This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

An Energy-Aware Protocol for Data Gathering Applications in Wireless Sensor Networks

Ming Liu[1], Yuan Zheng[2], Jiannong Cao[2], Guihai Chen[3], Lijun Chen[3],Haigang Gong[1]

[1] School of Computer Science and Engineering, University of Electronic and Technology of China, China
Email: wing.lm@gmail.com

[2] Internet and Mobile Computing Lab, Department of Computing, Hong Kong Polytechnic University, China
Email: csyuazheng@comp.polyu.edu.hk, csjcao@comp.polyu.edu.hk

[3] State Key Laboratory for Novel Software Technology, Nanjing University, China
Email: gchen@nju.edu.cn, chenlj@nju.edu.cn

***Abstract*—Data gathering is a major function of many applications in wireless sensor networks (WSNs). The most important issue in designing a data gathering algorithm is how to save energy of sensor nodes while meeting the requirement of applications/users such as sensing area coverage. In this paper, we propose a novel hierarchical clustering protocol for long-lived sensor network. EAP achieves a good performance in terms of lifetime by minimizing energy consumption for in-network communications and balancing the energy load among all nodes. EAP introduces a new clustering parameter for cluster head election, which can better handle the heterogeneous energy capacities. Furthermore, it also introduces a simple but efficient approach, namely intra-cluster coverage to cope with the area coverage problem. We evaluate the performance of the proposed protocol using a simple temperature sensing application. Simulation results show that our protocol significantly outperforms LEACH and HEED in terms of network lifetime and the amount of data gathered.**

I. INTRODUCTION

With the advances in technologies of micro electro-mechanical systems, embedding system technology and wireless communication with low power consumption, it is now possible to produce micro wireless sensors for sensing, wireless communication and information processing. Through the cooperation of sensor nodes, the WSNs collect and send various kinds of monitored environment information (e.g. temperature, humidity, etc.) to the sink node, which processes the information and reports to the user. Wireless sensor networks have a wide-range of applications, including military surveillance, disaster prediction, and environment monitoring, and thus have attracted a lot of attention from researchers in the fields of military, industry and academy.

In wireless sensor networks, the resource of sensor node is limited in terms of processing capability, wireless bandwidth, battery power and storage space, which distinguishes wireless sensor networks from traditional ad hoc networks [14]. In most applications, each sensor node is usually powered by battery and expected to work for several months to one year without recharging. Such expectation cannot be achieved without carefully scheduling the energy utilization, especially when sensors are deployed densely (up to 20 nodes/m^3 [1]), with severe problems such as scalability, redundancy, and radio channel contention. Due to high density, multiple nodes may generate and transmit redundant data for the same event to the sink node, causing unnecessary energy consumption and hence significantly decreasing network lifetime. For a sensor node, energy consumption includes three parts: data sensing, data processing, and data transmission/reception, amongst which, the energy consumed for communication is the most critical. Reducing the number of communication by eliminating or aggregating redundant sensed data and using the energy-saving link would save large amount of energy, and then prolong the lifetime of the WSNs.

Data gathering is a typical operation in many applications of WSNs, where data aggregation in a hierarchical manner is widely used for prolonging network lifetime. Data aggregation can eliminate data redundancy and reduce the communication load. Hierarchical mechanisms are helpful to reduce data latency and increase network scalability, which has been extensively exploited [2-8]. In this paper, we propose a distributed and energy-efficient protocol, called EAP for data gathering in WSNs. A node with high ratio of residual energy to the average residual energy of all neighbors within its cluster range will have a large probability of becoming the cluster head. This can better handle heterogeneous energy circumstances than existed clustering algorithms which elect the cluster head only based on node's residual energy. After the cluster formation phase, EAP constructs a routing tree over the set of cluster heads. Only the root node of this tree can communicate with the sink node by single-hop communication. Also, EAP utilizes a simple but efficient approach to solve the area coverage problem. With the increase in node density, using this approach, the network lifetime can be made linear in the number of deployed nodes.

The remainder of this paper is organized as follows: section 2 reviews related works. Section 3 describes the system model and the motivation of our work. Section 4 presents the detailed design of EAP. Section 5 reports the result of EAP effectiveness and performance via simulations and compares it with LEACH and HEED. Section 6 concludes the paper

II. RELATED WORK

The main task of sensor network is to forward the sensing data gathered by sensor nodes to the base station. One simple approach to the fulfillment of this task is direct data transmission. In this case, each node in the network directly sends sensing data to the base station. However, if the base station is remote from the sensor node, the node will soon die for suffering excessive energy consumption for delivering

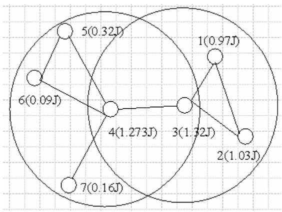

Fig. 1. Example for cluster heads election

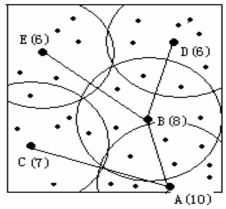

Fig. 2. Routing tree construction

data. To solve this problem, some algorithms that are aimed to save energy have been proposed one after another [3]-[7].

Heinzelman et al. [3] proposed an alternative clustering-based algorithm, called LEACH (Low-Energy Adaptive Clustering Hierarchy). In order to save energy, LEACH only chooses a fraction p of the all sensor nodes to serve as cluster heads, where p is a design parameter that must be determined before deployment. The rest sensor nodes join the proper cluster according to the signal strength from cluster heads. In order to share the energy load, its operation is divided into rounds, which can guarantee the cluster head rotate in each round.

S. Lindsey et al. proposed an algorithm related to LEACH, Called PEGASIS [4]. In order to reduce the energy consumption of sensor nodes, PEGASIS uses GREED algorithm to form all the sensor nodes in the system into a chain. According to its simulation results, the performance of PEGASIS is better than LEACH especially when the distance between sensor network and sink node is far away.

In [5], to deal with the heterogenous energy circumstance, the node with higher energy should have larger probability of becoming the cluster head. In this paper, each node must have an estimate of the total energy of all nodes in the network for compute the probability for becoming a cluster head. As a result, each node can not make a decision of becoming a cluster head only by its local information, so the scalability of this protocol will be influenced.

Sh.Lee et al. proposed a new clustering algorithm CODA [6] in order to relieve the unbalance of energy depletion caused by different distance from the sink. CODA divide the whole network into a few group based on the distance from the base station and the strategy of routing. Each group has its own number of cluster number and member node. CODA differentiates the number of cluster in terms of the distance from the base station. The farther the distance from the base station, the more clusters are formed in case of single hop with clustering. It shows better performance than applying the same probability to the whole network in terms of the network lifetime and the dissipated energy. However, the work of CODA relies on global information of node position, and thus it is not scalable.

In [8], the authors proposed a hybrid, energy-efficient, distributed clustering algorithm which periodically selects cluster head according to a hybrid of the node residual energy and a secondary parameter such as node proximity to its neighbors or node degree. Heed terminates in O(1) iterations and incurs low message overhead. It achieves fairly uniform cluster head distribution across the network.

Besides these clustering algorithms mentioned above, there exist several others algorithms such as [10], [11]. ACE cluster the network in a constant number of iterations using the node degree as the main parameter. Soro et al. Reference [10] proposed an unequal clustering size model for network organization, which can lead to more uniform energy dissipation among cluster head nodes, thus increasing network lifetime. Ye et al. Reference [11] proposed a clustering algorithm which achieves good cluster head distribution with no iteration and introduced a weighted function for the plain node to make a decision of joining a proper cluster.

III. MOTIVATIONS

Data gathering application is a typical application in wireless sensor networks. Our motivation is to study the problem in this kind application. In this section, we will make some assumptions about the network model before motivation.

A. Network Model

This paper assumes that N sensor nodes are randomly scattered in a two-dimensional square field A, and the sensor network has the following properties:

- This network is a static densely deployed network. It means a large number of sensor nodes are densely deployed in a two-dimensional geographic space, forming a network and these nodes do not move any more after deployment.
- There exists only one base station, which is deployed at a fixed place outside A.
- The energy of sensor nodes cannot be recharged.
- Sensor nodes are location-unaware, i.e. sensor node cannot get its location information through other mechanism such as GPS or position algorithms.
- The radio power can be controlled, i.e., a node can vary its transmission power depending on the distance to the receiver [5]. For instance, Berkeley Motes [12] have in total 100 power levels.

B. Motivations

In a real case, it is hard to guarantee that the battery capacity of all nodes is same. The amount of energy consumed for gathering data differs among cluster heads, depending on the number of cluster members and their positions in the monitored area. Energy consumption also differs among cluster members due to the different distance to a cluster-head. Furthermore, redeployment for prolonging network lifetime or denser observing will also cause the problem that residual energy is not equal among all sensor nodes.

In the current body of research done in the area of data gathering protocol in wireless sensor networks, we see that the selection of cluster heads of most existing cluster algorithms mainly depend on the residual energy of node. However, we argue that setting the residual energy as the primary parameter for cluster heads election doesn't help balance the energy load to the proper nodes, especially in heavy energy heterogeneous circumstance. In most local clustering algorithms, to prolong the sensor network lifetime, the probability of a sensor node's being selected as a cluster head primarily depend on its residual energy. However, in some special case, such strategy doesn't help balance the energy load to the proper nodes. As a result, it may cause the problem that some nodes will be exhausted quickly. For instance, as shown in Fig 1, there is a sensor network composed by seven nodes. Node 4 and node 3 locate in the cluster range of each other and the amount of residual energy of node 4 and node 3 is higher than the other nodes. Then, assume that each node's being selected as a cluster head only depends on the its residual energy. Obviously, the probability that node 3 is selected as a cluster head is highest. Consequently, the probability that the other nodes with lower residual energy are selected as a cluster head will increase, like node 5 or node 6. Because the energy consumed as a cluster head is more significant than consumed as a plain node, the energy of nodes within the cluster range of node 4 will be exhausted quickly.

In order to solve the problems mentioned above, we present a novel hierarchical clustering scheme EAP. In the next section, we will describe the EAP algorithm in details.

IV. EAP Protocol Design

Since cluster heads consume more energy than cluster members in receiving sensing data from their member nodes and performing signal processing functions on the data (e.g., data aggregation), and sending the aggregated data to the next hop node or base station, the role of the cluster head must be rotated among all sensor nodes. Therefore, EAP works in rounds as LEACH. Each round begins with a set-up phase while clusters are organized and the routing tree is constructed, followed by a working phase when data are sent to the sink node. For easy reference, we describe the states of nodes and control message in Table I.

TABLE I
DESCRIPTIONS OF STATES AND MESSAGES

State or Message	Description [a]
Candidate	The node is a candidate node
Head	The node is selected as cluster head
Plain	The node is a member node
Compete_Msg	Composed by the ID of sender
Join_Msg	Composed by the ID of sender and the ID of head
Weight_Msg	Composed by the ID of sender, weight, Ea, Eresidual
Schedule_Msg	Head assign slot time for its member nodes

A. Cluster Formation

In EAP protocol, each node needs to maintain a neighborhood table to store the information about its neighbors, as shown in table II. The ID indicates the unique identification of the neighbor nodes. Without losing the generality, we use an integer value to label a node's identification like TinyOS [13]. At the beginning of each round, each node broadcasts the E_Msg within radio range r and all nodes are cluster head candidates. Here we use r to denote the cluster range. All nodes within the cluster range of one node can be seen as the neighbors of this node. Each node receives the E_Msg from all neighbors in its cluster range, which includes the senders' ID and their residual energy, and updates the neighborhood table. Using Ea to denote the average residual energy of all the neighbors within the cluster range, and V_j represents any node in this cluster range, where m is the number of the nodes within the cluster range. We define

TABLE II
NODE 4' S NEIGHBORHOOD TABLE

ID	State	Residual Energy (J)[a]
3	Candidate	1.32J
7	Candidate	0.16J
6	Candidate	0.09J
5	Candidate	0.32J

$$E_a = \frac{\sum_{j=1}^{m} V_j . E_{residual}}{m} \tag{1}$$

After exchanging E_Msg, each node computes the broadcasting delay time t for competing for a cluster head according to the following equation.

$$t = k * T * \frac{E_a}{E_{residual}} \tag{2}$$

where k is a real value uniformly distributed between 0 and 1 and T is the time duration for cluster heads election.

In order to solve heterogeneous energy problem, EAP uses $\frac{E_a}{E_{residual}}$ as the primary clustering parameter for competing cluster heads. Observing equation 2, t is the time that each node broadcasts the Compete_Msg for competing cluster head, which is mainly determined by $\frac{E_a}{E_{residual}}$. As shown in Fig 1., we introduce this clustering parameter for cluster heads election. It is easy to find that the probability of the node 1 being a cluster head will increase. It means that the lifetime of the nodes with low residual energy within the cluster range of node 1 will increase. Compared with the previous works, which only depend on the residual energy of node [5][8][11], EAP can better handle the heterogeneous energy circumstance.

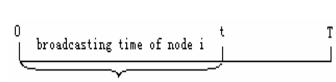

Fig. 3. Illustration for broadcasting Compete_Msg

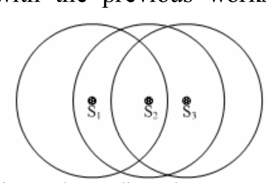

Fig.4. The gradient phenomenon

In EAP, if a node S_i has not received any Compete_Msg from its neighbor nodes during time duration (0, t) as shown in

Fig 3, this node will broadcast the Compete_Msg to its all neighbor nodes. Otherwise, it will give up competition. After S_i broadcasts Compete_Msg, this node will wait 2*Δt to make sure whether there exist other Compete_Msgs broadcasted by other nodes in its cluster range, where Δt denotes the time interval which can guarantee that all neighbor nodes can receive the Compete_Msg. If S_i has not received any Compete_Msg from its neighbors over Δt , it will set its state as Head, or else it will compare its weight with the weights of other broadcasting neighbors. If S_i's weight is the largest one, it will set its state as Head and other broadcasting neighbors give up competition, or else S_i set its state as Plain. Obviously, the procedure CF allows only one cluster head in a cluster range. If there are multiple Compete_Msg overheard, the one with largest weight will serve as the only cluster head.

```
state ← candidate
broadcast Node_Residual_Msg to all neighbor nodes
receive Node_Residual_Msg from all neighbor nodes
update neighborhood table NT []
t ← computation result of the broadcast delay time for competing a cluster head
while (the timer for cluster head election is not expired)
  if(CurrentTime <t)
    if (a Compete_Msg is overheard from a neighbor NT [i])
       state ← plain
       NT [i].state = head
    else
       continue
    endif
  else
    if (state = candidate)
      state ← head
      broadcast Compete_Msg
      wait (2*Δt )
      if (have not received any Compete_Msg)
        continue
      else
      if (the weight for head election is the largest one)
        continue
      else
        state = plain
        if (the value in weight broadcasted by NT [i] is the largest one)
          NT [i].state = head
        endif
      endif
    endif
```

Fig. 5. Cluster Formation (CF) algorithm

It is worthy to notice that sometimes there may exist a gradient phenomenon as shown in Fig 5, where $S_1.weight > S_2.weight > S_3.weight$. Consequently, S2 and S3 will give up competition and S1 is the only winner. In this case, after clustering phase, some nodes will be neither cluster heads nor member nodes. However, because the time interval Δt is short, the probability that several nodes within the same cluster range broadcasting the Compete_Msg in the same time interval (2*Δt) is considerably small. In addition, through expanding time duration T or decreasing the cluster radius, we can guarantee that only one head in a cluster range and each node will either be a cluster head or a plain node whp.

In order to minimize the energy consumption in each round, we argue that the plain nodes should join the nearest head. Because the cluster heads always keep rotation in whole lifespan of network, we can maintain uniform energy consumption among all nodes. So minimizing energy consumption for each round can help to prolong the network lifetime. The pseudo code for cluster formation is shown in Fig 5.

B. Selection of Active Member Nodes

Coverage is one of the most important issues in WSNs and has been studied in recent years [14]-[16]. In most case, "coverage" means area coverage. And K-coverage can be descried as that every point in the monitored field is covered by at least K sensors. In [14], authors think it is hard to guarantee full coverage for a given randomly deployment area even if all sensors are on-duty. Small sensing holes are not likely to influence the effectiveness of sensor networks and are acceptable for most application scenarios. It's enough to meet the application's requirements if the active nodes in the network could maintain reasonable area coverage—coverage expectation. Coverage mechanism is to choose a subset of active nodes to maintain the coverage expectation.

We introduce this idea into clusters that is called "intra-cluster coverage", which selects some active nodes within clusters while maintaining coverage expectation of the cluster. Based on our previous work [17], cluster head randomly chooses m' nodes according to equation (3).

$$p_{cover} = \sum_{i=K}^{m'} C_{m'}^{i} \left(\frac{r}{R}\right)^{2i} \left(1-\frac{r^{2}}{R^{2}}\right)^{m'-i} \qquad (3),$$

where P_{cover} is the coverage expectation of sensing field determined by specific applications; and r is sensing radius, R is cluster radius; m' is the number of active nodes. For example, distributing 200 nodes in a $100\times100m^2$ field, r = 12m, R = 30m, then the average number of cluster members is 60 or so. With intra-cluster coverage, if P_{cover} = 99% which means 99% of sensing field is expected to be monitored, 27 members should be active in each cluster to ensure 1-coverage of the cluster and 38 members to ensure 2-coverage. If P_{cover} =95%, only 16 nodes and 25nodes should be active to ensure 1-coverage and 2-coverage respectively.

Using intra-cluster coverage has three advantages. The first is to save energy in each round by turning redundant nodes' radio off so that network lifetime is prolonged. The second is to reduce TDMA schedule overhead. Once clusters

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

grouped, all cluster heads broadcast a TDMA schedule packet which contains the members ' ID and slot number allocated to the member. When node density is high, the number of cluster members turns larger so that the length of TDMA schedule packet turns longer, which consumes more energy to transmit and receive. However, the length of TDMA schedule packet would not too long with intra-cluster coverage because the number of active node varies slightly when node density goes higher. Apparently, through intra-cluster coverage, EAP can function as a topology control protocol but does not pay any extra energy cost.

C. Construction of Routing Tree

After the network is clustered, inter-cluster organization depends on the network application. For example, cluster heads can communicate with each other to aggregate their information via multiple hops or communicate with base station directly. For multi hop communication among cluster heads, the selected transmission range among cluster heads may vary to ensure a certain degree of connectivity and to control interference. For inter-cluster communication, the definition of connectivity depends on its multi hop organization and the relationship between the inter-cluster transmission range, R, the intra-cluster transmission range, r, and the density of nodes. In [8], authors demonstrate that the graph composed by cluster heads will be connected if $R \geq 6r$. However, we argue that the theoretical value for connectivity may be not applicable for a real application, i.e. the unreasonable inter-cluster range for inter-cluster communication is another inefficient use of energy. For example, we consider a typical setting of sensor network referenced in [8] (network size from (0,0) to (100,100), cluster range = 30m, sink at (50,175)). According to the above formulae for connectivity, the radio range for inter-cluster communication should be set as 180m, which means all cluster heads can almost communicate with base station directly.

In this paper, we set inter-cluster transmission range as 2.5r, where r is the intra-cluster range referenced as before. Because we assume the network is a dense network (> $1/100m^2$), it can guarantee that most cluster heads are member nodes of the largest connected component of graph composed by all cluster heads. In the next section, we will discuss the relationship between the inter-cluster transmission range, R and the number of independent connected component of a graph by experiments. The theoretical analysis will be made in the later work".

```
IF isClusterHead
Broadcast (myID, WEIGHT)
Wait T1
ParentNode = Neighbor which send Max WEIGHT
Send ( myID, CHILD) to ParentNode
IF isCluster Head
Booadcast TDMA schedule to active node
```

Fig. 6. Routing Tree Construction (RTC) algorithm

After clustering, cluster heads broadcast Weight messages within radius R, which contain node ID and weight W. Cluster head compared its own weight and the weight contained in Weight message received from its neighbor cluster head. If it has smaller weight, it selects the node that has the largest weight as its parents and sends the CHILD message to notify the parent node. Finally, after a specified time, a routing tree will be constructed, which root node has the largest weight among all cluster heads in the same independent connected component. After routing tree construction, cluster heads broadcast a TDMA schedule to their active member nodes to be ready for data gathering. The pseudo code for cluster formation is shown in Fig 6.

For example, as shown in Fig. 2, node A~E are cluster heads with their weight in parenthesis. B will receive *WEIGHT* message from A, C, D, E and select node A to be its parent. Similarly, node D and E choose B as their parent, while C chooses A as its parent. Node A receives *WEIGHT* message from node B and C, but their weight is less than node A. Then A will be the root node that communicates with the base station and routing tree is build.

We define weight *W* of node *i* as $w_i = \frac{D(RSS_i)\times E_a}{D(RSS_{\max})\times E_{residual}}$, where RSS_i *denotes* node i's received signal strength of the signal broadcasted by the base station, RSS_{max} is a constant which is determined by the location of base station, and function D is used for estimating the distance between node i and the base station. After the deployment of sensors, the base station broadcasts probing message to all sensors and sensors acquire the *RSS* according to the received signal strength. *RSS* maintains constant during the network lifetime unless base station varies its location or sensor nodes are mobile. Apparently, node that is closer to the base station and locates in a subregion with full energy would be the root node of routing tree for its higher weight.

TABLE III
SIMULATION PARAMETERS

Parameters	Value
Network Filed	(0,0)~(100,100)
Node numbers	100~500
Cluster radius *r*	30m
Sensing radius r_s	10m
Sink position	(50,200)
Initial energy	2 J
Data packet size	525 Bytes
Broadcast packet size	25 Bytes
$E_{threshold}$	0.01J
E_{elec}	50nJ/bit
e_{fs}	$10nJ/bit/m^2$
e_{amp}	$0.0013pJ/bit/m^4$
E_{DA}	5nJ/bit/signal
Threshold distance d_0	75m
Data Cycles per round(*L*)	5

V. Performance evaluation

A. Simulation parameter

In the simulation experiments, network lifetime has two definitions: *First Node Dies* (FND), the time when the first node dies in network and *Last Node Dies* (LND), the time when the last node dies. The parameters of simulations are listed in Table III. Unless otherwise specified, every simulation result shown below is the average of 200 independent experiments where each experiment uses a different randomly-generated uniform topology of sensor nodes. For simplicity, we assume the probability of signal collision and interference in the wireless channel is ignorable and the radio transmitter, radio amplifier and data fusion unit are the main energy consumers of a sensor node, so we calculate the energy consumption of these three components in the simulation. In simulation, we use the same radio model shown in [5] for the radio hardware energy dissipation. This radio model has been widely adopted in the following study [3] [6] [10] [11].

B. Simulation results and analysis

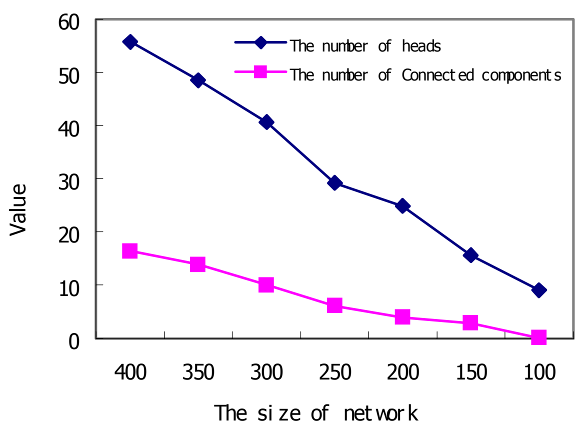

Fig 7. The number of heads and the number of connected components

Fig 7 is the illustration of the relationship between the number of cluster heads, the number of independent connected components, and network size. Companied with the network size enlargement, it can be seen that the number of independent components is increasing. When network size is (100x100), the number of connected component equal 1 in most case, which means the graph composed by cluster heads is connected. When network size is (400x400), after clustering phase, almost sixty cluster heads will be generated. However, such number of nodes cannot guarantee the connectivity of all cluster heads. As shown in Fig 7, the number of independent components almost reaches 20. Obviously, our algorithm can work well when the node density is high enough, namely more than 0.01/m2.

Fig 8 proves that EAP can effectively provide required QoS. On the one hand, providing QoS lower than required may save energy at risk of failing to meet application requirement. On the other hand, providing higher QoS than the required by a specific application will decrease the efficiency of energy utilizing. However, EAP cannot provide a perfect matching between expected QoS and obtained QoS. It maybe does not

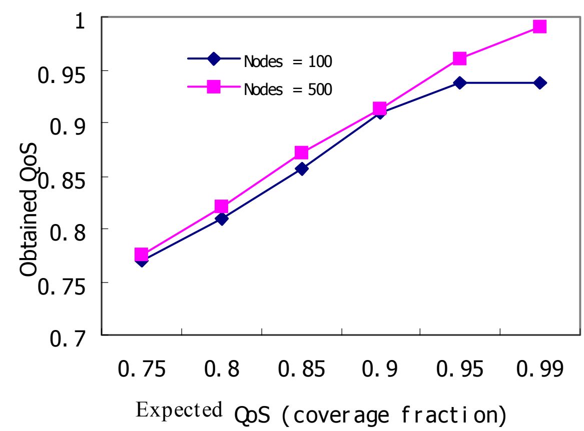

Fig 8. Expected QoS vs. obtained Qos

work well in some applications, which strictly require that the deviation between expected QoS and obtained QoS is small enough. As shown in Fig 8, if there are 100 nodes deployed in monitoring area, the obtained QoS cannot meet the application requirement when the expected QoS exceeds 95%. Because even if all nodes are turned on, they cannot cover

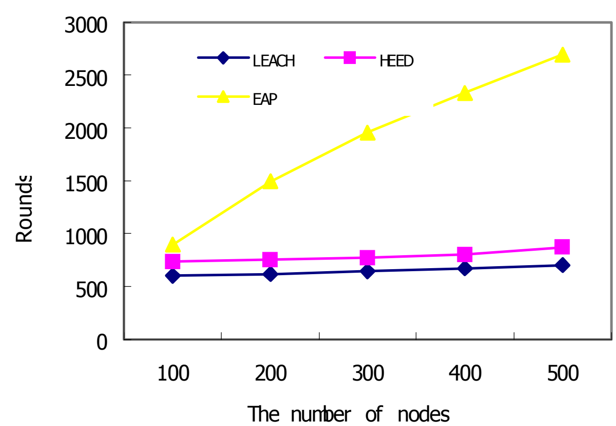

Fig 9. The number of nodes vs. network lifetime (expected QoS = 0.95)

95% fraction of whole monitoring area.

Because HEED and LEACH cannot provide the topology control function, each node need collect the temperature information and transmit it to its cluster head, even if it is a redundant node. So, the two algorithms fail to prolong network lifetime when node density is high. Conversely, for EAP protocol, through intra-cluster coverage method, the number of actual active nodes is only determined by expected QoS. As shown in Fig 9 and Fig 10, it is easy to find that sensor

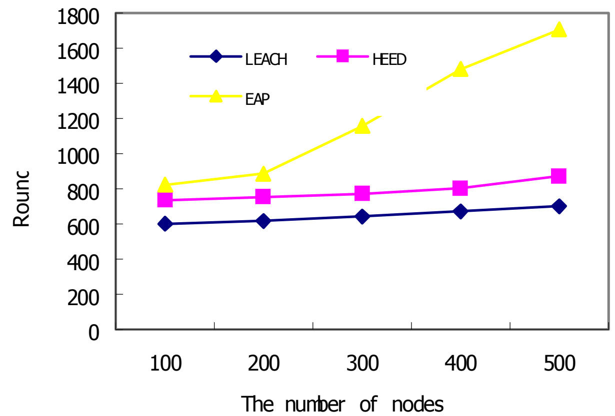

Fig 10. The number of nodes vs. network lifetime (expected QoS = 0.99)

network lifetime almost be linear in the number of nodes which are deployed in monitoring area.

VI. CONCLUSIONS

In this paper, we present EAP, a novel energy efficient data gathering protocol with intra-cluster coverage. EAP clusters sensor nodes into groups and builds routing tree among cluster heads for energy saving communication. In addition, EAP introduces the idea of area coverage to reduce the number of working nodes within cluster in order to prolong network lifetime. Simulation results show EAP outperforms far better than LEACH. Compared to HEED, though EAP performs almost the same as HEED when node density is low, it has far better performance than HEED when node density goes higher than 0.01nodes/m^2.

ACKNOWLEDGMENT

This research is supported in part by the China NSF under grant No. 60673154, the National Grand Fundamental Research 973 Program of China under grant No.2006CB303000, Hong Kong Polytechnic University under the ICRG grant A-PF77, and the China NSF under grant No.60573132. The Conference participation is supported by Nokia Bridging the World Program.

REFERENCES

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the World with Wireless Sensor Networks," in International Conference on Acoustics, Speech, and Signal Processing (ICASSP2001), Salt Lake City, Utah, May 2001.

[2] G. J. Pottie and W. J. Kaiser, "Wireless Integrated Newtork Sensors," Communications of the ACM, vol. 43, no. 5, pp. 51–58, May 2000.

[3] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the Hawaii International Conference on System Sciences, Maui, Hawaii, Jan 2000.

[4] S. Lindsey and C. Raghavendra. PEGASIS: Power-Efficient gathering in sensor information systems. In IEEE Aerospace Conference, volume 3, pages 3--1125 -- 3--1130. IEEE, 2002.

[5] W. R. Heinzelman, et al. An Application -Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications, vol. 1, no. 4, Oct. 2002

[6] SH. Lee, JJ. Yoo and TC. Chung. (2004). Distance-based Energy Efficient Clustering for Wireless Sensor Networks. Proceedings of the 29th Annual IEEE international Conference on Local Computer Networks (LCN'04).

[7] V. Mhatre, C. Rosenberg, "Homogeneous vs Heterogeneous Clustered Networks: A Comparative Study", in Proceedings of IEEE ICC 2004, June 2004.

[8] O. Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach," in Proceedings of IEEE INFOCOM, March 2004.

[9] Q. Xue, A. Ganz, "Maximizing Sensor Network Lifetime: Analysis and Design Guides", in Proceedings of MILCOM, October 2004.

[10] H. Chan and A. Perrig. (2004). ACE: An Emergent Algorithm for Highly Uniform Cluster Formation. Proceedings of the First European Workshop on Sensor Networks (EWSN).

[11] M. Ye, C.F. Li, G. Chen and Ji Wu, EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks, Accepted by International Journal of Ad Hoc & Sensor Wireless Networks.

[12] J. Hill, R.Szewczyk, A. Woo, S. Hollar, D.E. Culler and K.S.J. Pister. (2000). System Architecture Directions for Networked Sensor. Architectural support for Programming Languages and Operating Systems, pp.93-104.

[13] D. E. Culler, J.Hill, P. Buonadonna, R. Szewczyk, and A.Woo, "A Network-Centric Approach to Embedded Software for Tiny Devices," Proceedings of First InternationalWorkshop on Embedded Software (EMSOFT 2001), Tahoe City, California, October 2001, pp. 114-130.I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Network: a Survey,", Computer Network Journal, Vol 38, pp. 392-422,2002.

[14] H. Zhang and J.C. Hou, "Maintaining scheme coverage and connectivity in large sensor networks," in Proceedings of NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc wireless, and Peer-to-Peer Networks, 2004.

[15] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C.D. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks," in Proceedings of the First International Conference on Embedded Networked Sensor Systems, pp 28-39, ACM Press, 2003.

[16] Y. Gao, K. Wu, and F. Li, "Analysis on the redundancy of wireless sensor networks," in Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications (WSNA 03), September 2003, San Diego, CA.

[17] Ming Liu, et al. "A Distributed Energy-Efficient data Gathering and aggregation Protocol for Wireless sensor networks", in Journal of Software, Dec, 2005.