An Energy-Efficient Balanced Clustering Algorithm for Wireless Sensor Networks

Zhigao Du, Yi Liu, Depei Qian
Sino-German Joint Software Institute, School of Computer Science
Beihang University
Beijing, China
zhigao.du@jsi.buaa.edu.cn

***Abstract*—Clustering is a popular topology control method in wireless sensor networks, which can facilitate the network self-management and make it easy to devise the communication protocols. Also clustering can improve energy efficiency and the network scalability. Existing clustering algorithms concern much about the local energy consumption, but little about the overall energy consumption. A novel energy-efficient, balanced clustering algorithm EEBC is proposed in this paper. In EEBC the sensor nodes are clustered randomly at first, and then they conduct self-adaptive optimization to balance the size of clusters. The structure of the cluster is fixed after the optimization. The operation of EEBC is divided into rounds. At the end of each round the current cluster head selects a node from its cluster members as the next cluster head. The process of the cluster head rotation is transparent to other cluster members. The results of simulations show that EEBC outperforms existing algorithms in energy efficiency and clustering balance.**

Keywords-wireless sensor networks; balanced clustering; energy-efficient; self-adaptive iteration; cluster head rotation

I. INTRODUCTION *(HEADING 1)*

Grouping nodes into clusters is widely considered as an efficient topology control method in wireless sensor networks(WSNs). Every cluster has one leader which is often reffered to as cluster head(CH) and many followers known as cluster members(CM). Clustering results in a two-tier hierarchy in which CHs form the higher tier while CMs form the lower tier. In a typical cluster-based WSN, the cluster member nodes report their data to the respective CHs directly, and the CHs aggregate the data and send them to the sink nodes via multi-hop. As the scope of inter-cluster communication is limited to CHs in cluster-based WSNs, CMs have less work to do and they can turn off their communication modules most time. CHs' data aggregation can reduce the data transported in network, which helps to conserve communication bandwidth and battery energy. Also a CH can manage and coordinate its CMs' activities to further enhance the network efficiency and prolong the network lifetime.

Many clustering algorithms have been proposed for WSNs. In LEACH[1] a sensor node selects itself as CH with a probability ρ, which reflects the expected amount of CHs in network and whether this node has played the role of CH recently. HEED[2] selects CHs from sensor nodes according to a combination of their residual energy and intra-cluster communication cost. In EEDC[4] a sensor node triggers cluster structure update only when it finds that significant change has taken place in local area. Clustering is regarded as a set division problem in MCCP[5]. GRIDS[6] is an on-demand protocol, which constructs cluster architecture only when there are on-going data packets. CPCP[7] and UCR[8] try to find a coherent way to solve clustering and other problems in WSNs, specifically coverage and hot spot problem.

Although much attention has been paid on clustering issue, the existing algorithms still have some defects. Most existing clustering algorithms put the local energy consumption on first position, but concern little about the balance of clusters. Unbalanced cluster structure will cause unbalanced energy consumption and shorten the network lifetime. Another problem of most algorithms is the cost of cluster head rotation. In order to distribute the energy consumption of CH in WSN, network operation is divided into rounds in most algorithms and CH role is played by different nodes in each round. Cluster formation process is conducted at the beginning of each round to select new CHs. This process wastes time and crucial battery energy.

In this paper we propose a novel energy-efficient, balanced clustering algorithm called EEBC. Our main contributions include:

- After clustering randomly firstly, the nodes conduct self-adaptive iterations to balance the size of clusters.
- The structure of the cluster is fixed after the iterations. At the end of each round current CHs choose CHs in next round according to the residual energy. The process of the cluster head rotation is transparent to other CMs.

The rest of this paper is organized as follow. Section Ⅱ defines the clustering problem. The details of EEBC are described in Section Ⅲ, followed by experiment results in Section Ⅳ. Section Ⅴ concludes the paper.

This research is supported by National Science Foundation of China through the grant 90412011, 60673180 and 90612004.

II. PROBLEM DESCRIPTION

This paper tries to address the clustering problem in large-scale dense WSNs. Some assumptions and symbols used in the paper are presented as follow.

- All nodes are homogeneous and have the same capability(processing/communication/battery capacity).
- Sensor nodes are immobile after deployment.
- Nodes are location-unaware and there is no clock synchronization among nodes.
- We denote the set of nodes in network as V. The total number of nodes is denoted as N=|V|.
- The cluster structure is denoted as set $U=\{u_1,u_2\ldots u_k\}$, where u_i represents the ith cluster.
- According to LEACH protocol, there exists an optimum size of cluster for best energy efficiency. We denote the optimum interval of cluster size as $[N_{opt\_min}, N_{opt\_max}]$.

Definition 1. If $|u_i|\in[N_{opt\_min}, N_{opt\_max}]$, u_i is called a *standard cluster*. If $|u_i|<N_{opt\_min}$,, u_i is called a *small cluster*. If $|u_i|> N_{opt\_max}$,, u_i is called a *large cluster*.

We denote the number of standard clusters in network as α, the aim of clustering algorithm can be described as follow.

$$\forall i, j, u_i \cap u_j = \Phi \tag{1}$$

$$\sum_{i=1}^{k} |u_i| = N \tag{2}$$

$$\text{Maximize } \frac{\alpha}{k} \tag{3}$$

$$\mathit{Minimize} \sum_{u=1}^{k} (|u_i| - \frac{N}{k})^2 \tag{4}$$

Equation (1) describes the clustering is integrated. Equation (2) defines that cluster overlap is not allowed. Formula (3) and (4) define the balance of clustering. Of course energy-efficiency is also important for clustering algorithm.

III. EEBC ALGORITHM DESIGN

EEBC can be divided into three phases: random clustering, self-adaptive optimization and cluster head rotation.

A. random clustering

After network startup, the sensor nodes construct cluster structure randomly firstly.

1) CH selection: Each sensor node waits for a random time between (0, T_r) after startup. During this time if it receives *CH announcement* messages from neighboring nodes, it gives up chance to become a CH. If *CH announcement* message is not received, it decides whether to become a CH with a probability P_h, which is the expected ratio of CHs in network.

2) Cluster formation: When a sensor node receives a *CH announcement* message, it calculates the distance from the CH accoring to free space channel mode[3] and the RSSI(received signal strength indicator). If the distance is less than $R/2$, where R denotes the sensor node's communication range, it adds the CH in its cluster head candidates list. At time T_c, sensor node chooses the one with highest RSSI in the list as its CH and sends *Join* message to the CH.

3) Finalization: If a sensor node is neither a CH nor a CM when random clustering is over at time T_l, it becomes a CH.

We limit the distance between CH and its CM less than $R/2$ during the random clustering phase and self-adaptive optimization phase, so that any two nodes in a cluster can communicate with each other directly and new CH can reach all CMs directly after cluster head rotation. As shown in figure 1, node M is a CH after self-adaptive optimization. If node A becomes the new CH after CH rotation, then any CM node B satisfies equation (5).

$$AB < AM + BM < R/2 + R/2 = R \tag{5}$$

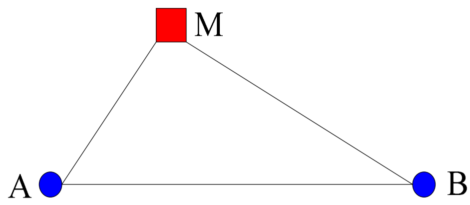


Figure 1. Any two nodes in a cluster can reach each other

B. Self-adaptive optimization

After random clustering we try to make the size of every cluster approach the optimum value through selft-adaptive iterations. In each iteration sensor node gathers information of neighboring clusters and adapts its own membership to optimize the cluster structure.

Definition 2. If $|u_i|<N_{abd\_thred}$, then u_i is called a *tiny cluster* and should be canceled. N_{abd_thred} is predefined threshold and satisfies $N_{abd\_thred}<N_{opt\_min}$.

1) Cluster size broadcast: At the beginning of each iteration, CH broadcasts control message to announce its cluster size. Each node maintains a size list of local clusters.

2) Cluster head adaption: If a cluster head finds its cluster is a *tiny cluster*, it announces to abandon the cluster and become an ordinary node.

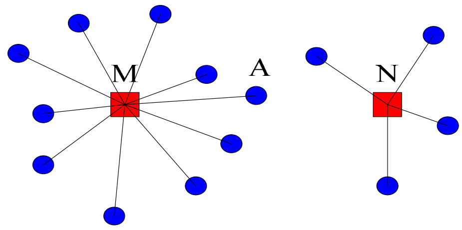
3) Ordinary node adaption: ordinary nodes decide how to adapt its cluster membership based on its size list.

a) Ordinary node in a large cluster: If there exists *small clusters* in its size list, it will choose one from them randomly and migrate to it with a probability P_t. If the probability P_t is not used, too many nodes may migrate from a *large cluster* to a neighboring *small cluster.* As a result, the two cluster switch their roles, and this may cause migration oscillation in followed iterations. If there is no *small cluster* in its list, the node is in the intersection region of some *large clusters*. It waits for a random delay and becomes a CH with a probability P_l, and recruits memebers from neighboring *large clusters*.

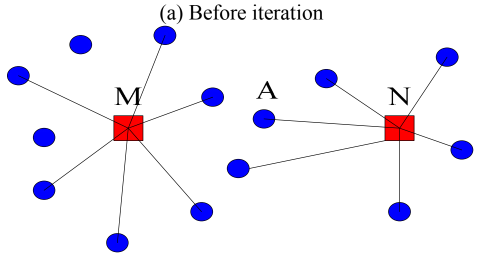
b) Ordinary node in a standard cluster: The nodes in *standard clusters* do nothing in iteration.

c) Ordinary node in a small cluster: If there exists *large clusters* in its size list, the node does nothing but waiting for nodes in *large clusters* to migrate to its cluster. If there is no *large cluster* in its list, then the node is in the insection region of some *small clusters*. It waits for a random delay and becomes a CH with a probability P_s, and tries to merge local *small clusters*.

d) Ordinary node in a tiny cluster: Its cluster has been canceled. The node chooses a neighboring *small cluster* randomly and joins it. If there is no *small cluster* in its size list, it waits for a random delay and becomes a CH with a probability P_a to construct a new cluster.



(a) Before iteration



(b) After iteration

Figure 2. Node A migrates from cluster M to cluster N

C. Cluster head rotation

After self-adaptive optimization, the network enters working phase. During this time the cluster count and the nodes' membership keep fixed. As CHs do more work than CMs, they will consume more battery energy. In order to avoid that CHs die earlier than CMs, energy consumption of cluster heads should be distributed among all nodes. Network operation is divided into rounds, and in a cluster CH role is played by all members in turn round by round.

At the beginning of each round, CH starts a timer that expires when the round is over. During network operation, CM piggybacks its residual energy to its CH. At the end of the round, CH nominates the member with most residual energy in its cluster as next CH. Then the two nodes swap their addresses to carry out CH rotation.

Because periodic clustering is avoided and CH rotation is transparent to ordinary nodes, it is time-efficient and energy-efficient. As the time span of round is controlled by CH, the clock drift will result in time difference between clusters. But it will not affect the network function.

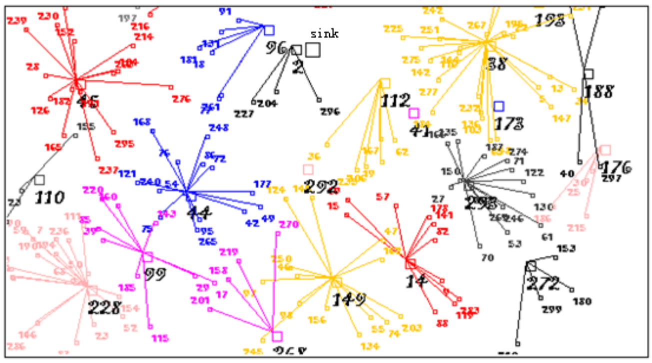
IV. Simulation Results

To evaluate the performance of EEBC protocol, we used NS-2[9] simulator to conduct experiments. Here, the effect of self-adaptive optimization is validated. Also we compared EEBC with LEACH and HEED in clustering balance and control message cost.

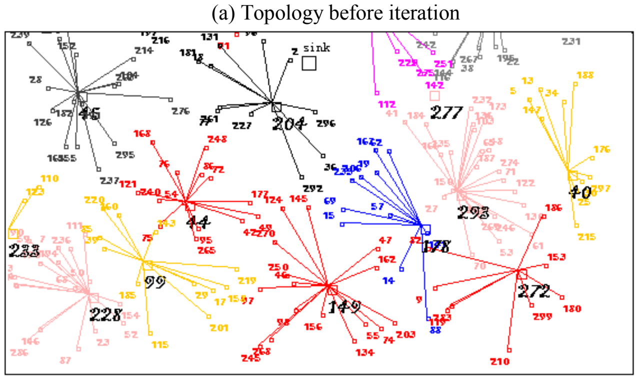
The example scenario consists of 300 nodes randomly deployed in the field of 150*150m^2 with the sink node at the center of the field. The communication range of nodes is 50m. 802.11 is used as MAC protocol. All results presented here are averaged over 20 runs.

A. The effect of self-adaptive optimization

Network topology graphs before and after self-adaptive optimization are presented in figure 3. It shows that the self-adaptive optimization can improve the clustering balance. We use standard deviation of clusters' CM count to evaluate the clustering balance. Figure 4 shows that the standard deviation will decrease when more iterations are conducted. And the optimization reduces the standard deviation at most 7.4 percent.



(a) Topology before iteration



(b) Topology after iteration

Figure 3. The effect of self-adaptive optimization

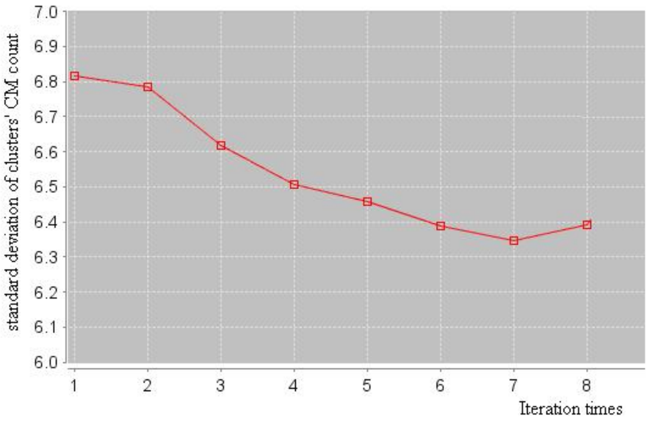


Figure 4. Iteration times vs clustering balance

B. *The clustering balance*

We compared the clustering results of EEBC, HEED and LEACH. In EEBC, the optimum interval of cluster size is set [13, 17], and the *tiny cluster* threshold N_{abd_thred} is set 5. In LEACH, the ratio between CMs and CH is set 16. The results of EEBC presented in figure 5 are based on 8 self-adaptive iterations, while results of HEED are based on 5 iterations.

It is shown in figure 5 that EEBC improves the clustering balance compared with LEACH and HEED. It reduces the standard deviation of CM count by 19.6 percent compared with LEACH, and 23.8 percent compared with HEED.

Figure 5 shows that the cluster size is less than we expect in all three algorithms. It is because that there exist some isolated nodes during clustering, which decrease the average cluster size.

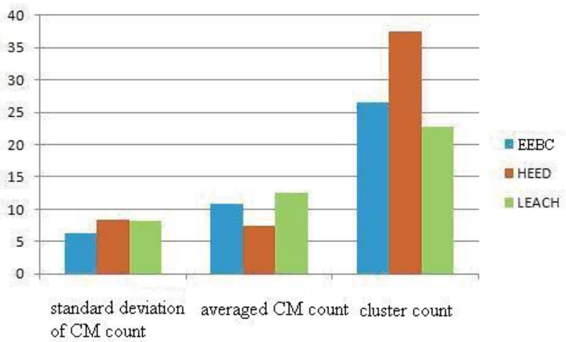


Figure 5. Clustering results

C. *The control message cost*

The clustering control message cost of EEBC, HEED and LEACH is shown in figure 6. It shows that with the increase of round count there is a linear increase in control message cost of HEED and LEACH. It is because that in the two algorithms the clustering process is repeated in each round. As some clustering iterations have to be done in HEED, its control message cost is higher than LEACH.

In EEBC because the self-adaptive optimization is executed in the first round, the message cost of EEBC in the first round is much higher than HEED and LEACH. But CH rotation in EEBC is energy-efficient. From figure 6 we can see the clustering cost of HEED and LEACH surpasses EEBC from the 5th and 20th round. Considering the lifetime of a WSN, EEBC is much more energy-efficient than HEED and LEACH.

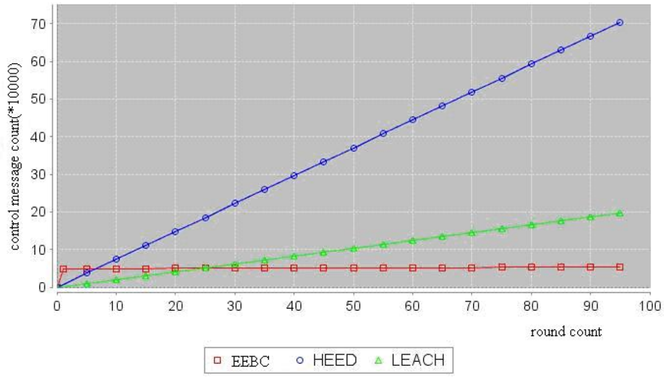


Figure 6. The control message cost

V. CONCLUSION

In this paper we propose a novel distributed clustering algorithm. After randomly clustering, the CM counts of clusters are optimized through self-adaptive optimization. The cluster structure maintain stable during network operation. In order to reduce the energy and time cost, periodic clustering is avoided and CH rotation is designed to be transparent to CMs. The experiment results show that EEBC is more balanced and energy-efficient than existing algorithms.

REFERENCES

[1] W.B. Heinzelman, A.P. Chandrakasan and H.Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE transactions on wireless communications, 2002,1(4): 660~670.

[2] Younis O and Fahmy S, "Heed: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks," IEEE Trans. on Mobile Computing, 2004,3(4):660−669.

[3] T. Ruppe, "Wireless Communications: Principles & Practice," Englewood Cliffs, NJ: Prentice-Hall, 1996.

[4] Ming Yu, Kin K. Leung and Aniket Malvankar, "A dynamic clustering and energy efficient routing technique for sensor networks," IEEE transactions on wireless communications, 2007, 6[8]: 3069~3079.

[5] Pu Wang, Cheng Li and Jun Zheng, "Distributed minimum-cost clustering protocol for underwater sensor networks(UWSNs)," in Proc. Of ICC. Glasgow: IEEE Society. 2007: 3510~3515.

[6] D. El Ghanami, T.J. Kwon and A. Hafid, "GRIDS: geographically repulsive insomnious distributed sensors- An efficient node selection mechanism using passive clustering," in Proc. Of IEEE international conference on wireless and mobile computing, networking and communicaiton. Washington DC: IEEE Society. 2008: 241~246.

[7] Stanislava Soro and Wndi B. Heinzelman, "Cluster head election techniques for coverage preservation in wireless sensor networks," Ad hoc Networks, 2008, In Press, Corrected Proof.

[8] Guihai Chen, Chengfa Li, Mao Ye and Jie Wu, "An unequal cluster-based routing protocol in wireless sensor networks," Wireless Networks, 2007, 42(2): 183~198.

[9] UCB/LBNL/VINT Network Simulator [Online]. Available: http://www.isi.edu/nsnam/ns/.