EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks

Mao Ye[1], Chengfa Li[1], Guihai Chen[1] and Jie Wu[2]
[1]National Laboratory of Novel Software Technology, Nanjing University, China
[2]Department of Computer Science and Engineering, Florida Atlantic University, USA

Abstract

Data gathering is a common but critical operation in many applications of wireless sensor networks. Innovative techniques that improve energy efficiency to prolong the network lifetime are highly required. Clustering is an effective topology control approach in wireless sensor networks, which can increase network scalability and lifetime. In this paper, we propose a novel clustering schema EECS for wireless sensor networks, which better suits the periodical data gathering applications. Our approach elects cluster heads with more residual energy through local radio communication while achieving well cluster head distribution; further more, it introduces a novel method to balance the load among the cluster heads. Simulation results show that EECS outperforms LEACH significantly with prolonging the network lifetime over 35%.

1 Introduction

Continued advances of MEMS and wireless communication technologies have enabled the deployment of large scale wireless sensor networks (WSNs) [1]. The potential applications of WSNs are highly varied, such as environmental monitoring, target tracking and military [2]. Sensors in such a network are equipped with sensing, data processing and radio transmission units while the power is highly limited. Due to the sensors' limited power, innovative techniques that improve energy efficiency to prolong the network lifetime are highly required.

Data gathering is a common but critical operation in many applications of WSNs, where data aggregation and hierarchical mechanism are commonly used techniques. Data aggregation can eliminate the data redundancy and reduce the communication load [3]. Hierarchical (clustering) mechanisms are especially effective in increasing network scalability and reducing data latency, which have been extensively exploited. LEACH [4] which is the first clustering protocol, proposes a two-phase mechanism based on single-hop communication. The plain node transmits the data to the corresponding cluster head and the cluster head transmits the aggregated data to the base station (BS). HEED [5] selects cluster heads through $O(1)$ time iteration according to some metric and adopts the multi-hop communication to further reduce the energy consumption. PEGASIS [6] improves the performance of LEACH and prolongs the network lifetime greatly with a chain topology. But the delay is significant although the energy is saved. There are some other related work [7–9] which efficiently use energy through clustering.

In this paper, we propose and evaluate an energy efficient clustering scheme (EECS) for periodical data gathering applications in WSNs. In the *cluster head election* phase, a constant number of candidate nodes are elected and compete for cluster heads according to the node residual energy. The competition process is localized and without iteration, thus it has much lower message overhead. The method also produces a near uniform distribution of cluster heads. Further in the *cluster formation* phase, a novel approach is introduced to balance the load among cluster heads. EECS is fully distributed and more energy efficient and the simulation results show that it prolongs the network lifetime as much as 135% of LEACH.

The remainder of this paper is organized as follows. Section 2 outlines the data gathering issues in WSNs. Section 3 exhibits the details of EECS and Section 4 analyzes the properties of EECS. Section 5 evaluates the performance of EECS. Finally, Section 6 gives the conclusion and future work.

2 Problem Outline

Data gathering is a typical application in WSNs. Sensors periodical sense the environment and transmit the data to the base station (BS), and the BS analyzes the data to draw some conclusions about the activity in the area. We make a few assumptions about the network model and introduce the radio model before the problem statements.

0-7803-8991-3/05/$20.00 © 2005 IEEE

2.1 Network Model

To simplify the network model, we adopt a few reasonable assumptions as follows: 1)N sensors are uniformly dispersed within a square field A; 2)All sensors and BS are stationary after deployment; 3) The communication is based on the single-hop; 4)Communication is symmetric and a sensor can compute the approximate distance based on the received signal strength if the transmission power is given; 5)All sensors are location-unaware; 6)All sensors are of equal significance.

We use a simplified model shown in [4] for the radio hardware energy dissipation as follows. We refer readers to [4] for more details. To transmit an l-bit data to a distance d, the radio expands:

$$E_{T_x}(l,d) = \begin{cases} l \times E_{elec} + l \times \epsilon_{fs} d^2, & d < d_{crossover} \\ l \times E_{elec} + l \times \epsilon_{mp} d^4, & d \geq d_{crossover} \end{cases} \quad (1)$$

The first item presents the energy consumption of radio dissipation, while the second presents the energy consumption for amplifying radio. Depending on the transmission distance both the free space ϵ_{fs} and the multi-path fading ϵ_{mp} channel models are used [11]. When receiving this data, the radio expends: $E_{R_x}(l) = l \times E_{elec}$. Additionally, the operation of data aggregation consumes the energy as E_{DA}.

2.2 Problem Statement

Once a sensor node runs out its energy, we consider the network is dead because some area cannot be monitored any more. Periodical data gathering applications in large scale sensor networks appeal the design of scalable, energy efficient clustering algorithms. Thus our primal goals in EECS are as follows: 1) *fully distributed manner.* Sensors interact with each other through localized communication; 2) *low control overhead.* It is desirable to reduce control overhead to extend the time of data gathering; 3) *load balanced clustering mechanism.* Balance the load among the sensors, especially among the cluster heads. In the next section, we will describe the EECS algorithm in details.

3 EECS Details

EECS is a LEACH-like clustering scheme, where the network is partitioned into a set of clusters with one cluster head in each cluster. Communication between cluster head and BS is direct (single-hop). For easy reference, we summarize the notations in Table 1.

Table 1: Meanings of the Notations

Notation	Meaning
T	a threshold between 0 and 1
$E_{residual}$	the residual energy of node
CH	the set of cluster heads
m_j	the sum of members in cluster j
P	the set of plain nodes
CH_i	the i^{th} node in CH
P_j	the j^{th} node in P
$d(x,y)$	the distance between node x and y
$EX(x)$	the expectation of x
$R_{compete}$	broadcast radius of candidate nodes

In the network deployment phase, the BS broadcasts a "hello" message to all the nodes at a certain power level. By this way each node can compute the approximate distance to the BS based on the received signal strength. It helps nodes to select the proper power level to communicate with the BS. As will shown in Section 3.2, we will use this distance to balance the load among cluster heads. In *cluster head election* phase, well distributed cluster heads are elected with a little control overhead. And In *cluster formation* phase, a novel weighted function is introduced to form load balanced clusters. Detailed descriptions of these two phases are in the following subsections.

3.1 Cluster head election

In this phase, several cluster heads are elected. Nodes become `CANDIDATE` nodes with a probability T and then broadcast the `COMPETE_HEAD_MSGs` within radio range $R_{compete}$ to advertise their wills. Each `CANDIDATE` node checks whether there is a `CANDIDATE` node with more residual energy within the radius $R_{compete}$. Once the `CANDIDATE` node finds a more powerful `CANDIDATE` node, it will give up the competition without receiving subsequential `COMPETE_HEAD_MSGs`. Otherwise, it will be elected as `HEAD` in the end.

3.2 Cluster formation

In this phase, each `HEAD` node broadcasts the `HEAD_AD_MSG` across the network, while the `PLAIN` nodes receive all the `HEAD_AD_MSGs` and decide which cluster to join. Most of existed metric for `PLAIN` nodes to make decisions is the distance metric. For example in [4] or [7], the `PLAIN` nodes choose the cluster head that require minimum communication according to the received signal strength. However, pursuing efficient energy consumption of the `PLAIN` nodes only may lead `HEAD` nodes exhausted quickly during the *data transmission* phase.

In the *data transmission* phase, the consumed energy of cluster head i, $E(CH_i)$ is as follows, assuming $d(CH_i, BS) > d_{crossover}$.

$$E(CH_i) = m_i l E_{elec} + (m_i + 1) l E_{DA} + l(E_{elec} + \epsilon_{mp} d^4) \quad (2)$$

Observing formula 2, energy consumption of $E(CH_i)$ is composed of three parts: data receiving, data aggregation and data transmission. In the field, several cluster heads may be near the BS, while some are far away. The energy expended during data transmission for far away cluster heads is significant, especially in large scale networks. Since $d(CH_i, BS)$ has been fixed after *cluster head election* , we should justify the cluster size for each cluster head to balance their load across the network. The larger $d(CH_i, BS)$ is, the smaller member size m_i the cluster head CH_i should accommodate.

Energy consumption of the PLAIN node P_j during transmitting the data to CH_i obey the formula 1. Let $E(P_j)$ be the energy consumed by P_j. If P_j always chooses the cluster head CH_{best} with $\min\{E(P_j)\}$, CH_{best} may be exhausted due to long distance data transmission to the BS and immoderate cluster size, although the energy of P_j is saved. Thus, PLAIN node P_j in EECS chooses the cluster head by considering not only saving its own energy but also balancing the workload of cluster heads,i.e. two distance factors: $d(P_j, CH_i)$ and $d(CH_i, BS)$.

We introduce a weighted function $cost(j,i)$ for the PLAIN node P_j to make a decision, which is

$$cost(j,i) = w \times f(d(P_j, CH_i)) + (1-w) \times g(d(CH_i, BS)), \quad (3)$$

and P_j chooses CH_i with $\min\{cost\}$ to join.

In formula 3, f and g are two normalized functions for the distance $d(P_j, CH_i)$ and $d(CH_i, BS)$ respectively:

$$f = \frac{d(P_j, CH_i)}{d_{f\_max}} \quad g = \frac{d(CH_i, BS) - d_{g\_min}}{d_{g\_max} - d_{g\_min}} \quad (4)$$

where $d_{f\_max} = EX(\max\{d(P_j, CH_i)\})$, $d_{g\_max} = \max\{d(CH_i, BS)\}$ and $d_{g\_min} = \min\{d(CH_i, BS)\}$.

f subfunction in *cost* guarantees that members choose the closest cluster head in order to minimize energy consumption of the cluster members, While g subfunction makes the nodes join the cluster head with small $d(CH_i, BS)$ to alleviate the workload of the cluster heads farther from the BS. w is the weighted factor for the tradeoff between f and g. The experiments in Section 6 will show that the optimal value of w depends on the specific network scale.

3.3 Synchronization issues

Synchronization between each phase should be guaranteed that each node has enough time to complete the procedure; while within each phase, synchronization among the nodes is not necessary and idle nodes will turn to sleep till the phase ends. In EECS, it is achieved by having the BS periodically broadcast synchronization signals to all nodes.

4 EECS Analysis

In this section, we analyze the performance of EECS in details and explain how to set the parameters T and $R_{compete}$.

Lemma 1. *The control overhead complexity across the network is $O(N)$, where N is the number of nodes.*

Proof. Observing EECS, every node sends out small constant-length control messages each round without iteration. Each HEAD node sends three messages which are COMPETE_HEAD_MSG, HEAD_AD_MSG and SCHEDULE_MSG; each CANDIDATE node sends two messages which are COMPETE_HEAD_MSG and JOIN_CLUSTER_MSG; while the others send JOIN_CLUSTER_MSGs only. Clearly, the total control overhead is $NT + N$, whose asymptotic order is $O(N)$. □

Good quality HEAD nodes should be guaranteed by enough competition of the CANDIDATE nodes. Since T is the only crucial factor which affects the sum of CANDIDATE nodes, it must be large enough to guarantee enough CANDIDATE nodes. On the other hand, the larger T is, the more overhead is produced in the *cluster head election* phase. So, we must properly set T to reduce the overhead with guaranteed HEAD quality.

In LEACH, there is no interaction during the *cluster head election*. So the control overhead is near optimal, which is $2NP + N(1-P) = NP + N$, where P is similar to T in [4]. Thus the overhead of EECS is only $(1+T)/(1+P)$ times of LEACH. In HEED, HEAD nodes are elected with iteration. Although the communication is localized and the algorithm terminates in $O(1)$ iteration, HEED still produces much more overhead with the upper bound $N_{iter} \times N$. Clearly, our approach is better than HEED. The above property shows that the control overhead of EECS is low significantly.

Lemma 2. *There is at most one cluster head in every $R_{compete}$ radio covered range.*

Proof. Let S be the set of all sensor nodes. And for $\forall x \in CH$, let $C_x = \{y | d(y,x) \le R_{compete}, y \in S\}$. For contradiction, we assume that there is a node $y \in C_x$ which is also a cluster head. According to the competition metric in *cluster head election*, $x.E_{residual} > z.E_{residual}, \forall z \in C_x$. Since $y \in C_x$, then $x.E_{residual} > y.E_{residual}$. The communication is symmetric in the network model of EECS. If y is the cluster head, $y.E_{residual} > x.E_{residual}$ as x is within the distance $R_{compete}$, which is a contradiction.

So, for $\forall x \in CH$, $\forall y \in C_x$, there is $y \notin CH$. □

In [4], the author proves that there is an optimal number of cluster heads k_{opt} in a given scene. Since EECS is a LEACH-like protocol, we want to elect k_{opt} cluster heads every round. According to *Lemma* 2, $R_{compete}$ affects the cluster heads directly. So we compute the optimized value of $R_{compete}$, denoted by R_{opt} in the following lemma.

Lemma 3. *There is an optimal range* R_{opt} *for* $R_{compete}$, *which is* $\sqrt{\frac{A}{\pi k_{opt}}}$, *where* k_{opt} *is the optimal range of* $|CH|$.

Proof. Let $P(CANDIDATE)$ be the probability of one node being CANDIDATE node, so the sum of CANDIDATE nodes n is $P(CANDIDATE) \times N$. In the $R_{compete}$ radius range, there are m nodes in CANDIDATE state(boundary cases are ignored), where $m = n \times \frac{\pi R^2_{compete}}{A}$.

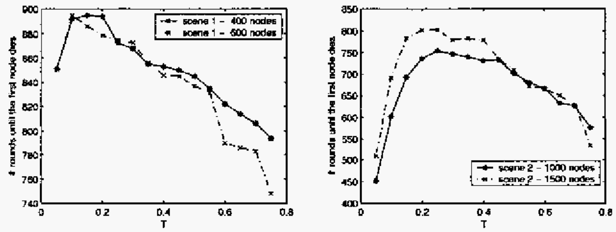
Since all nodes have the same capacity, these m nodes have equal probability to be HEAD, then the probability of one node being HEAD node $P(HEAD) = P(HEAD|CANDIDATE) = \frac{A}{\pi R^2_{compete} N}$. So the expectation of the sum of cluster heads $EX(|CH|) = N \times P(HEAD) = \frac{A}{\pi R^2_{compete}}$.

In order to optimize energy consumption, we want to let $EX(|CH|)$ equal to k_{opt} in [4]. Combining the induction in [4] and the formula of $EX(|CH|)$, we can find that the optimal radius R_{opt} is $\sqrt{\frac{A}{\pi k_{opt}}}$. □

In LEACH, cluster heads are elected simply at random. As a result, the distribution of the cluster heads are not ensured and may be non-uniform. Some members have to expend much more energy to communicate with the corresponding cluster heads far away. The last two lemmas show that there is one and only cluster head within any $R_{compete}$ with high probability. Thus the cluster heads in EECS are distributed evenly.

5 Simulation

In this section, we evaluate the performance of EECS protocol implemented with MATLAB. For simplicity, we assume the probability of signal collision and interference in the wireless channel is ignorable. And we adapt the same MAC protocols in EECS as in LEACH. In order to explain the relations between the network scale and the parameters in EECS, we run each kind of simulation in two different scenes, which are normal scale scene (scene 1) and large scale scene (scene 2) respectively. The parameters of simulations are listed in TABEL2, and the parameters of the radio model are the same as LEACH [4]. Unless otherwise specified, every simulation result shown below is the average of 100 independent experiments where each experiment uses a different randomly-generated uniform topology of sensor nodes.



(a) normal scene (b) large scene

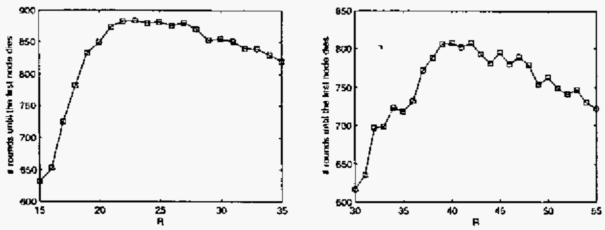
Figure 1: The impaction of T on the network lifetime:(a) normal scene, (b) large scene

Table 2: Parameters of Simulations

Parameter	scene 1	scene 2
Area	100×100	200×200
Location of BS	$(50, 200)$	$(100, 350)$
N	400(600)	1000(1500)
Initial energy	$0.5J$	$1.0\ J$
E_{elec}	$50\ nJ/bit$	
ϵ_{fs}	$10\ pJ/bit/m^2$	
ϵ_{mp}	$0.0013\ pJ/bit/m^4$	
$d_{crossover}$	$87\ m$	
E_{DA}	$5\ nJ/bit/signal$	
Packet size	$4000\ bits$	

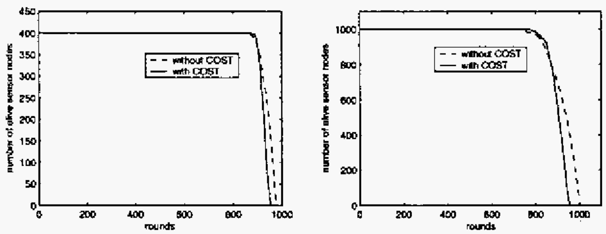
Lifetime is the criterion for evaluating the performance of sensor networks. In the simulation, we measure the lifetime in terms of round when the first node dies. We use the energy utilization rate η to evaluate the efficiency of energy consumption which is defined as the ratio of the total energy consumed when the first node dies to the initial total energy. A high η implies that energy consumption is distributed well across the network.

We first examine the impact of T on the network lifetime, as the scales are different. We have done two



(a) normal scene (b) large scene

Figure 2: The impaction of $R_{compete}$ on the network lifetime:(a) normal scene, (b) large scene



(a) normal scene (b) large scene

Figure 3: The impaction of *cost* on the network lifetime:(a) normal scene, (b) large scene

independent experiments in different scales. In normal scale, $N = 400, 600$, $R_{compete} = 26, 22$, $w = 0.8$; in large scale, $N = 1000, 1500$, $R_{compete} = 40, 35$, $w = 0.6$. As T varies from 0.05 to 0.75, Figure 1 shows the relation between T and the network lifetime. There is an optimal range for the value of T, which is about $0.1 \sim 0.3$ in the given scene. According to the explanation about T in Section4, T must be properly set with guaranteed HEAD quality and low overhead. Another point needed to be mentioned that the optimal value T_{opt} decreases when the network density increases. It can be explained that there is an optimal sum of CANDIDATE nodes in a given network coverage size.

In the experiment shown in Figure 2, we demonstrate *Lemma* 3 by observing the relation between $R_{compete}$ and the network lifetime. In scene 1, $N = 400$ and $k_{opt} = 4 \sim 7$, so the optimal value R_{opt} is between $21 \sim 28$; In scene 2, $N = 1000$ and $k_{opt} = 6 \sim 10$, so R_{opt} is between $36 \sim 46$. Observing the impact on network lifetime when $R_{compete}$ varies, Figure 2 suggests that the optimal value of $R_{compete}$ is about 25 in scene 1 and about 40 in scene 2. Both results fall into the optimal range computed prior.

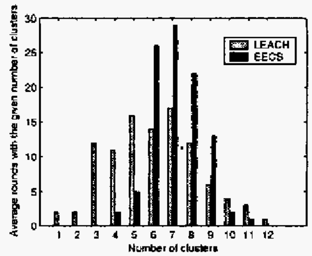
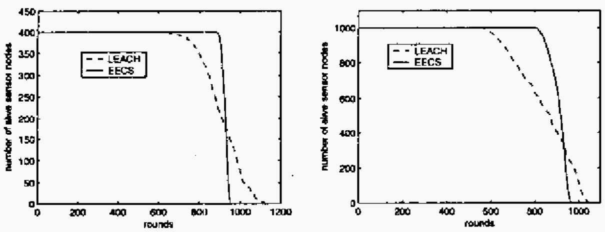


Figure 4: The number of clusters in each round in both EECS and LEACH (scene 1)

In Figure 3, the experiment shows the efficiency of *cost* introduced to balance the load among the cluster heads, where the dash line denoted as the method without considering the cluster heads' load balance issue. We set w at 0.8 in scene 1 and 0.6 in scene 2 respectively. Comparing the $without - cost$ method ($w = 1$) with the $with - cost$ method, we find that the *cost* indeed extends the network lifetime. The value of w is determined by the specific scene. While the network grows larger, the difference among $d(CH_i, BS)$s impacts the load balance among the cluster heads more and more distinctly. So w should be decreased and the PLANE node will consider more about the load of cluster head when joining the cluster. That's why the value of w is bigger in scene 1 than in scene 2. In this paper, the *cost* function is simple, and we will optimize the *cost* function in the next work.

Finally, we compare the performance of EECS with the original-LEACH [4] based on the same assumptions in [4]. In scene1, $k_{opt} = 6$, $T = 0.2$, $R_{compete} = 26$ and $w = 0.8$; in scene2, $k_{opt} = 9$, $T = 0.15$, $R_{compete} = 40$, $w = 0.6$. In Figure 4, it exhibits the distribution of the number of clusters in random selected 100 rounds in both EECS and LEACH. Shown as the figure, the number of clusters varies widely in each simulation run in LEACH; on the other hand, the cluster number varies narrowly at the k_{opt} range in EECS. In LEACH, the clusters in each round is not controlled although the expectation is aware; while in EECS, we use the R_{opt} radio radius to set up k_{opt} clusters in all probability in each round. Figure 5 shows the variation of total number of sensors still alive when the simulation time lapses. In scene1, EECS prolongs the lifetime over 35% against LEACH. The energy utilization rate is about 93% in EECS, while only 53% in LEACH. The reason is that EECS always achieves the well distributed cluster heads with considering the residual energy; further, we consider to balance the load among the cluster heads with weighted function.



(a) normal scene (b) large scene

Figure 5: Performance comparison of EECS and LEACH:(a) normal scene, (b) large scene

In Figure 5-b, the efficiency of EECS is more distinct when the network scale grows. In [5], the author mentions that the original LEACH outperforms HEED When based on the same assumptions in [4] which is identical with EECS. In order to save energy further, HEED adopts the multi-hop communication among the cluster heads during the inter-cluster communications in the *data transmission* phase. Notice that we focus on the cluster set-up algorithm but not the data transmission approach in our current work. Future work will consider the multi-hop technique in the inter-cluster communication. Readers should refer to [12] for details about the multi-hop routing in clustered networks.

6 Conclusion and Future Work

In this paper, we present a novel distributed, energy efficient and load balanced clustering scheme applied for periodical data gathering. EECS produces a uniform distribution of cluster heads across the network through localized communication with little overhead. What's more, a novel approach has been introduced to distribute the energy consumption among the sensors in the *cluster formation* phase. Simulation results show that EECS prolongs the network lifetime as much as 135% of LEACH and the total energy is efficiently consumed.

All of our contributions here are focused on the cluster set-up stage. There are still much space to improve the performance of data transmission. In the large scale sensor networks, multi-hop communication is a mainstream technique for energy saving. We will remove the assumption of single-hop and design an energy efficient protocol for both intra-cluster and inter-cluster data transmission in the future work.

Acknowledgments

The work is partly support by China 973 project (No. 2002CB312002) and TRAPOYT award of China Ministry of Education.

References

[1] D. Estrin, et. al., "Instrumenting the world with wireless sensor networks," in *the International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 2001.

[2] Akyildiz I.F., et. al.,"A Survey on Sensor Networks," *IEEE Communications Magazine*, August, pp. 102-114, 2002.

[3] B. Krishnamachari, et. al., "The impact of data aggregation in wireless sensor networks," in *the 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW'02).*

[4] W. Heinzelman, et. al., "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, 1(4):660-669, 2002.

[5] O. Younis, et. al., "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," *IEEE Transactions on Mobile Computing*, 3(4):660-669, 2004.

[6] S. Lindsey, et. al.,"PEGASIS: Power-Efficient Gathering in Sensor Information Systems," *IEEE Aerospace Conference Proceedings*, Vol. 3, 9-16 pp. 1125-1130, 2002.

[7] S. Bandyopadhyay, et. al., "An Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," in *IEEE INFOCOM'03.*

[8] H. Chan, et. al., "ACE: An Emergent Algorithm for Highly Uniform Cluster Formation," in *the First European Workshop on Sensor Networks (EWSN)*, January 2004.

[9] J. Kamimura, et. al., "Energy-Efficient Clustering Method for Data Gathering in Sensor Networks," in *the Annual International Conference on Broadband Networks*, October 2004.

[10] J. Hill, et. al., "System Architecture Directions for Networked Sensors", *Architectural support for Programming Languages and Operating Systems*, pp. 93-104,2000.

[11] T. Rappaport, Wireless Communication: Principles & Practice. Prentice-Hall, 1996.

[12] V. Mhatre, et. al., "Design guidelines for wireless sensor networks: communication, clustering and aggregation," *Ad Hoc Networks Journal, Elsevier Science*, 2(1):45-63, 2004.