Wang J, Cao YT, Xie JY *et al.* Energy efficient backoff hierarchical clustering algorithms for multi-hop wireless sensor networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 26(2): 283–291 Mar. 2011. DOI 10.1007/s11390-011-1131-x

Energy Efficient Backoff Hierarchical Clustering Algorithms for Multi-Hop Wireless Sensor Networks

Jun Wang[1,2,3] (王　珺), Yong-Tao Cao[4] (曹涌涛), Jun-Yuan Xie[1,2] (谢俊元), *Member, CCF*
and Shi-Fu Chen[1,2] (陈世福)

[1] *State Key Lab for Novel Software Technology at Nanjing University, Nanjing 210093, China*

[2] *Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China*

[3] *Institute of Communication and Information Engineering, Nanjing University of Posts and Telecommunications Nanjing 210003, China*

[4] *China Development Center, Trend Micro Corp., Nanjing 210012, China*

E-mail: wang_jun@njupt.edu.cn; roger_cao@trendmicro.com.cn; {jyxie, chensf}@nju.edu.cn

Received July 8, 2009; revised January 21, 2011.

Abstract Compared with flat routing protocols, clustering is a fundamental performance improvement technique in wireless sensor networks, which can increase network scalability and lifetime. In this paper, we integrate the multi-hop technique with a backoff-based clustering algorithm to organize sensors. By using an adaptive backoff strategy, the algorithm not only realizes load balance among sensor node, but also achieves fairly uniform cluster head distribution across the network. Simulation results also demonstrate our algorithm is more energy-efficient than classical ones. Our algorithm is also easily extended to generate a hierarchy of cluster heads to obtain better network management and energy-efficiency.

Keywords multi-hop wireless sensor network, clustering algorithm, backoff strategy

1 Introduction

In wireless sensor networks (WSN), communication bandwidth and energy are significantly more limited than in a tethered network environment. These constraints require innovative design techniques to use the available bandwidth and energy efficiently[1−3].

Clustering algorithms appeared from ad-hoc networks, which is inspired by wired networks, such as the Internet. LEACH[4] is the first typical clustering protocol designed for sensor networks. Because it is the first mature clustering algorithm, LEACH becomes a baseline for successors. Since the birth of LEACH, more and more clustering algorithms have been proposed and show their advantages in energy efficiency and node management compared with flat routing protocols, such as directed diffusion[5]. HEED[6] is another classical clustering approach proposed in 2004, to achieve better load balance, HEED considers the residual energy as the criterion to choose cluster-heads. On the other hand, HEED is too strict to destroy the randomness of the algorithm, which might lead to worse energy efficiency. In order to realize more load balancing, Cao presents a new adaptive backoff strategy in [7] to not only realize load balance among sensor node, but also ensure that the elected cluster-heads are evenly-distributed.

The above protocols share the commonality that only one-hop clusters are considered. One-hop clusters require any pair of sensor nodes in the same cluster to be able to communicate directly (i.e., the sensor nodes are within each other's transmission range). In large networks, this constraint may lead to a large number of cluster-heads, potentially increasing the efforts of inter-cluster control information flow, which consumes energy as well as network bandwidth. Moreover, even if sensor nodes in the same cluster can communicate directly, [4] proved that multi-hop communication is more energy-efficient than direct communication. In addition, one-hop clusters demand that all sensors are equipped with the capability of tuning the power for the

Short Paper

Supported by the National Natural Science Foundation of China under Grant No. 60872018, 60721002, 60875038, the National Basic Research 973 Program of China under Grant No. 2007CB310607, SRFDP Project under Grant No. 20070293001, the Science and Technology Support Foundation of Jiangsu Province under Grant No. BE2009142 and BE2010180; the Scientific Research Foundation of Graduate School of Nanjing University under Grant No. 2011CL07.

variable-range communication. However, in some cases, sensors are very simple and all the sensors transmit at a fixed power level, so data between two communicating sensors who are not within each other's radio range is forwarded by other sensors in the network.

Hence, it is desirable to find an effective solution for constructing multi-hop clusters, where the distance of sensor nodes is no longer limited to one hop.

The Max-Min d-cluster algorithm proposed in [8] generates d-hop clusters which can achieve better load balance and generate fewer clusters than early algorithms. But this algorithm does not ensure that the total energy consumption is minimized. Moreover, the Max-Min d-cluster algorithm suffers high implement complexity, making it unsuitable for resource-limited sensor networks. Assuming that sensors are distributed according to a homogeneous spatial poisson process, Bandyopadhyay *et al.*[9] first introduced a multi-hop hierarchical clustering algorithm for wireless sensor networks, they prove that the algorithm can work better than previous algorithms in multi-hop networks. But the random property of the algorithm, a lot of "forced cluster-heads" will appear in many cases, i.e., the node has no cluster to join and has to communicate directly with BS, which is often located far away from the node's vicinity. The resulting long-range transmission to the BS is very energy demanding. Also the residual battery energy has not been considered in their algorithm.

2 Energy-Efficient Backoff-Based Clustering Algorithm

Taking into account sensor nodes' residual energy, our proposed algorithm in multi-hop sensor networks can achieve better dynamic load balance than [9]. It also achieves fairly uniform cluster head distribution across the network. Moreover, it does not need any prior knowledge about the topology information. Compared with the Max-Min d-hop algorithm[8], the clustering process in our algorithm terminates in $O(1)$ iterations and incurs low implementation complexity. Although we borrow the backoff idea of [7], the proposed algorithm in our paper is totally new because the backoff strategy and the whole algorithm execution process have been re-designed so that it can work well in much complicated multi-hop networks. Both theoretical analysis and simulation experiments show the proposed algorithm outperforms the previous ones.

2.1 Algorithm Description

The operation of our algorithm is divided into rounds. Similar to other round-based clustering algorithms[4,6-7,9], each round begins with a set-up phase, and followed by steady-state phase, when the data is sensed and transferred from nodes to cluster-heads, finally reaches the base station (BS). The steady-state phase is much longer than set-up phase in most cases. In such round-based clustering algorithms, clock synchronization is inevitable. Because nodes need clock synchronization to not only start and end each round, but also determine the time when data is sensed. The nodes must all be time-synchronized in order to start the set-up phase at the same time. Now many mature clock synchronization methods are used in sensor networks[10].

The parameters used in this algorithm are listed in Table 1. All nodes are equipped with the same battery, which has an initial energy $E_{\max}$. And each node knows its residual battery energy, which can be assured by its circuit system[6]. T_{CF} is the maximum clustering-forming time, when T_{CF} elapses, the whole networks will enter the steady-state phase. The transmission range of all messages is limited to k-hop to avoid "message explosion". Also each node has a unique node ID.

Table 1. Parameters Used in the Algorithm

E^i_{res}	the estimated current residual energy in the node i
$E_{\max}$	the fully charged battery energy
T_{CF}	the maximal cluster-forming time
k	number of hops

In the set-up stage, all nodes are initially in the waiting mode. Each node i waits for a random amount of t_i according to the following equation before making an attempt to be the cluster-head:

$$t_i = -\frac{1}{\lambda_i}\ln(1 - x_i) \qquad (1)$$

where x_i is a random variable uniformly distributed over the interval [0, 1]. That is, t_i is a random variable whose probability density function is $f(t_i) = \lambda_i e^{-\lambda_i t_i}$, so it only depends on λ_i. Here we set $\lambda_i = \alpha\frac{E^i_{res}}{E_{\max}}$, where α is a constant. How to set the value of α will be discussed in the Subsection 2.2.

Although the computing method of the waiting time t_i seems a little strange, it ensures that nodes with higher residual energy are more likely to be triggered earlier and become the cluster-head within a k-hop neighborhood, which will be proved in implication of Property 1 and Property 2 in Subsection 2.2.

When node i's timer fires, that means node i has not received any messages from other nodes in the time period t_i, node i elects itself as a cluster-head and broadcasts an *ADV_CH* message, which is a triplet of (*CH_ID*, *RN_ID*, *TTL*). *CH_ID* and *RN_ID* represent cluster-head ID and relay-node ID, and *TTL* is a time-to-live value for the message. Here node i will broadcast an $ADV\_CH(i, i, k)$ message to its neighbors. Different from the one-hop-cluster formation in [4, 7], the advertisement is forwarded to all the sensors that are no more than k hops away from the cluster-head, where k is a system parameter set in the configuration.

Upon receiving an $ADV\_CH(m, n, x)$ message, node j will stop its timer and become a subordinate node. It records the message information, including the cluster-head ID m, the relay-node ID n and TTL x. Then it decreases TTL of the message by one, if x is greater than zero, it will forward $ADV\_CH(m, n, x-1)$ in a broadcasting approach, otherwise the message will be dropped. If one node simultaneously receives more than one ADV_CH message, i.e., it falls within the range of more than one self-elected cluster-heads, it records all messages which it received and generates a simple routing table to store the cluster-head ID, relay-node ID and its hop to the cluster-head, that is $k - x + 1$. Such intermediate nodes will become a relay-node just like the router in the Internet. In single-hop sensor networks, all subordinate nodes will never keep any routing information.

For a self-elected cluster-head, if it receives an ADV_CH message from other nodes, it will not forward this message.

When T_{CF}, the maximal time of set-up elapses, a subordinate node j needs to decide to join which cluster. It will firstly check its distance to those candidate cluster-heads from its routing table. It will choose the nearest cluster-heads. If the distance between the node and two different cluster-heads are the same, the node choose the cluster-head whose cluster ID is smaller. Then the subordinate node will broadcast a JOIN message, which is also a triplet of $(CH\_ID, RN\_ID, myID)$. Upon receiving a JOIN message, a subordinate node checks whether it is the relay node. If it is the case, it checks its routing table and finds the next-hop relay node and generates a new JOIN message with the new relay-node. Otherwise it will drop the message.

If a cluster-head receives such a JOIN message, it checks if the destination cluster-head in the message is itself. If it is the case, it records the subordinate's decision; otherwise it will drop the message.

When T_{CF} elapses, a node which has not sent or received any ADV_CH message will become a "forced cluster-head" during this round[4]. That is, any sensor that is neither a cluster-head nor has joined any cluster will become a cluster-head itself; we call these cluster-heads the forced cluster-heads, and they have to directly communicate with the remote base station. It is well known that radio communication with low-lying antennas and near-ground channels has an exponential path loss, i.e., the minimum output power required to transmit a signal over a distance d is proportional to d^n, $2 \leqslant n \leqslant 4$. Since transmitting data directly to the remote base station is very energy demanding, the clustering algorithm should prevent too many nodes from involving in such long-distance communications. In the following subsection, we will discuss how to decrease the number of forced cluster-heads. After every t units of time, the cluster-head transmits the aggregated information to the processing center.

Then the whole networks complete the cluster-formation phase and enter the steady-stage phase. Another set-up phase will begin until the next round.

2.2 Properties of the Algorithm

Through our analysis, we find our algorithm has the following properties.

Property 1. *For any two nodes* 1 *and* 2, $Pr[t_1 < t_2] = \frac{\lambda_1}{\lambda_1+\lambda_2}$.

Proof.

$$Pr[t_1 < t_2] = \int_0^\infty \lambda_1 e^{-\lambda_1 t_1} \int_{t_1}^\infty \lambda_2 e^{-\lambda_2 t_2} \mathrm{d}t_2 \mathrm{d}t_1$$
$$= \int_0^\infty \lambda_1 e^{-(\lambda_1+\lambda_2)t_1} \mathrm{d}t_1 = \frac{\lambda_1}{\lambda_1+\lambda_2}. \quad \square$$

Implication of Property 1. Since we set $\lambda_i = \alpha E^i_{res}/E_{\max}$, we have $Pr[t_1 < t_2] = \frac{E_1}{E_1+E_2}$. In other words, the algorithm ensures that the node with more residual energy is more likely to become a cluster-head since its timer is more likely to trigger earlier. In contrast, Bandyopadhyay's algorithm disregards nodes' residual battery energy. Therefore, our cluster-head selection algorithm is likely to perform better dynamic load-balancing for wireless sensor networks.

Property 2. *Consider nodes* $1, 2, \ldots, n$ *with* $\lambda_1, \lambda_2, \ldots, \lambda_n$ *respectively.*

$$Pr[t_i < t_k \forall k \neq i] = \frac{\lambda_i}{\sum_k \lambda_k}.$$

Proof. Similar to the proof of Property 1. $\square$

Implication of Property 2. Different nodes have different numbers of nodes, n, as its k-hop neighbors. Property 2, after plugging in $\lambda_i = \alpha E^i_{res}/E_{\max}$, simply says that the chance of a node becoming a cluster-head is proportional to its own residue energy divided by the "total energies" within k hops.

We also hope that the network elects enough cluster-heads in one round before the time T_{CF} elapses: for instance at least 50% of nodes are expected to initialize cluster-formation within one minute. The reason is that when too few cluster-heads are elected, it is very likely that there is no self-elected cluster-head in some node's proximity. So the node has to become a "forced cluster-head" and communicates directly with

BS, which is often located far away from the node's vicinity. Thus the selection of λ should satisfy the following inequation: $p\{t < T_{CF}\} \geqslant \sigma$, for some target percentage σ. In this case

$$\int_0^{T_{CF}} \lambda e^{-\lambda t} \mathrm{d}t \geqslant \sigma \Rightarrow \lambda \geqslant \frac{-\ln(1-\sigma)}{T_{CF}}. \quad (2)$$

Based on (2), we calculate that a λ of 0.013 86 is needed to ensure that 50% of the nodes will initialize the cluster-forming process within one minute ($\sigma = 0.5$, $T_{CF} = 60\,\mathrm{s}$).

Property 3. *For any two nodes* 1 *and* 2, *$Pr[|t_1 - t_2| < T_c] = 1 - \frac{\lambda_1 e^{-\lambda_2 T_c} + \lambda_2 e^{-\lambda_1 T_c}}{\lambda_1+\lambda_2}$, where T_c is the maximum node-to-node delay between two k-hop-away sensor nodes.*

Proof.

$$\begin{aligned}
&Pr[|t_1 - t_2| < T_c] \\
&= \int_0^{\infty} \lambda_1 e^{-\lambda_1 t_1} \int_{t_1 - T_c}^{t_1+T_c} \lambda_2 e^{-\lambda_2 t_2} \mathrm{d}t_2 \mathrm{d}t_1 - \\
&\quad \int_0^{T_c} \lambda_1 e^{-\lambda_1 t_1} \int_{t_1 - T_c}^{0} \lambda_2 e^{-\lambda_2 t_2} \mathrm{d}t_2 \mathrm{d}t_1 \\
&= \frac{\lambda_1}{\lambda_1+\lambda_2}(e^{\lambda_2 T_c} - e^{-\lambda_2 T_c}) - \\
&\quad \frac{\lambda_1}{\lambda_1+\lambda_2}(e^{\lambda_2 T_c} - e^{-\lambda_1 T_c}) + 1 - e^{-\lambda_1 T_c} \\
&= 1 - \frac{\lambda_1 e^{-\lambda_2 T_c} + \lambda_2 e^{-\lambda_1 T_c}}{\lambda_1+\lambda_2}. \qquad \square
\end{aligned}$$

Implication of Property 3. Substituting $\lambda_i = \alpha E^i_{res}/E_{\max}$, we get

$$\begin{aligned}
&Pr[|t_1 - t_2| < T_c] \\
&= 1 - \frac{E^1_{res} e^{-\alpha E^2_{res} T_c/E_{\max}} + E^2_{res} e^{-\alpha E^1_{res} T_c/E_{\max}}}{E^1_{res} + E^2_{res}}.
\end{aligned}$$

To ensure $Pr[|t_1 - t_2| < T_c] \leqslant \varepsilon$, we need

$$\begin{aligned}
f(E^1_{res}, E^2_{res}) &\triangleq \frac{E^1_{res} e^{-\alpha E^2_{res} T_c/E_{\max}} + E^2_{res} e^{-\alpha E^1_{res} T_c/E_{\max}}}{E^1_{res} + E^2_{res}} \\
&\geqslant 1 - \varepsilon. \qquad (3)
\end{aligned}$$

The function $f(E^1_{res}, E^2_{res})$ will reach its minimum when $E^1_{res} = E^2_{res} = E_{\max}$. (The proof will not be presented here because of page limit.)

So we have

$$\begin{aligned}
f(E^1_{res}, E^2_{res}) &= \frac{E^1_{res} e^{-\alpha E^2_{res} T_c/E_{\max}} + E^2_{res} e^{-\alpha E^1_{res} T_c/E_{\max}}}{E^1_{res} + E^2_{res}} \\
&\leqslant f_{\min}(E^1_{res}, E^2_{res}) = f(E_{\max}, E_{\max}) = e^{-\alpha T_c}.
\end{aligned}$$

Thus, (3) is satisfied if

$$f_{\min}(E^1_{res}, E^2_{res}) = e^{-\alpha T_c} \geqslant 1 - \varepsilon. \qquad (4)$$

By choosing an appropriate α, i.e., $\alpha \leqslant -\frac{\ln(1-\varepsilon)}{T_c}$, we should be able to bound $\Pr[|t_1 - t_2| < T_c]$. The proposed algorithm is able to ensure that the probability that two nodes within each other's cluster range are both cluster-heads is small, i.e., cluster heads are well scattered. For example, when $\varepsilon = 0.01$ and $T_c = 0.001$ (1 ms), choosing $\alpha \leqslant 10$ can satisfy the above inequation.

Property 3 implies that elected cluster heads are well scattered, namely, the volunteer cluster-heads are evenly distributed in the working region. They will not clump in one region so that most of ordinary nodes can find one cluster-head in their neighborhood. As a result, compared with Bandyopadhyay's algorithm, the chance that a sensor node becomes a "forced cluster-head" will be greatly decreased.

We can conclude that in comparison with Bandyopadhyay's algorithm the proposed algorithm is a promising clustering approach to use for extending system life because it is able to not only perform dynamic load-balancing, but also effectively decrease the number of "forced cluster-heads".

2.3 Simulation Experiments and Results

We conduct simulation experiments to evaluate the performance of the proposed algorithm. The entire simulation is conducted in a 100 m × 100 m region, which is between ($x = 0$, $y = 0$) and ($x = 100$, $y = 100$). 100 nodes with 2 Joule initial energy are randomly spread in this region. The node-to-node transmission range (r) and node-to-BS transmission range are set to 15 meters and 200 meters respectively. Initially, each node is assigned a unique node ID and x, y coordinates within the region. The base station locates in the (50, 175). The maximum number of hops k is set to 2.

We firstly try to find out the optimal α based on system life measured in simulation experiments. Similar to [7], assuming $\sigma = 0.5$, $T_{CF} = 60$ seconds, $\varepsilon = 0.01$ and $T_c = 0.001$, we choose $1 \leqslant \alpha \leqslant 10$ to satisfy (2) and (4). Fig.1 shows how the choice of α impacts system life (working rounds). The analytical optimal value of α will be deduced in our future work. In this paper, we choose $\alpha = 4$ in the following simulation experiments.

Simulation experiments proceed with rounds. In each round, one ordinary node, if it has enough residual energy to function properly, collects sensor data and sends a packet (packet size $L = 10000$ bit) to its CH or BS. Similar to [4], we also define "system life" as the time (the working rounds in the paper) until the first node dies.

Fig.2 and Fig.3 are the output of one of the simulations of the Bandyopadhyay's algorithm and our

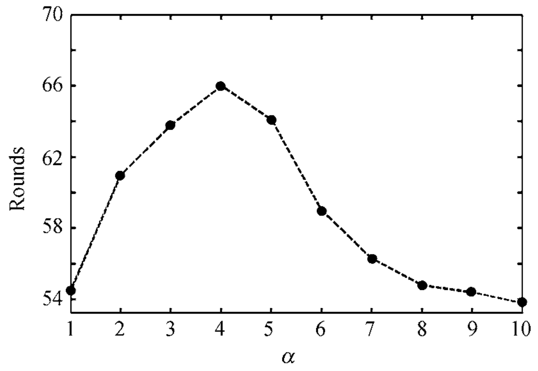

Fig.1. System life vs. α.

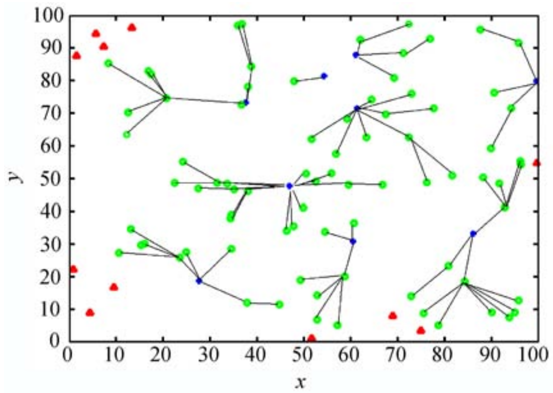

Fig.2. Simulation result of Bandyopadhyay's algorithm.

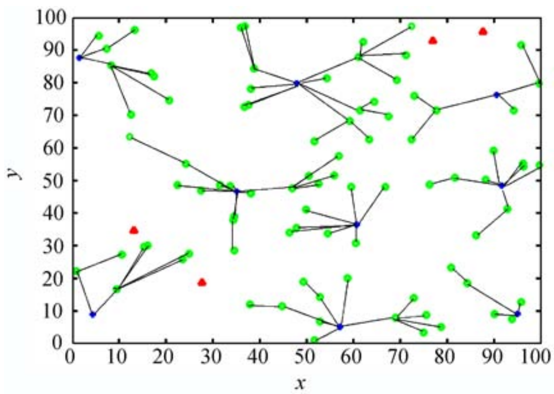

Fig.3. Simulation result of the proposed algorithm.

proposed algorithm. For Bandyopadhyay's algorithm, we set the optimal value $p = 0.05$. Note here we set $k = 2$ and $r = 15\,\text{m}$, i.e., two-hop clustering networks are formed in the simulation. In Fig.2 and Fig.3 the "plus" and "circle" represent "cluster-head" and "ordinary node" respectively while "triangle" represents "forced cluster-head". These figures show that Bandyopadhyay's algorithm and our approach both elect 11 cluster-heads. However, the number of "forced cluster-heads" is quite different: only 4 "forced cluster-heads" appear in the simulation of the proposed algorithm while the number is 11 in Bandyopadhyay's algorithm's operation. This is because cluster-heads elected by our algorithm are well-scattered.

Next, we measure the system life for three clustering protocols: Bandyopadhyay's algorithm, HEED and our algorithm, where system life is the time until the first node dies. HEED[9] can be extended for multi-hop sensor networks. For HEED, we set $p_{\min}$ to 0.0005 and CH_{prob} to 5%. As mentioned in Section 1, each node in HEED's operation must distribute its own cost, which is energy-consuming. Fig.4 illustrates our algorithm outperforms Bandyopadhyay's algorithm and HEED in terms of system life.

Then we study the relationship between system life and effective sensor data. Observing the simulation results of Fig.5. We can see that our algorithm will produce more effective sensor data than Bandyopadhyay's algorithm and HEED over time since our algorithm has effectively reduced extra energy consumption.

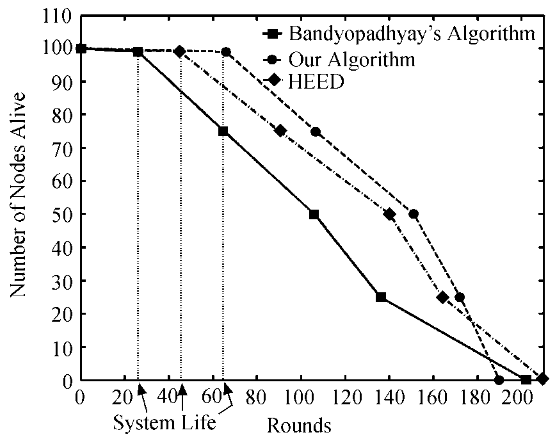

Fig.4. System life using different clustering algorithms.

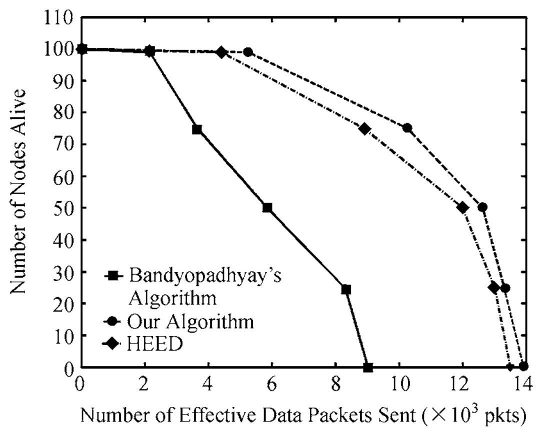

Fig.5. Number of survival nodes per given amount of effective data packets sent.

3 Backoff-Based Hierarchical Clustering Algorithm

Multi-level hierarchical technique is an efficient tool to organize large-scale networks such as Internet or cellular networks. Imaging large-scale networks, such as Internet or GSM networks, hierarchy is the only choice for network management. The authors in [9, 11] also mentioned that multiple levels of clustering might decrease system energy consumption in wireless sensor networks, although they did not give any detail plan. Similarly, although the authors in [7] propose a back-off algorithm, they did not try to extend the method to organize a multi-level networks. Our algorithm is inspired by the above research work. We expect that such a technology is able to reduce the total energy consumption. Meanwhile, it also can serve for better network management. In this section we extend the algorithm introduced in Subsection 2.1 with the multi-level hierarchical technique.

In an h-level clustering hierarchy, there are h levels in the clustering hierarchy with level 1 being the lowest level and level h being the highest. In the multi-level clustering hierarchy, the total energy is the energy spent by the sensors to communicate the information to level-1 cluster-heads (CHs), plus the energy spent by the level-1 CHs to communicate the aggregated information to level-2 CHs, plus the energy spent by the level-h CHs to communicate the aggregated information to the processing center[7].

3.1 Algorithm Description

The hierarchical clustering algorithm works in a bottom-up fashion. Taking the advantage of the algorithm described in Subsection 2.1, the level-1 cluster-heads has been elected from the whole network. Then level-2 cluster-heads are elected from level-1 cluster-heads, and so on until the highest level cluster-heads have been elected.

In the level j $(j = 2, \dots, h)$, the i-th cluster-head of level-$(j-1)$ waits for a random amount of t_i^j according to the following equation before making an attempt to be the level-j's cluster-head:

$$t_i^j = -\frac{1}{\lambda_i^j}\ln(1 - x_i^j) \tag{5}$$

where x_i^j is a random variable uniformly distributed over the interval [0, 1]. That is, t_i^j is a random variable whose probability density function is $f(t_j^i) = \lambda_j^i e^{-\lambda_j^i t_j^i}$. Here we set $\lambda_i^j = \alpha_j \frac{E_{res}^i}{E_{\max}}$, where α_j is a constant related to level j. Generally for simplicity, we just set the same expected ratio for each level, which means $\alpha_1 = \alpha_2 = \cdots = \alpha_n = \alpha$. So we have a simple computation method for λ_i^j, i.e., $\lambda_i^j = \alpha\frac{E_{res}^i}{E_{\max}}$, which only depends on node i's residual energy. As described in Subsections 2.2 and 2.3, the analytical optimal value of α is fairly difficult, so the experimental method is recommended to compute the optimal α. We also choose $\alpha = 4$ in the following simulation experiments.

Each level-$(j-1)$ cluster-head (CH), which is to become a level-j's CH, advertises itself as a level-j cluster-head. This advertisement is forwarded to all the sensors within k_j hops of the advertising CH. Similar to level-1 cluster formation phase, here level-$(j-1)$ CH, node i will broadcast an $ADV\_CH(i, i, k_j)$ message to its neighbors.

Any node, which is either a level-$(j-1)$'s CH or an ordinary sensor, forwards the packets according to the method which is introduced in Subsection 2.1. For example, upon receiving an $ADV\_CH(m, n, x)$ message, node j will stop its timer if it is a level-$(j-1)$ CH. Then it records the message information, including the cluster-head ID m, the relay-node ID n and TTL x. Then it decreases TTL of the message by one, if x is greater than zero, it will forward $ADV\_CH(m, n, x-1)$ in a broadcasting approach, otherwise the message will be dropped. If one node simultaneously receives more than one ADV_CH message, i.e., it falls within the range of more than one self-elected cluster-heads, it will record all received messages and generate a simple routing table to store the cluster-head ID, relay-node ID and its hop to the cluster-head, that is $k - x + 1$.

When T_{CF} elapses, each level-$(j-1)$ CH which received advertisements chooses to join the cluster of the closest level-j CH; the remaining CHs will become forced level-1 CHs. In such a way, the level-j cluster-heads are elected from level-$(j-1)$ CHs, and so on until the highest level cluster-heads have been elected. Note that any level-i CH is also a CH of level $(i-1)$, $(i-2), \dots, 1$. The detailed algorithm description could be found in Subsection 2.1.

The optimal value of k_j can be computed from Section 4 in [9]. It is fairly complicated, so for lower implementation complexity, we can simply set $k_h = 2k_{h-1} = \cdots = 2^{h-2}k_2 = 2^{h-1}k_1$ to try to keep network connectivity, i.e., when $k_1 = 2$, we have $k_2 = 4, \dots, k_j = 2^{j-1}$.

The multi-level hierarchical protocol indeed increases inner-cluster communications compared with single-level clustering algorithm, so it is best applied to those applications where the base station (BS) is far from the sensor networks, such as an unmanned airplane or a satellite acts as the BS to collect data from sensors deployed in the desert. The cost of inner-cluster communication will become negligible in such cases.

3.2 Optimal Parameters for the Algorithm

For simplicity, our analysis in this subsection assumes a fixed mode in which $\lambda^j = \alpha_j$, which means the residual energy would be considered in this analysis. Sure an adaptive mode would further improve the results of our system, so the performance results of the fixed mode can serve as a conservative bound for those of the adaptive mode. In this subsection, we will discuss how many cluster-heads should be deployed in each level in the hierarchy, i.e., the tradeoff between redundancy and performance should be considered. More cluster-heads produced in the system will improve the performance by diminishing so-called "forced cluster-heads", but inevitably increase the total cost.

To limit our scope, we will only consider the non-adaptive mode in our analysis here. Specifically, we would like to determine the optimal backoff constant α_j $(j = 1, \ldots, h)$ that minimizes the total energy consumed. Note that α_j also relates to the optimal number of cluster-heads in the system. We use the same analytical process in [11].

For simplicity, we assume that the sensors are uniformly distributed in a circular region A of radius a meters with the sink located at the center of the circle. The analysis can be easily extended to accommodate other shapes and sink locations. Sensors send packets to their respective cluster-heads, using multi-hop paths (if necessary). Each hop in these paths is roughly of characteristic distance d_{char}[4]. That is, each node forwards the data to a node that is approximately d_{char} closer to the destination.

The analytical process is the same with [11]. So we omit most analytical process here. During each cycle, sensors collect data. The data generated is then sent to the cluster-heads in a packet of r bits. Each cluster-head compresses the data it receives from the sensors of its cluster and then forwards the data to the sink. We denote the number of cluster-heads at level-j by n_j $(j = 0, 1, \ldots, h)$. Note that $n_0 = n$. The data is sent out of a level-j node to its cluster-head at a rate of r_j bits/cycle and $r_o = r$. Here we use analytical process used in [11] to strengthen the validation of our algorithm.

Let E_{aj} be the total energy consumed by all of the aggregators of each level j in a single cycle for the compression process. According to [11]'s analysis, we can get:

$$E_{aj} = n_j \times f_a\Big(\frac{n_{j-1}}{n_j} \times r_{j-1}\Big)$$

where $f_a(x) = \mu x$, μ for some constant.

We now consider E_{cj}, the total energy consumed by sending data from level-j cluster-heads to level-$(j+1)$ cluster-heads in a single cycle. From [11]'s analytical process, we have $E_{cj} = \frac{2\beta a n_j r_j}{3 n_{j+1}^{1/2}}$. Thus, the total energy consumed in a single cycle is

$$\sum_{j=1}^{h} E_{aj} + \sum_{j=0}^{h} E_{cj} \tag{6}$$

which is a function of the n_j[11]. Given values of r, a, β and data compression payload for a particular system, we can calculate the values of n_j to minimize the above total energy consumption.

Then we set $p_j = n_j/n_{j-1}$ $(j = 1, \ldots, h)$. From (5), we have $\int_0^{T_{CF}} \lambda^j e^{-\lambda^j t} \mathrm{d}t = p_j$ (T_{CF} is the cluster-formation time in each level), we can get the optimal backoff constant as follows:

$$\lambda^j = \alpha_j = \frac{-\ln(1 - p_j)}{T_{CF}}.$$

These values can then be used to configure the proposed protocol. The total cluster-formation time will be hT_{CF}.

3.3 Simulation Experiments

We use the algorithm to generate a clustering hierarchy with different numbers of levels in it to see how the energy spent in the network conserves with the increase in number of levels of clusters. Most of the parameters in the experiment is the same with Subsection 2.3. Different from Subsection 2.3, the location of the base station is far from the networks because hierarchical clustering networks are best applied to those applications where an unmanned airplane or a satellite will act as the base station as we mentioned in Subsection 3.1. The base station locates in the (50, 10000).

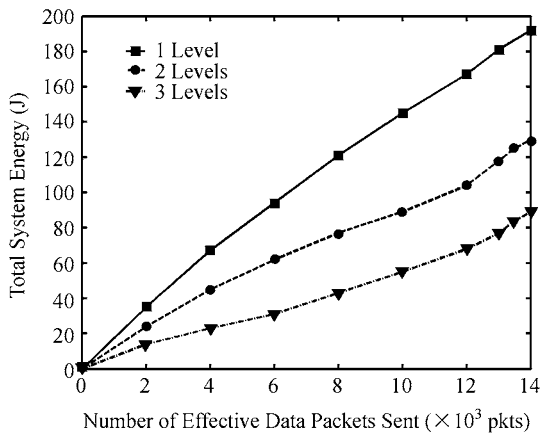

Fig.6. Energy consumptions vs. amount of effective data packets sent.

The simulation experiments are conducted with $h = 1, 2, 3$ respectively, and we set $k_1 = 2$, $k_2 = 4$ and $k_3 = 8$

for three levels. We also set $T_{CF} = 60$ seconds and $\alpha = 4$ in the experiments.

Firstly we compare the energy-efficiency in the networks with different number of levels. Fig.6 shows the total number of effective sensor data sent by network nodes to the base station for a given amount of energy. From the results, we can conclude that as the number of levels in the hierarchy increases, more effective sensor data for a given amount of energy are sent, i.e., the increase of levels can improve the energy efficiency.

Next, we measure the system life for the networks with different numbers of levels. Fig.7 illustrates as the number of levels in the hierarchy increases, the hierarchical networks have better performance than single-level clustering networks in terms of system life because the energy-consuming long-range communication has been greatly decreased in the hierarchical sensor networks.

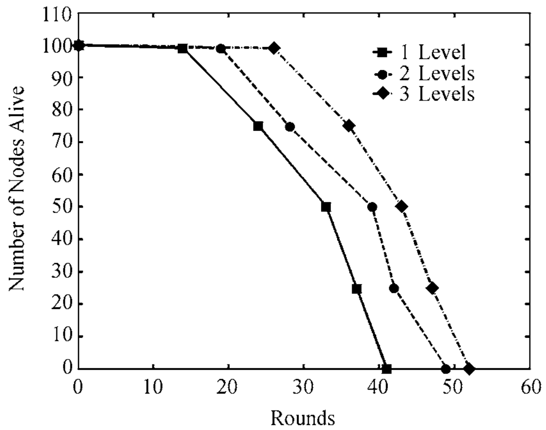

Fig.7. System life in different numbers of levels of hierarchy.

4 Conclusion

In this paper, we have proposed a distributed algorithm for organizing sensors into a hierarchy of clusters with the objective to spread this energy usage over all nodes and improve the system life in the wireless sensor networks. We use the multi-hop technique in the intra-cluster communications in order to save energy. The algorithm not only uses an adaptive backoff strategy to realize load balance among sensor node, but also ensures that the elected cluster-heads are well-distributed. The desirable properties of our algorithm include even distribution of energy load among sensor nodes, implementation simplicity and $O(1)$ time complexity. Simulation results also indicate that our scheme reduces the number of "forced cluster-heads" substantially and prolongs the system life by nearly 30% compared with previously proposed schemes.

In this paper, it is assumed that the communication environment provided by underlying MAC protocol is ideal; in the future we intend to consider an underlying medium access protocol and investigate how that would affect the optimal probabilities of becoming a cluster-head and the run-time of the algorithm.

References

[1] Akyildiz I F, Su W, Sankarasubramaniam Y *et al.* A survey on sensor networks. *IEEE Communications Magazine*, 2002, 40(8): 102-114.

[2] Zhao F, Guibas L. Wireless Sensor Networks: An Information Processing Approach. Morgan Kaufmann, 2004.

[3] Pottie G J, Kaiser W J. Wireless integrated network sensors. *Communications of the ACM*, 2000, 43(5): 51-58.

[4] Heinzelman W B, Chandrakasan A P, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Tran. Wireless Communications*, Oct. 2002, 1(4): 660-670.

[5] Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. ACM/IEEE Int. Conf. Mobile Computing and Networking* (*MOBICOM*), Boston, USA, Aug. 6-11, 2000, pp.56-67.

[6] Younis O, Fahmy S. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 7-11, 2004.

[7] Cao Y, He C. A distributed clustering algorithm with an adaptive backoff strategy for wireless sensor networks. *IEICE Transactions on Communications.*, 2006, 89-B(2): 609-613.

[8] Amis A D, Prakash R, Vuong T H P, Huynh D T. Max-Min *D*-cluster formation in wireless ad hoc networks. In *Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 26, 2000, pp.32-41.

[9] Bandyopadhyay S, Coyle E J. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proc. IEEE INFOCOM 2003*, San Francisco, USA, Mar. 30-Apr. 3, April 2003, pp.1713-1723.

[10] Sundararaman B, Buy U, Kshemkalyani A D. Clock synchronization for wireless sensor networks: A survey. *Ad Hoc Networks*, 2005, 3(3): 281-323.

[11] Chen Y P, Liestman A L, Liu J. A hierarchical energy-efficient framework for data aggregation in wireless sensor networks. *IEEE Trans. VT*, May, 2006, 55(3): 789-796.

[12] Yu M, Leung K K, Malvankar A. A dynamic clustering and energy efficient routing technique for sensor networks. *IEEE Transactions on Wireless Communications*, Aug. 2007, 6(8): 3069-3079.

Jun Wang received the B.E. and M.E. degrees in electronic engineering from the P.L.A. University of Science and Technology, Nanjing, China. Now she is a Ph.D. candidate in computer science in Nanjing University. She is also an associate professor in the Nanjing University of Posts and Telecommunications. Her research interests include wireless sensor networks and broadband networking technology.

Yong-Tao Cao received the B.E. and M.E. degrees in electronic engineering from P.L.A. University of Science and Technology, Nanjing, China, and Ph.D. degree from Shanghai Jiaotong University. He works in Trend Micro Corp.

Jun-Yuan Xie is a professor in Department of Computer Science and Technology at Nanjing University. He is a member of CCF. His research interests include network security and artificial intelligent.

Shi-Fu Chen is a professor in Department of Computer Science and Technology at Nanjing University. His research interests include network security and artificial intelligent.