). \end{aligned} \tag{7}$$

Here, α is a constant between 0 and 1. When two objects belong to the same cluster and they reject each other, they first update the size of their estimates:

$$\begin{aligned} & M_i \leftarrow (1-\alpha)M_i + \alpha,\ M_j \leftarrow (1-\alpha)M_j + \alpha, \\ & M_i^+ \leftarrow (1-\alpha)M_i^+,\ M_j^+ \leftarrow (1-\alpha)M_j^+, \\ & \quad \text{if } (Label_i = Label_j) \wedge (Label_i \neq 0) \\ & \qquad \wedge (Label_j \neq 0) \wedge \big(Acceptance(i,j) = False\big). \end{aligned} \tag{8}$$

Here, an object x with a smaller estimate loses its cluster.

$$Label_x \leftarrow 0,\ M_x \leftarrow 0,\ M_x^+ \leftarrow 0, \tag{9}$$

where $x = (x|M_x^+ = Min_{k\in[i,j]} M_k^+)$. When two objects belong to different clusters and they accept each other, they first estimate how the size of their clusters will be changed:

$$\begin{aligned} & M_i \leftarrow (1-\alpha)M_i,\ M_j \leftarrow (1-\alpha)M_j, \\ & \text{if } (Label_i \neq Label_j) \wedge (Label_i \neq 0) \\ & \quad \wedge (Label_j \neq 0) \wedge \big(Acceptance(i,j) = True\big). \end{aligned} \tag{10}$$

Then, an object x in a smaller cluster changes its cluster.

$$Label_x \leftarrow Label_{(k|k\in(i,j),k\neq x)}, \tag{11}$$

where $x = (x|M_x = Min_{k\in[i,j]} M_k)$. When none of the above conditions holds, nothing happens.

4 ANTCLUST-based energy-efficient clustering method

4.1 Proposal Outline

In this paper, based on ANTCLUST, we propose a novel clustering method for energy-efficient data gathering in sensor networks. In our method, we regard a sensor node as an ant and a cluster as a nest. The similarity of one sensor node to another corresponds to the distance from the sensor node to the cluster-head of another. Sensor nodes meet through wireless communications. Since a sensor node usually has an omni-antenna, a radio signal is broadcast nearby, and it is received by all sensor nodes within its transmission range. In addition, it is a one-way communication, while in ANTCLUST both encountered objects adjust their clusters.

A cycle of data gathering, called a round, consists of four phases: (i) cluster-head candidacy, (ii) cluster formation, (iii) registration, and (iv) data transmission. The behavior of sensor nodes in the third and fourth phases is identical to the cluster set-up phase through the data transmission phase in LEACH. In the cluster-head candidacy phase, a sensor node that has decided to become a cluster-head broadcasts an advertisement within a limited range that message contains its identifier, residual energy, and coordinates. In our method, all sensor nodes initially consider themselves candidates for cluster-head. Those sensor nodes that receive advertisements from other sensor nodes abandon their candidacy and join a cluster. In the cluster formation phase, sensor nodes meet through radio communications. A percentage of sensor nodes that are not cluster-heads broadcast information about themselves. The message is in the same form as advertisements for candidacy. Each of the neighboring sensor nodes which receive broadcast messages determines which cluster to join based on information about its own cluster and the newly advertised clusters. Next in the registration phase, each sensor node registers itself as a cluster member by sending a registration message to a cluster-head. After receiving registration messages from all of its members, a cluster-head creates a TDMA schedule and notifies cluster members of the schedule via broadcasting. In the data transmission phase, cluster members send their data to the cluster-head according to the specified TDMA schedule. The cluster-head receives its members' data, aggregates them into one, and sends it to a base station using different CDMA codes among clusters. All communications except those noted with TDMA or CDMA are performed by CSMA. The beginning of each round and the timing of phase-changes are synchronized among sensor nodes.

In LEACH, to organize clusters in a round, a predetermined percentage of sensor nodes consume energy by broadcasting their candidacy within the entire sensor network, and the other sensor nodes expend energy by registering themselves as cluster members. In our mechanism, on the other hand, some sensor nodes consume energy by broadcasting their candidacy within the limited range, and the other sensor nodes expend energy in local communications for meetings and registering themselves as cluster members. Hence, to attain more energy-efficient clustering than LEACH, the range of broadcastings for candidacy and meeting should be limited. In ANTCLUST, a sufficient number of meetings is repeated until clusters become stable. On the other hand, in our method, we limit the number of social sensor nodes that advertise their cluster information per round to avoid excessive energy consumption. Detailed comparisons between our method and ANTCLUST will be given in subsection 4.4.

The following two subsections give details of the cluster-head candidacy phase and the cluster formation phase. A sensor node has an unique identifier i, residual energy e_i, and coordinates c_i. For constructing clusters, a sensor node also maintains an identifier $head_i$ of a cluster-head, residual energy E_i of a cluster-head, coordinates C_i of a cluster-head, an estimator M_i of the number of cluster members, a threshold value $Template_i$, the probability $P_i = [0, 1]$ of its own candidacy, a radius R for broadcasting its candidacy for cluster-head, a radius $r < R$ for broadcasting cluster information for meetings, and proportion $P_{ex} = [0, 1]$ of social sensor nodes that cause meetings. Among the eight parameters, threshold $Template_i$ and probability P_i of candidacy are initialized when sensor node i is deployed, and they are updated every round. The last three parameters are also initialized at deployment, but they can be adjusted according to conditions surrounding sensor node i. The other parameters are initialized at the beginning of each round.

Empirically, P_i is set at 0.5 for homogeneous sensor nodes with the same amount of initial energy. The initial value of P_{ex}, also identical among sensor nodes, is chosen between 0% to 20%. When P_{ex} is set at 0%, there is no meeting and each node chooses its cluster from those candidacy messages that it directly received from cluster-heads. In this case, we cannot expect equalization of residual energy among sensor nodes, but we can reduce energy consumption in cluster formation. On the other hand, as P_{ex} increases, sensor nodes often find other clusters and a chance to join a better cluster. As a result, the lifetime of a sensor node can be extended at the sacrifice of energy consumed in meetings. This means that there is a tradeoff in P_{ex}. We show some simulation results to investigate the effect of P_{ex} later.

4.2 Cluster-head candidacy phase

At the beginning of a round, all sensor nodes consider themselves candidates for cluster-head. Parameters are initialized as:

$$head_i \leftarrow i,\ E_i \leftarrow e_i,\ C_i \leftarrow c_i,\ M_i \leftarrow 1. \tag{12}$$

Assuming that the cluster-head candidacy phase has a T time unit duration, sensor node i announces its candidacy within the radius of R at $T(1 - P_i)$. To prolong the lifetime of a sensor network, energy consumption among sensor nodes must be balanced. If some sensor nodes die of starvation, the distance from cluster members to the cluster-head increases, so gathering sensor data consumes more energy. In addition, information about some parts of the region cannot be obtained due to a decline in the number of active sensor nodes. So in our method sensor nodes with more residual energy are more likely to become cluster-heads because they adjust P_i in accordance with the residual energy of neighboring sensor nodes as explained in the next subsection. An advertisement contains an identifier $head_i$, residual energy E_i, coordinates C_i, an estimator M_i of the number of cluster members, and its own residual energy e_i. When a candidacy is announced, E_i is obviously identical to e_i.

When a sensor node that has not yet announced its candidacy receives an advertisement message from another sensor node, it abandons its candidacy and becomes a member of the cluster. Furthermore, when a sensor node that already belongs to a cluster receives another advertisement message, it considers the offer and conducts the same procedure as in the next cluster formation phase to determine which cluster it should join.

4.3 Cluster formation phase

After the cluster-head candidacy phase, the cluster formation phase is initiated to organize better clusters through local communications, i.e., meetings, as in ANTCLUST.

At the end of the cluster-head candidacy phase, every sensor node belongs to a cluster as either a cluster-head or as a member. A percentage P_{ex} of sensor nodes in a sensor network decides to be social and broadcasts information about its clusters within a radius of r. On receiving an advertisement, sensor nodes within radio signal range meet the sensor node and find the cluster. The format for a meeting advertisement is the same as for candidacy. The way that a sensor node decides to induce a meeting is similar to the probabilistic decision algorithm for cluster-head candidacy in LEACH. If a sensor node is a cluster-head, it does not cause a meeting. Hereafter we describe a case where sensor node i received an advertisement from sensor node j.

If sensor node i is not a cluster-head, then it adjusts its cluster. First, sensor node i decides whether to accept cluster-head $head_j$ to which sensor node j belongs by comparing the distance to $head_j$ with threshold $Template_i$.

$$Acceptance(i,j) \Leftrightarrow \big(d(i, head_j) \leq Template_i\big). \quad (13)$$

Here, $d(i, head_j)$ represents the distance between cluster-head $head_j$ and sensor node i derived from their coordinates c_i and C_j. If c_i and C_j are in the form of x, y, and possibly z coordinates, $d(i, head_j)$ is an Euclidean distance. When sensor node i accepts cluster-head $head_j$, that is, sensor node i considers that cluster-head $head_j$ is close enough, sensor node i compares the two clusters. If sensor nodes i and j belong to the same cluster, sensor node i

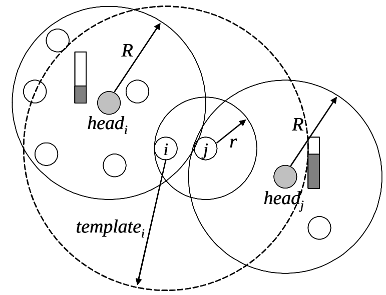

Figure 2. An example where $(head_i \neq head_j) \wedge (Acceptance(i,j) = True)$

increases its estimate of size.

$$M_i \leftarrow M_i + 1, \quad (14)$$
$$\text{if} \;\; (head_i = head_j) \wedge (Acceptance(i,j) = True).$$

If they belong to different clusters it implies that there is another cluster close to sensor node i, as illustrated in Fig. 2. Cluster-head $head_j$ is in sensor node i's template, but its advertisement has not been heard by sensor node i. For energy-efficient data gathering, it is effective for a sensor node i to choose a cluster that is closer to sensor node i since sensor node i can save energy by sending sensor data to a closer cluster-head. In addition, sensor node i should choose a cluster-head with more residual energy to avoid driving an energy-poor sensor node to starvation. Finally, a cluster with fewer members is preferred, since energy expended in gathering sensor data to a cluster-head is proportional to the number of cluster members. Thus, sensor node i changes its cluster:

$$head_i \leftarrow head_j,\ E_i \leftarrow E_j, \quad (15)$$
$$C_i \leftarrow C_j,\ M_i \leftarrow M_j + 1, \quad (16)$$
$$\text{if} \;\; (head_i \neq head_j) \wedge \big(Acceptance(i,j) = True\big)$$
$$\wedge \Big(\frac{E_j}{M_j \cdot d^2(i, head_j)} \geq \frac{E_i}{M_i \cdot d^2(i, head_i)}\Big). \quad (17)$$

Except in the above conditions, sensor node i does nothing for cluster formation.

Regardless of whether sensor node i is a cluster-head, it updates probability P_i of its cluster-head candidacy to reflect the relationship among its own residual energy e_i and that of sensor node j, e_j:

$$P_i \leftarrow \begin{cases} \min(1,\ P_i + p), & \text{if } e_i > e_j \\ \max(0,\ P_i - p), & \text{if } e_i < e_j \\ P_i, & \text{if } e_i = e_j. \end{cases} \quad (18)$$

Here, p is a constant value which satisfies $p = [0, 1]$. Thus, the probability of a candidacy is determined in relation to the residual energy of surrounding sensor nodes, not by its absolute amount. Next, sensor node i updates threshold $Template_i$:

$$Template_i = \frac{\overline{d(i,\cdot)} + Max(d(i,\cdot))}{2} \tag{19}$$

where, $\overline{d(i,\cdot)}$ and $Max(d(i,\cdot))$ give the mean and maximum distance between sensor node i and all cluster-heads that sensor node i recognizes through receiving advertisements.

4.4 Comparisons to ANTCLUST

There are several differences between our method and ANTCLUST. As mentioned in 4.1, meetings in sensor networks are one-way and one-to-many. Therefore, only sensor nodes that receive an advertisement adjust their clusters. On the other hand, a sensor node that emits a message does not meet other sensor nodes and cannot learn anything about the recipients. In our method similarity is not defined by the relationship between two sensor nodes but rather the familiarity of a sensor node to the cluster-head of a sender. Since a sensor node sends its sensor data to a cluster-head in cluster-based data gathering, the distance to a cluster-head concerns a sensor node. When sensor data are relayed to the base station from sensor nodes by other sensor nodes by a kind of multi-hop routing mechanism, similarity must be defined in accordance with distance among sensor nodes.

Random meetings are iterated until cluster formation converges in ANTCLUST. On the other hand, the number of meetings per round is limited in our method to reduce energy consumption. As a result, obtained clusters in a round are not necessarily the most energy-efficient. One possible solution is to resume meetings in the following rounds by starting with the same set of cluster-heads. However, due to the data gathering in the preceding round, the residual energies of the cluster-heads have decreased, and some can no longer afford the expensive role of cluster-head anymore. To organize better clusters, some mechanisms for updating the cluster's information and switching cluster-heads are introduced. However, they also consume energy. In this paper, only a limited percentage of sensor nodes initiate newly meetings per round.

There are five conditions to consider in ANTCLUST, but there are only two in our method. The first and second conditions for creating a new cluster in ANTCLUST are not necessary in our method since all sensor nodes belong to clusters before meetings. The third condition for enlarging a cluster also exists in our method, but there are some differences. First, an estimator M_i is increased by 1 in our method instead of employing a parameter α to simplify the algorithm. In our next step we will consider another algorithm for increasing the estimator that leads to more ap-

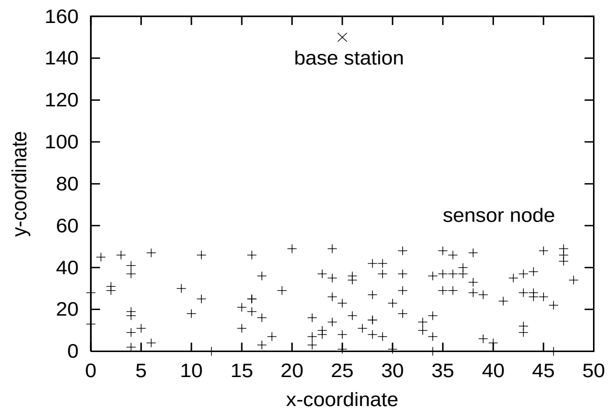

Figure 3. Random 100 node sensor network

propriate cluster re-configuration (Eq. 17). Second, our method does not employ an estimator M^+ that reflects the degree of belonging to a cluster. The estimator becomes effective when two objects in the same cluster do not accept each other in ANTCLUST. In our method, this fourth condition does not hold. A sensor node belongs to a cluster because the distance to a cluster-head is smaller than to its template. In other words, it has already accepted the cluster-head. Thus, two sensor nodes in the same cluster always accept the cluster-head. Consequently, we ignored the estimator M^+. The last condition, where objects belonging to different clusters accept each other, is interpreted in our method as sensor node i finds another cluster whose cluster-head is close enough to sensor node i. Then it chooses a better cluster, based not only on the cluster size M but also on the residual energy and distance.

5 Simulation Experiments

We evaluated the effectiveness of our method through simulation experiments. We considered sensor networks of 100 sensor nodes randomly arranged at lattice points in a 50×50 region. A base station was located at (25, 150). An example of a generated network is shown in Fig. 3. We used two models of initial residual energy of sensor nodes: uniform at 0.5 J, and random from 0.2 J to 0.5 J. We set E_{elec} at 50 (nJ/bit), ε_{amp} at 100 (pJ/bit/m^2), and E_{fuse} at 5 (nJ/bit/message) in equations (1) and (2). The size n of sensor data was 2000 bits. An advertisement message was 60 bits long. In the following figures and tables, average values over 100 simulation experiments are depicted.

Figures 4 through 7 show the number of sensor nodes that remained alive and the number of rounds for different settings of radius R for reporting candidacy from 10 to 70 and the number of sensor nodes from 50 to 400 for the same

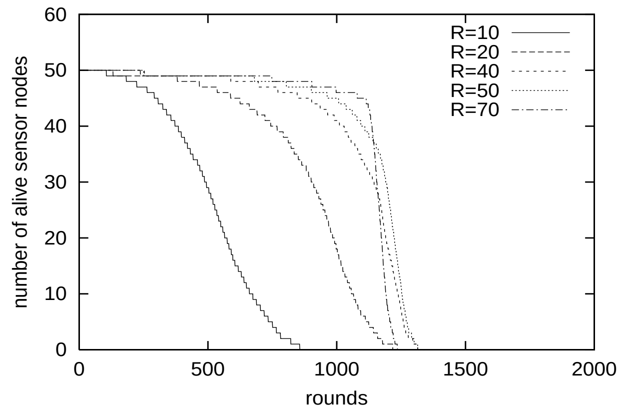

Figure 4. Number of alive sensor nodes (50 nodes, $R = 10 - 70,\ r = 20,\ P_{ex} = 10\%$)

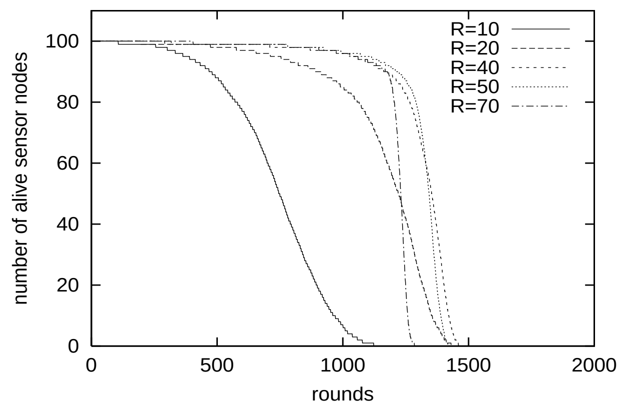

Figure 5. Number of alive sensor nodes (100 nodes, $R = 10 - 70,\ r = 20,\ P_{ex} = 10\%$)

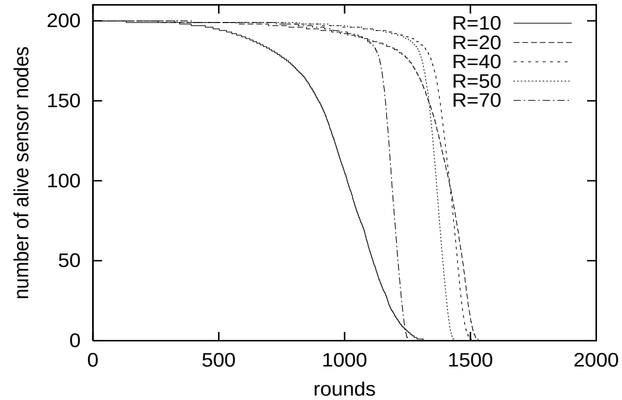

Figure 6. Number of alive sensor nodes (200 nodes, $R = 10 - 70,\ r = 20,\ P_{ex} = 10\%$)

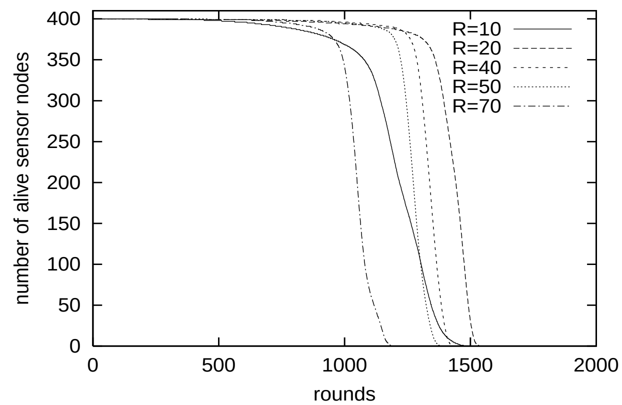

Figure 7. Number of alive sensor nodes (400 nodes, $R = 10 - 70,\ r = 20,\ P_{ex} = 10\%$)

size region. The broadcasting radius r for meetings is set at 20, and the percentage P_{ex} of social sensor nodes is set at 10%. All sensor nodes initially have energy of 0.5 J. These figures indicate that there is a trade-off between radius R and the lifetime of a sensor network. In LEACH, there is trade-off between the lifetime and the number of cluster-heads is. In our method, R is related to the number of resultant clusters. As R increases, the number of clusters decreases. If there are fewer clusters, the number of sensor nodes that become energy-consuming cluster-heads decreases. On the other hand, the number of cluster members increases, it requires more energy to collect sensor data in a cluster. In addition, the diameter of a cluster expands, so cluster members need more energy to send their sensor data to the cluster-heads. When R is small, many clusters are organized into a sensor network. The energy consumption of cluster members is reduced, whereas the number of cluster-heads increases, and much energy is lost.

Table 1. Density of sensor nodes and desired radius

the number of sensor nodes (density)	desired R (the number of clusters)
50 (0.02)	50 (1～2)
100 (0.04)	50 (2～3)
200 (0.08)	40 (3～4)
400 (0.16)	20 (5～8)

Several factors affect the desired radius R, but we expect that each sensor node can determine the appropriate radius R by observing its environment and estimating the density. Table 1 summarizes the desired R against the num-

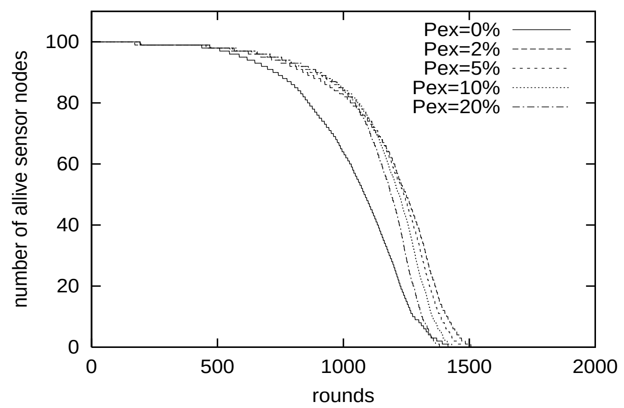

Figure 8. Number of alive sensor nodes (100 nodes, $R = 20,\ r = 20,\ P_{ex} = 0\% - 20\%$)

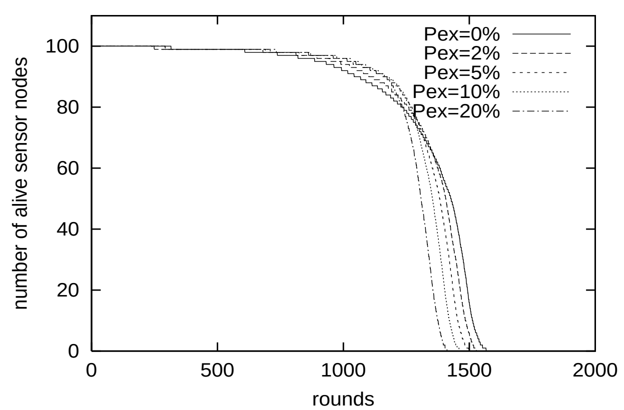

Figure 9. Number of alive sensor nodes (100 nodes, $R = 40,\ r = 20,\ P_{ex} = 0\% - 20\%$)

ber of sensor nodes. To determine the desired R, we focus on the maximum number of rounds which results in more than 80% of sensor nodes remaining alive. It can be seen that a larger radius is preferable to a sparse network, and vice versa. Since a cluster-head which is closer than threshold $Template_i$ is considered by sensor node i, it is efficient and effective to consider $Template$ in deciding the radius R. When a sensor node becomes a cluster-head, it receives registration messages from its cluster members. By dividing the number of members by πR^2, it can easily estimate the density d around itself. One possible way is to adjust the radius R_i for candidacy of sensor node i is to apply $Template_i + f(d)$, where $f(d)$ is a monotonically decreasing function of density d.

Figures 8 and 9 show the results when the percentage P_{ex} of social sensor nodes changes from 0% to 20% for $R = 20$ and 40, respectively. As shown in Fig. 8, when the

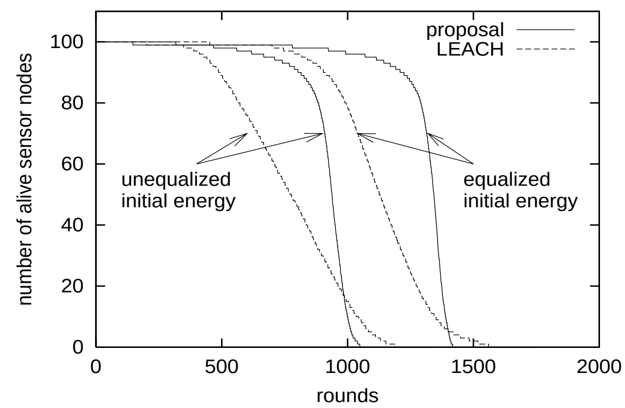

Figure 10. Comparison among proposed method and LEACH

number of meetings increases with a larger percentage P_{ex}, more energy-efficient clusters are organized so that more than 80% of sensor nodes remain alive longer, but increased energy consumption is sacrificed in meetings. On the other hand, in Fig. 9, where a cluster-head advertises its candidacy to larger extent, exchanging cluster information results in a shorter network lifetime of a sensor network. When the radius R for broadcasting candidacy grows, the possibility increases that a sensor node close to a cluster-head decides to advertise for cluster-information. Since the radius r is smaller than R, the broadcast signal does not effectively reach sensor nodes in the other clusters, and so no cluster changes occur.

These result imply that parameters R, r, and P_{ex} have some relationships. In addition, they are dependent on several conditions including the density of nodes, the extent of the region, the model of energy consumption, the residual energy, and so on. We need to consider a scheme where each node dynamically and autonomously determines an appropriate set of control parameters observing its surroundings, but it remains as a future research work.

Figure 10 shows the comparison results between LEACH and our proposed method for cases where the initial residual energy is homogeneous and heterogeneous among sensor nodes. Due to space limitation, only typical results are shown, but the same tendency were observed with other settings. For LEACH, we used the percentage of cluster-heads as 5% [2], with which LEACH shown the best performance in the simulation experiments. For our proposed method, we set the radius R for broadcast of candidacy at 50, the radius r for broadcast of exchanging cluster information at 20, and the percentage P_{ex} of social sensor nodes at 10%. It is shown that, independent of the initial energy condition, the instant that the first sensor node halts

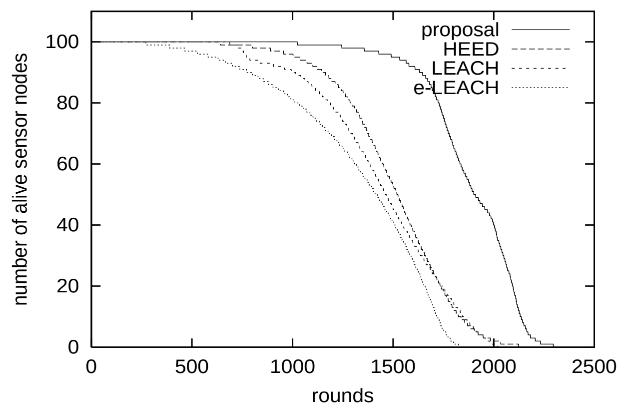

Figure 11. Comparison of the number of alive sensor nodes with the other clustering methods

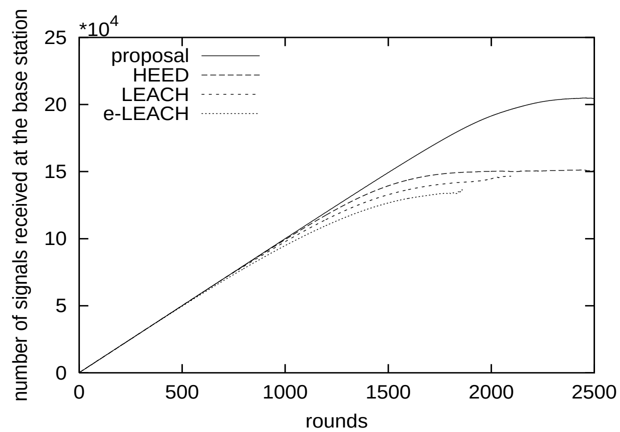

Figure 12. Comparison of the amount of data received at BS with the other clustering methods

for the starvation of battery in our method is earlier than in LEACH. However, in our method, more sensor nodes live longer than in LEACH. For example, the number of rounds in which more than 80% of the sensor nodes keep alive is more than in LEACH by 25% to 55%. In sensor networks, the energy consumption of cluster-heads differs according to the number of cluster members and the distance to a base station. The energy consumption of cluster members also depends on the distance to a cluster-head. However, in LEACH, every sensor has the same chance to become a cluster-head. A sensor node with insufficient residual energy occasionally becomes a cluster-head, even if there is a sensor node with rich battery power nearby. It exhausts its energy, stops operating, and disrupts the gathering of sensor data in its cluster. On the other hand, in our proposal sensor nodes with more residual energy become cluster-heads. Sensor nodes with less residual energy conserve energy. As a result, the residual energy is well equalized among sensor nodes, and the lifetime of a sensor network is prolonged.

In Fig. 11, we show the comparison results to other clustering methods, i.e., a distributed-version of LEACH-C [5], which we called e-LEACH, and HEED [6], which is another distributed and self-organizing clustering scheme. Figure 12 illustrates the results in terms of the total amount of sensor data received at the base station. We employed another model of energy consumption in transmitting a k-bit message to distance d as [5]:

$$E_{T_x}(k,d)=\begin{cases} k\cdot E_{elec}+k\cdot\varepsilon_{fs}\cdot d^2, & \text{if } d<d_0 \\ k\cdot E_{elec}+k\cdot\varepsilon_{mp}\cdot d^4, & \text{if } d\geq d_0, \end{cases} \quad (20)$$

where the threshold d_0 was introduced to take into account the effect of multi-path fading. Conditions of simulation experiments were the same as in [6]. In our proposal, R, r, and P_{ex} were set at 40, 20, and 10%, respectively. It is shown that our proposal outperforms the others. The reason why e-LEACH leads to shorter lifetime of a sensor network than LEACH is that the size of messages becomes longer for sensor nodes to advertise the amount of residual energy.

6 Conclusions and Future Work

In this paper, based on ANTCLUST, we proposed a novel clustering algorithm for energy-efficient data gathering in sensor networks. Sensor nodes with more residual energy became cluster-heads, improving the organization of clusters by local interactions among sensor nodes. Simulation experiments verified that our proposed method prolonged the lifetime of sensor networks as much as 150% of LEACH.

Since we proposed the clustering method as a replacement of the cluster-based candidacy phase and the cluster formation phase of LEACH, it suffers from some impractical assumptions of LEACH. For example, sensor nodes are assumed to change their phases synchronously, but it needs some energy-expensive and complicated synchronization mechanism. In addition, sensor nodes must have a communication device capable of all of CSMA, CDMA, and TDMA. However, since a sensor node is small and has a limited capacity, it cannot afford such a multi-functional device. We plan to remove these assumptions to have a more practical and useful scheme.

In addition, we are now considering a further efficient clustering algorithm where sensor nodes autonomously adjust control parameters after observing its surroundings. We also consider the coverage area of sensor networks for energy-efficient cluster-based data gathering. Furthermore, we extend our method to the case where sensor data are sent to a base station through communications among cluster-heads, i.e., multi-hop transmissions.

Acknowledgements

This work was partly supported by "The 21st Century Center of Excellence Program" and a Grant-in-Aid for Scientific Research (A)(2) 16200003 from The Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

[1] I. F. Akyildiz, W.Su, Y. Sankarasubramaniam, and E.Cyirci, "Wireless sensor networks: a survey," *Computer Networks*, pp. 393–422, March 2002.

[2] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of HICSS-33*, pp. 3005–3014, January 2000.

[3] S. Lindsey, C. Raghavendra, and K. Sivalingam, "Data gathering in sensor networks using the energy*delay metric," in *IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, pp. 924–935, April 2001.

[4] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering-based heuristic for data gathering and aggregation in sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC, 2003)*, March 2003.

[5] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, pp. 660–670, October 2002.

[6] O. Younis and S. Fahmy, "Distributed clustering in adhoc sensor networks: A hybrid, energy-efficient approach," in *Proceedings of IEEE INFOCOM*, March 2004.

[7] E. Bonabeau, G. Theraulaz, and M. Dorigo, *Swarm Intelligence*. New York : Oxford University Press, October 1999.

[8] M. Dorigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," *Future Generation Computer Systems*, vol. 16, pp. 851–871, June 2000.

[9] N. Labroche, N. Monmarché, and G. Venturini, "A new clustering algorithm based on the chemical recognition system of ants," in *Proceedings of ECAI 2002*, pp. 345–349, July 2002.

[10] A. E. Langham and P. W. Grant, "A multilevel k-way partitioning algorithm for finite element meshes using competing ant colonies," in *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 2, pp. 1602–1608, July 1999.

[11] A. E. Langham and P. W. Grant, "Using competing ant colonies to solve k-way partitioning problems with foraging and raiding strategies," in *Proceedings of the 5th European Conference on Artificial Life*, September 1999.

[12] D. L. Hall and J. Llinas, *Handbook of Multisensor Data Fusion*. CRC Pr I Llc, June 2001.

[13] A. Savvides, C. Han, and M. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of Mobicom 2001*, pp. 166–179, July 2001.

[14] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," in *Proceedings of the first ACM international workshop on Wireless Sensor Networks and Applications*, pp. 105–111, September 2002.

[15] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 81–95, September 2003.

[16] R. Yamaoka, "The communication and community of ants," *NATURE INTERFACE*, no. 6, pp. 58–61, 2001.

[17] R. Yamaoka, "How do ants recognize their nestmates?," *Quarterly Journal Biohistory*, no. 23, 1999.