Shan-shan MA, Jian-sheng QIAN

School of Computer Science and Technology, China University of Mining and Technology

Location-Unaware Node Scheduling Schemes Based on Boundary Nodes in Wireless Sensor Networks

***Abstract**. Node scheduling scheme of sensor nodes is one of the most important method to solve the energy-constrained wireless sensor networks. Traditional methods of node scheduling that without location information leads to a node in the border of monitored region first death due to no more chance enter into sleep state, and then the death spread to the central region. This phenomenon is called as inequality sleep problems. To address this problem, we propose a method to determine some boundary nodes only using the minimum cost of nodes and the neighbors' distance without any location information. We develop a location-Unaware nodes-scheduling schemes based on these determined boundary nodes called as LUNSB. Simulation results demonstrate that B-LUNS not only alleviates the inequality sleep problems, but also prolongs network lifetime.*

Streszczenie:** Jedną z najważniejszych metod ograniczenia energii sieci czujnikowej jest odpowiedni schemat szeregowania węzłów sieci. W tradycyjnych metodach szeregowania węzły na granicy monitorowanego obszaru uszkadzane są jako pierwsze, ponieważ mają małe szanse na wejście w stan spoczynku. Aby rozwiązać ten problem zaproponowano określenie kilku węzłów granicznych przyjmując, jako kryteria, tylko minimalny koszt oraz odległość do sąsiednich węzłów, bez uwzględnienia lokalizacji. Opracowano schematy szeregowania węzłów o nieznanej lokalizacji LUNSB w oparciu o granice zdefiniowane powyżej. Wyniki symulacji wykazują, że B-LUNS nie tylko łagodzą problem nierównego stanu spoczynku ale i przedłużają czas życia sieci.* ***Schematy szeregowania węzłów o nieznanej lokalizacji oparte o węzły graniczne w bezprzewodowej sieci czujnikowej

Keywords: wireless sensor networks; inequality sleep problems; node scheduling algorithm; Location-Unaware
Słowa kluczowe: Bezprzewodowe sieci czujnikowe, Problemy nierównego stanu spoczynku, Algorytm szeregowamnia węzłów, Nieznana lokalizacja

Introduction

Large-scale wireless sensor networks rely on thousands of tiny sensors to observe and influence the physical world. One of the main challenges is to maintain long network lifetime as well as sufficient sensing area [1]. It is well accepted that a sensor network should be deployed with high density in order to prolong the network lifetime. But such high-density network causes severe problems of scalability, redundancy, and radio channel contention.

A broadly-used strategy for reducing energy consumption in wireless sensor networks is to turn off redundant sensors by scheduling sensor nodes to work alternatively [2]. But selecting the optimal sensing ranges for all the sensors is a well-known NP-hard problem [3]. Existing reports [4 and 5] for determining redundant nodes are mostly location-based with the Global Positioning System (GPS) and the directional antenna technology. However the energy cost and system complexity involved in obtaining geography (location, direction, or distance) information may compromise the effectiveness of proposed solutions as a whole since GPS and other complicated hardware devices consume too much energy and the costs are too high for tiny sensors.

Several researchers have proposed the node scheduling schemes without the accurate location information. Jiang et al. [6] proposed random scheduling for sensor nodes that can saving energy without exchanging neighborhood information. But turning off sensors randomly may generate blind points and cannot ensure the quality of network coverage. Gao et al [1] propose a mathematical model to describe the redundancy in randomly deployed sensor networks. The results indicate that: a sensor requires about 11 neighbors to get a 90% probability of being a complete redundant sensor. If we only require a sensor's 90% sensing area to be covered by its neighbors, 5 neighbors are necessary. Based on this theoretical analysis, a Lightweight Deployment-Aware Scheduling (LDAS) scheme to turn off redundant sensors has been proposed [7]. LDAS uses a weighted random voting method to decide who will be eligible to fall asleep. These methods can effectively reduce network energy consumption without any location or directional information. But none of them take the border effect into account.

As exposed in [3], the nodes located near the borders of the monitored area have no other choice but to be active in each round since they are the only ones able to monitor further pieces of their own sensing coverage. Therefore, the boundary nodes may run out of their energy faster than other sensors, and then the death spread to the central region. We call this phenomenon as inequality sleep problems. However, it is difficult to determine whether the node is in the boundary location without GPS. In this paper, we propose a method to determine the boundary nodes only using the minimum cost of nodes [8] and the distance between the neighbors, without any location information.

The rest of this paper is organized as follows: section 2 describes the system model and preliminaries. Section 3 provides details to determine some boundary nodes and section 4 introduces the node-scheduling schemes based on the determined boundary-nodes（LUNSB）. In section 5, we present our experimental results for performance evaluation. Finally, section 6 gives a summary and conclusion.

System model and preliminaries

In this paper, we consider sensor nodes for which r_t is the transmission range and r_s is the sensing range. And our analysis is based on the following assumes: (1) sensors are stationary and are deployed randomly within an area; (2) A sensor's sensing range is a circle area; (3) all sensors are supposed to have the same sensing range and no two sensors can be deployed exactly at a same location; (4)no geography information is available; (5)a node can estimate the distance between itself and a neighbor; (6)$r_t \geq 2r_s$, under this condition, coverage implies connectivity[9]; (7) Sink is laid at the center of the monitoring region.

Definition 1 (Neighbor): the 1-hop neighbor set of sensor i is defined as $N(i)=\{ j\in\aleph \mid d(i,j)\leq r_t, i\in\aleph, j\neq i\}$. Where $\aleph$ represents the sensor set in the deployment region. d(i,j) denotes the distance between sensor i and j, and r_t is the radius of the sensing range.

Definition 2 (Completely and partially redundant sensor): Let C_i be the sensing area of sensor i. If $\bigcup_{j\in N(i)} C_j \supseteq C_i$, we call sensor i a completely redundant sensor, since sensor i's sensing area can be covered by its 1-hop neighbor completely. If $\bigcup_{j\in N(i)}(C_j \cap C_i \neq \varnothing)$ and $\bigcup_{j\in N(i)} C_j \not\supseteq C_i$, we call sensor i a partially redundant sensor.

Definition 3 (Boundary-node and Corner-node): Let d_i be the distance between node i and the region boundary. If $d_i < r_s$, we call node i a boundary-node. And if $d_i < 0.5r_s$, we call node i a corner- node.

Determination of the boundary nodes

Initially, every sensor may calculate the minimum cost value to the Sink through flooding [8]. We define the minimum cost value of node i as L_i.

Theorem 1: A sensor m in the deployment region. If $\forall i \in N(m)\ L_m > L_i$ is true, then sensor m is a corner-node.

The nodes determined by **theorem 1** are usually farthest from the Sink node and located at the corner of the monitoring area. Generally, these nodes run out their energy quicker than others. On account of deployed densely, the distance from these nodes to the boundary is usually less than $0.1r$. So we can get **Lemma 1**.

Lemma 1: Assuming m is a corner-node determined by Theorem 1. If $i \in N(m)$ and $d(i,m) < 0.5r_s$ is true, then sensor i is also a corner-node.

Theorem 2: A sensor m in the deployment region. In N(m), there is a sensor that has the maximal the minimum cost value than others. Assuming this sensor is i. If $L_m < L_i$ and $d(i,m) \le 0.6r_s$ is true, then sensor i is a corner-node.

Proof: When the Sink is set at the center of the monitoring region, the closer to the boundary, the greater the minimum cost value of the sensor. Assume a sensor m in monitoring areas. In N(m), there must be a sensor that has the maximal cost value than others. Supposing this sensor is i. When m is not a boundary-node, the distance between sensor i and sensor m is close to r_t as densely deployment. But when m is a boundary-node, some sensing areas are out of the motoring region. So there are fewer neighbors. The distance between sensor i and sensor m may be smaller than the situation that m is not a boundary-node. The smaller of the distance between the two nodes indicates the two nodes are closer to the boundary. We set $d(i,m) \le 0.6r_s$ and we can also adjust the value according to the actual distribution of the nodes.

Lemma 2: Assuming m is a corner-node determined by **theorem 2**. If $i \in N(m)$ and $d(i,m) \le 0.6r_s$ is true, then sensor i is a boundary-node.

As Fig1 shows, 2000 nodes are distributed randomly in the 100m×100m square region. Without location information, using our theorems and lemmas, 403 boundary-nodes (red rectangle) including 52 corner-nodes have been identified.

Location-unaware node scheduling schemes based on boundary-nodes

Coverage Redundancy Problem

Set S_i be the sensing area of sensor i, and B_i be the area covered by sensor i's neighbors. Then the rule of common nodes is:

$$\frac{B_i}{S_i} \ge Threshold$$

Threshold is the percentage of the redundant area, and can be set according to the QoS requirement of the networks. ***Threshold*** = 1 means completely redundant and threshold<1 means partially redundant. Moreover, we can get Bi based on the results of paper [1]. However, this rule is not suitable for boundary-nodes owing to some areas of S_i are out of.

In the case that 2000 nodes are randomly distributed in 100m×100m square region, with all 10m r_s. Testing 100 times, there are 733 boundary-nodes averagely. Then using the methods proposed in section 3 to identify, and approximately 405 boundary-nodes including 34 corner-nodes could be identified. After statistics, the distance between the identified corner-nodes and the bounder of the monitoring border is averagely $0.153r_s$, and the average distance between other boundary-nodes to the border is $0.37r_s$.

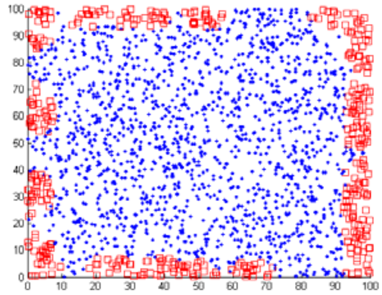


Fig 1 Performance of boundary-nodes determined

We can calculate the area that not in the monitoring region when the distance is $0.37r_s$ between the nodes and the border.

$$\arccos(0.37)r^2 - 0.37r^2\sin(\arccos(0.37)) \le S_{notin}$$
$$\le 2\arccos(0.37)r^2 - 2r\sin(\arccos(0.37))\cdot 0.37r$$
$$-\frac{1}{4}\pi r[\sin(\arccos(0.37)) - 0.37]$$

Similarly, we can calculate the area that not in the monitoring region when the distance is $0.15r_s$ between the corner-nodes to the border.

$$\arccos(0.153)r^2 - 0.153r^2\sin(\arccos(0.153)) \le S_{notin}$$
$$\le (\arccos(0.153) + \frac{\pi}{4})r^2 - 0.153r^2[\sin(\arccos(0.153)) + 0.153]$$

So we can get the judgment of boundary-nodes is:

$$\frac{B_i}{S_i - S_{notin}} \ge Threshold\cdot$$

where S_{notin} selects the minimum value.

Location-Unaware Node Scheduling Schemes Based on Boundary-Nodes (LUNSB)

Initially, assume that all nodes are active. Combined with classic LEACH cluster protocol, time is divided into fixed-length periods called rounds. Each round begins with a node-scheduling phase, in which every sensor determines whether it can be on-duty or off-duty. Then those active sensors cluster group competitively. The detail steps are below:

Step 1: Networks initialization. Firstly, Sink broadcast the ADV message through flooding. Every node may calculate the minimum cost to the Sink and record the neighbors' information. Then every sensor individually determines whether it is boundary-node or corner-node using the method proposed in section 3.

Step 2: Node-scheduling. Each node includes three states: on-duty, ready-to-off, and off duty. At the on-duty state, a sensor is responsible for sensing and communication. At the beginning of each round, every sensor determines whether it is a redundant node based on the rules in 4.1. If it is, it enters the ready-to-off with a random short time T_w to avoid that a blind point occurs after multiple sensors turning off at the same time [2]. If the node at the ready-to-off state received other sensor's ***sleep***-message, the node returns

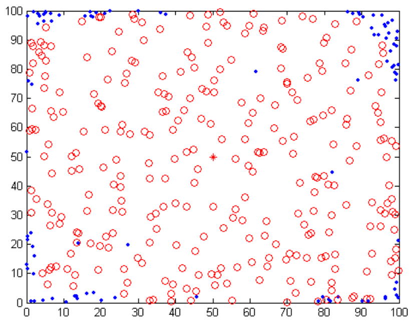
the on-duty state. Otherwise, it broadcasts sleep-message after waiting T_w time and then goes to off-duty state; fall asleep for a period of time Ts.
Step 3: Clustering. Active nodes randomly select nodes as cluster heads based on LEACH algorithm. Then the cluster heads broadcast ***hello*** messages and other active nodes select the closest head to join.
Step 4: Sensing.
Step 5: The current round end and return step 2.

Simulation study

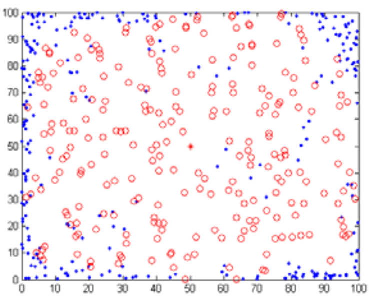
In this section, we present some experimental results as the performance evaluation of our algorithm. In our simulation, the energy model is from real sensor hardware in [10], where the transmitting, receiving (idling), and sleeping power consumption ratio is 20:4:0.01. 2000 sensors are deployed randomly in a 100m×100m square. Each sensor has a sensing range of 10 meters. In the simulation, we assume that the channel condition is perfect.

Effectiveness

2000 nodes are randomly deployed in the monitoring region. Set Threshold = 90%, run LDAS [7] and LUNSB in the same node layout conditions to compare. Then we sampled on the 5000 and 6000 round respectively, as shown in Fig2 and Fig3. Because nodes are too dense, for easy comparing, the nodes that are off-duty states are not marked. In the following figures, hollow circle represents the active nodes, and the solid point represents the death nodes.

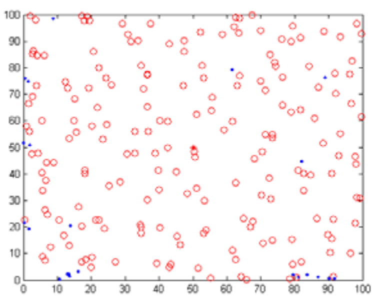


(a) 5000 round

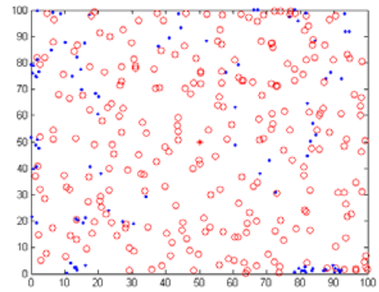


(b) 6000 round

Fig 2 LDAS



（a） 5000 round



（b） 6000 round

Fig 3 LUNSB

Compared the (a) in Fig2 and Fig3, we can find that after running 5000 rounds, there are 95 death nodes including 90 boundary-nodes using LDAS algorithm; While there are only 23 death nodes with 19 boundary-nodes using LUNSB. And after running 6000 rounds, in Fig2 (b), we can find that some monitoring blind points have appeared at the corner, due to more death sensors. While in Fig3 (b), there is no blind point with fewer death sensors and distributing evenly. Therefore, it shows that LUNSB can effectively alleviate the phenomenon of the death spread to the central region.

Network Lifetime

Fig4 shows the compared results of the number of nodes still alive on the same running round in the same simulation setting. Set ***Threshold*** = 90%, running the LDAS scheduling algorithm, the first death sensor occurs on the 1556 round, and half of the total nodes have died on the 6938 round; While running LUNSB algorithm, the first death sensor appears on the 1972 round, and after 7328 rounds half of the total sensors have run out their energy. We also change the ***Threshold***=85% to compare. The details are shown in Fig4. The results indicate that LUNSB can effectively prolong the lifetime of networks.

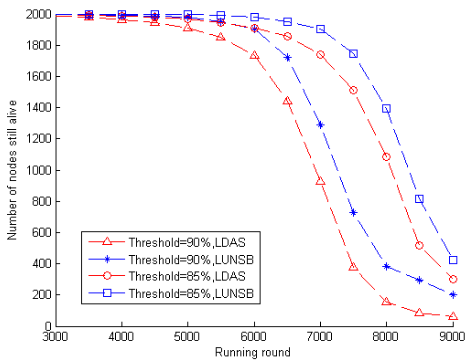


Fig 4 The number of nodes still alive

Sensing Coverage Percentage

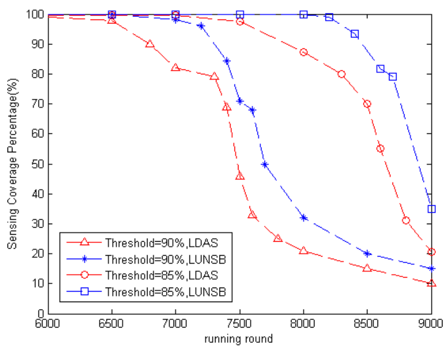


Fig 5 Sensing Coverage Percentage Compared

Fig5 shows that the sensing coverage percentage in the same nodes deployed with ***Threshold***=90% and ***Threshold***=85% respectively. From Fig5, it can be seen that it has always better coverage applying our algorithm with the same QoS requirement and the same round.

Conclusions

In this paper, we study the inequality sleep problems in location-unaware networks. To solve the problem that the boundary nodes may run out of their energy faster than other sensors, we proposed a method to determine some boundary nodes only using the minimum cost value of each node and the neighbors' distance without any location information. Then we proposed a location-unaware node scheduling schemes based on these determined boundary-nodes (LUNSB). Our simulation results indicate that LUNSB can effectively reduce network energy consumption and prolong the lifetime of the network without any location or directional information. Our future work includes analyzing the redundancy problem under unequal node distribution.

Acknowledgements

This work was supported under the National Science Foundation of China (50904070, 51104157);

REFERENCES

[1] Gao Y, Wu K, Li F. Analysis on the Redundancy of Wireless Sensor Networks[A]. WSNA'03[C]. New York: ACM Press, 2003.108-114

[2] Tian D, GEORGANAS N. A coverage-preserving node scheduling scheme for large wireless sensor networks[A]. Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Application[C].Atlanta, Georgia, USA, ACM Press,2002.32-41

[3] Z. Zhou, S. Das, and H. Gupta. Variable radii connected sensor cover in sensor networks. In Proc. of the IEEE Communications Society Conference on Sensors and Ad Hoc Comm. And Networks (SECON), September 2004

[4]F.Ye，Zhong, S. Lu, L. Zhang,. PEAS : A robust energy conserving protocol for long-lived sensor networks. in Proc. of the ACM MobiCom Conf, 2004, 129-143

[5]Hsin. C.F, & Liu,M. Netwrok coverage using low duty-cycled sensors: Random & coordinated sleep algorithms. In International Sympsium on Information Processing in Sensor Networks. 2004,433-442

[6]Jiang J, Li F, et al. Random scheduling for wireless sensor networks [A]. ISPA'09[C]. Sydney: IEEE CS Press, 2009:324-332

[7] KUI Wu, et al. Lightweight Deployment-Aware Scheduling for Wireless Sensor Networks[J]. Mobile Networks And Application, 2005, 10(6):837-852

[8] Ye F, Lu S, Zhang L. GRAdient Broadcast: A Robust Long-lived Large Sensor Network[J]. ACM Wireless Networks, 2005, 11(2):285-298

[9] H. Zhang, J.C. Hou. Maintaining sensing coverage and connectivity in large sensor networks[J]. Ad Hoc and Sensor Wireless Networks. 2005,1(1):89-124

[10] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Networked Sensors," *Proceedings of Ninth International Conference on Architectural Support for Porgramming Languages and Operating Systems*, November 2000, Cambridge, MA

***Authors**: Shan-shan Ma: lecturer, receiven the BS(2000), MS(2003), and is currently pursuing the Ph.D. degree at school of computer science and technology, China University of Mining and Technology. Email: ssma@cumt.edu.cn; Jian-sheng Qian: Prof. Dr. of China University of Mining and Technology, E-mail: qianjsh@cumt.eud.cn.*