Sleep Scheduling Method Based on Half-Sleep State in the Distributed Sensor Network

Pan Deng[1], Jianwei Zhang[2], Feng Chen[1], Jiafu Wan[3(✉)], Biying Yan[1], and Long Zhao[2]

[1] Lab. of Parallel Software and Computational Science,
Institute of Software Chinese Academy of Sciences, Beijing, China
{dengpan,chenfeng,biying}@iscas.ac.cn
[2] State Key Laboratory of Software Development Environment,
Beihang University, Beijing, China
hitzjw@163.com, zhaolong@nslde.buaa.edu.cn
[3] College of Electrical Engineering & Automation,
Jiangxi University of Science and Technology, Ganzhou, China
jiafuwan_76@163.com

Abstract. In order to extend the sensor's lifetime, this paper researched deeply into the sleep scheduling mechanism in the distributed sensor network. Now, the commonly used sleep scheduling methods based on the coverage have the problem of response delay, so we proposed a sleep scheduling method based on the half-sleep state to overcome this shortcoming. Under the control of regional agent node, this method adopts a minimum coverage algorithm based on the approximate solution to select the minimum coverage node set. Experimental results show that the proposed scheduling method can both reduce power consumption of the whole network effectively and extend the lifetime of the sensor noticeably.

Keywords: Sensor Network · Sleep Scheduling · Half-sleep

1 Introduction

In recent years, sensor networks and related techniques such internet of things and cyber-physical systems are developing very rapidly [1-3]. To save energy and make the lifetime of sensor longer, most of sensor network usually put part of sensor nodes in sleep state during the operation, whereas other sensor nodes which can cover the monitoring area keep in work [4-7]. The above mechanism is so-called sleep scheduling mechanism.

In order to realize the above sleep scheduling mechanism, it is necessary to decide which nodes should go to sleep and which should keep work. Now two kinds of methods usually are used to realize the scheduling. The first one is each sensor node go to sleep with probability p and keep work with probability 1-p. The strategy based on the probability is divide again into two kinds, which includes static probability and alterable probability. Static probability method means that each sensor node goes to

© Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2014
V.C.M. Leung et al. (Eds.): TridentCom 2014, LNICST 137, pp. 496–505, 2014.
DOI: 10.1007/978-3-319-13326-3_49

sleep with a predefined probability [4]. Alterable probability method can compute the probability of becoming a redundant node according to the perception of the nodes within its radius. This method can adjust dynamically the probability of becoming redundant, which has a lot of flexibility [8]. Beside the method based on the probability, another widely used method is to select some nodes which can cover the monitoring region, and at the same time close all the other nodes. Now, many sleep scheduling mechanism are based on this idea, for example the scheduling mechanism based on DELLC protocol [9], the two-phase sleep scheduling mechanism, and the dynamic sleep scheduling mechanism based on the pre-wakeup idea [6], and so on.

The above probability method will achieve the coverage region at a certain probability, so it is not used when the users need to realize the full coverage. The other method can achieve the full coverage, but after selecting the monitoring node, it will put all the other node go to sleep, and then these sleeping nodes will wakeup periodically to determine whether it will go to work. When an event occurs, the above mechanism will has some response delay, so we proposed a half-sleep scheduling mechanism to expend the lifetime of the sensor, and at the same time, to avoid the problem of the delay. Half-sleep state is a kind of sensor state, which refers to that the sensor under this state will close the data collection module, and only keep its communication module.

2 Related Definitions

Suppose all the sensors are placed in *R* which is a two-dimensional rectangle, and the coverage region of each sensor *s* is a circular (recorded as *C*(*s*)) with the center at *s* and the radius equal to *r*. If *S* is the sensor node set, the coverage region of *S* (recorded as *C*(*S*)) is the union of the coverage regions of all the sensors, i.e. $C(S) = \bigcup_{s \in S} C(s)$.

Definition 1: Suppose *R* is a region and *S* is a node set. If the coverage region of *S* can cover *R*, i.e. $R \subseteq C(S)$, the node set *S* is called the coverage set of region *R*.

Definition 2: For a given region *R* and a node set *S*, and S' is the subset of S. If S' is also a coverage set of *R*, and any proper subset of S' is not the coverage set of the region *R*, we then call S' as the least coverage set. And the least coverage set with the smallest node is called the minimum coverage.

Let's take Fig. 1 as an example. There are eleven sensors in that rectangle, and the monitoring area of each sensor is a circle with the radius equal to *r*. It is easy to see that in order to monitor the whole area, the three black sensors nodes are enough.

Definition 3: We call the node under the monitoring state as the monitoring node, and call the node which closes the monitoring module, under the half-sleep state as the half-sleep node. The half-sleep nodes keep half-sleep state for all the most time, and they will become into the monitoring node when they receive the wakeup message from the monitoring node and then open the monitoring module.

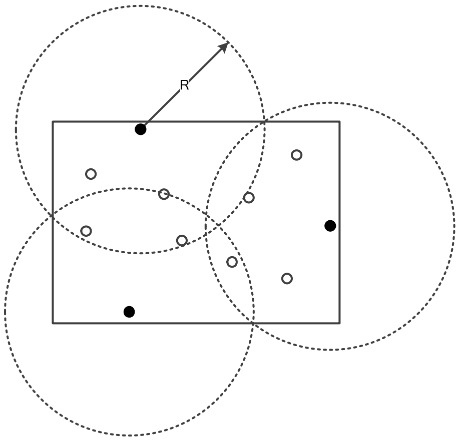


Fig. 1. Minimum coverage

3 Half-Sleep Scheduling Mechanism

This paper used the approximate solution to realize the half-sleep schedule which has the following goodness: on the one hand using the approximate solution can control the time complexity under the polynomial magnitude; on the other hand, the number of the solution of the least coverage set is larger than that of the minimum coverage set, the difference between the number can make the half-sleep schedule more fairly.

3.1 Basic Ideas

Region agent node will periodically send wakeup or sleep message to the ordinary node in the network to decide the state of the ordinary node. Once the ordinary node receives the message, it will determine whether it needs to open its data collection module. If the node will go to sleep according to the message, it should send the collective data to the certain nodes before it go to sleep.

The whole process of the sleep scheduling mechanism includes the following four steps:

Step1: set the random factor (the setting method of which will be introduced in the following section), the aim of which is to make all the nodes go to sleep relatively fairly.

Step2: according to the above random factor, using the minimum coverage algorithm to get an approximate solution to create the monitoring nodes set;

Step3: traverses each node in the coverage set, and sends the wakeup message to each node until receiving all the answers from each node.

Step4: send sleep message to all the nodes which are not selected as the new monitoring node. The process of this sleep scheduling mechanism is over.

During the process, if it can’t receive the answer from any of the nodes in Step3, this process will be pronounced a failure, and then begin a new scheduling process after deleting the node which don’t send back the answer.

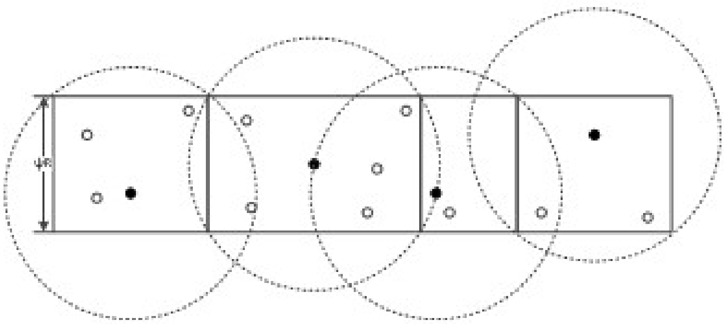
3.2 Minimum Coverage Algorithm Based on MRC

In order to find out the least coverage set of the monitoring nodes, we divide the wakeup process to the following three steps:

(1) According to the random factor φ $(\frac{\sqrt{2}}{2} \leq \varphi < 1)$, the process firstly divides the whole network into several rectangles, the width of which is φR .

(2) Secondly, find the least coverage set for every rectangle.

(3) Finally, merge all the least coverage sets of each rectangle to achieve the coverage set of the whole network.



● candidate detection node

○ candidate sleep node

Fig. 2. Rectangle division

Fig. 2 gives an instance of the rectangle division based on the random factor, where the black node represent the candidate nodes under the monitoring state. The algorithm to find a least coverage set for the rectangle is as follows:

(1) Suppose the node set in the rectangle is BS = {bs1, bs_2... bs_m}, and all the monitoring radii of each node are R;

(2) For the circle arc the node bs_i coverage, its right half part will intersect the rectangle at two pointes (proved by the following **Theorem 1**), and record the smaller X-coordinate value of the one of the two point as $x_{right}(bs_i)$;

(3) Begin from x_{left} which lies in the leftmost of the rectangle to find two nodes which can cover the left part of the rectangle, and record the node with the largest $x_{right}(bs_i)$ as bs_i;

(4) Set x_{left} to $x_{right}(bs_i)$, and put bs_i into the least coverage set; continue the process until $x_{left} \geq x_{right}$;
(5) Return the node set, and the algorithm is end.

The above algorithm cannot always get the optimum solution, but there isn't the big difference between the achieved solutions and the optimum solution because the value of random factor φ is limited to $\frac{\sqrt{2}}{2}$ to 1. This conclusion can be proved by **Theorem 2**.

Theorem 1: When the value of the random factor φ is smaller than 1, the right part of the cover circle of each node within the rectangle will intersect with the rectangle at two points.

Theorem Proving: The prerequisite of a circle and a line not intersecting is the distance from the circle center to the line is smaller than the radius of the circle. The width of the rectangle is $b = \varphi R$, so for every node in the rectangle, the distances from the node to the upper side of the rectangle or to the bottom side of the rectangle will satisfy with the relation: $d \leq b$; Circle center is also a node within the rectangle, so the distances from the circle center to the upper side or to the bottom size are $d_c \leq b$. So it can be deduced that $d_c \leq b = \varphi R < R$. So the coverage circle of the node in the rectangle is sure to intersect with the upper side and the bottom size at the same time. The proving is over.

Theorem 2: When the value of the random factor φ ranges from $\frac{\sqrt{2}}{2}$ to 1, the ratio between the appropriate resolutions set size and the optimal resolutions set size will not more than 2.

Theorem Proving: Suppose the achieved coverage network under the optimal resolutions is {g1, g_2... g_m}, the set size achieved under the appropriate resolutions is $|MRC|$, and the set size achieved under the optimal resolutions is $|OPT|$. Now we use the induction to prove the theorem, which as follows: when m = 1, the optimal resolutions use one node (g_1) to cover the monitoring region. In the worst case, if the nodes selected under the appropriate resolutions locate in the edge of the network, it will need two nodes at most, so $|MRC_1| \leq 2 = 2|OPT_1|$. Now, we begin to induce, suppose we will have $|MRC_k| \leq 2|OPT_k|$ when $m = k$. When $m = k + 1$, g_{k+1} will need at most two nodes to cover, so we will have $|MRC_{k+1}| \leq |MRC_k| + 2 \leq 2|OPT_k| + 2 \leq 2|OPT_{k+1}|$, i.e. $|MRC_{k+1}| \leq 2|OPT_{k+1}|$. So the ratio between the appropriate resolutions set size and the optimal resolutions set size will not more than 2. The proving is over.

4 Experimental Results and Analysis

4.1 Analysis of Scheduling Results

Fig. 3 is a 10×10 region, and within which there are 200 randomly-generated nodes, and each node is supposed to represent a sensor with the monitoring radius is 2. Now, we adopt the minimum coverage algorithm to select a minimum coverage set.

The algorithm finally selected 20 nodes from the 200 nodes, and the coverage is 100% based on the Monte Carlo method, which can satisfy the monitor need.

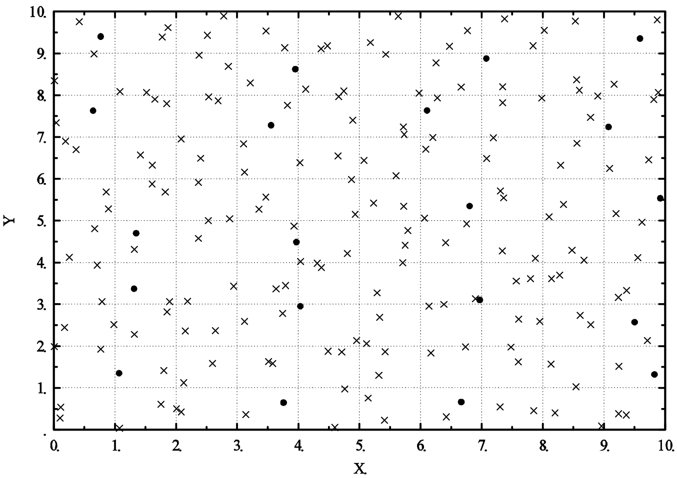


Fig. 3. Diagram of scheduling results of half-sleep scheduling

4.2 Analysis of System Power Consumption

The power consumption of the sensor consists of two parts, which are message handling consumption *m* and data collection consumption *d*. In order to test the saving of the system power consumption under the half-sleep scheduling mechanism, we adopt the energy-saving coefficient *r*, which is the ratio between the energy-saving under the sleep state with the total consumption when all the sensors are wakeup. The bigger the coefficient *r*, the lower the system power consumes. The coefficient *r* is computed as follows:

$$r = \frac{(N-n)*d}{N*(m+d)} = (1-\frac{n}{N})*\frac{d}{m+d} \quad (1)$$

where *N* is the total number of the sensors, *n* is the monitoring nodes the above minimum coverage algorithm select to keep work, and all the other node beyond the *n* nodes are all go to sleep. *d*/(*m*+*d*) is a constant predefined according to the specific monitoring equipment, so the finally value of the coefficient r is determined by the number of nodes in the half-sleep state, i.e. $r' = 1 - n/N$.

Suppose we need to monitor a 100×100 region, the following tests analyze the system power consumption from two aspects, which are the monitoring radius R of each sensor and the total number SN of the sensors.

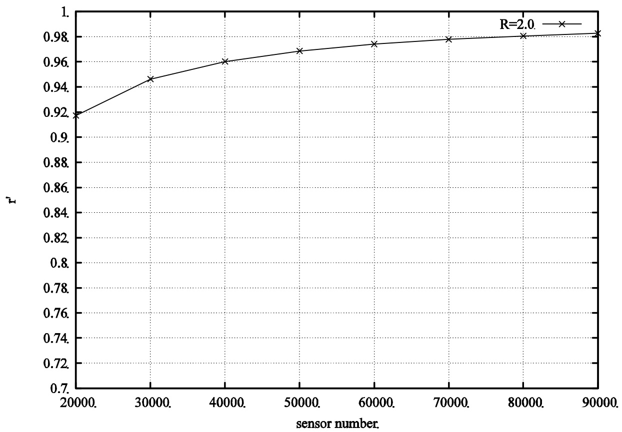


Fig. 4. Relations between energy saving coefficient and sensors number

(1) The affect of the total number of the sensors (R = 2.0)
The curve in Fig. 4 represents the change of the coefficient along with the increase of the total number of the sensors (from 20000 to 90000), where the monitoring radius is 2.0. From the curve we can see that the system achieves good energy saving effect along with the increase of the total number. In a certain static region, when the monitoring radius is set down, the number of the sensors need to monitor this region is also in a relatively fixed range. So, with the increase of the total number, the system energy saving coefficient will increase at a speed of 1 - 1/x.

(2) The affect of the monitoring radius (SN = 20000)
The curve in Fig. 5 represents the change of the coefficient along with the increase of the monitoring radius (from 2 to 9), where the total number of sensors is set to 20000. Similar with the curves in Fig. 4, the system achieves good energy saving effect along with the increase of monitoring radius. The curve in Fig. 5 is steeper than that in the Fig. 4, which is because that the relation between the monitoring area and the radius is a kind of square relation.

In the main, we can increase the energy-saving coefficient through either increasing the total number of the sensors or increasing the monitoring radius. But increasing the total number of the sensor will increase the total power consumption, so in the actual environment, we should make an effort to increase the monitoring area of each sensor to decrease the system power consumption.

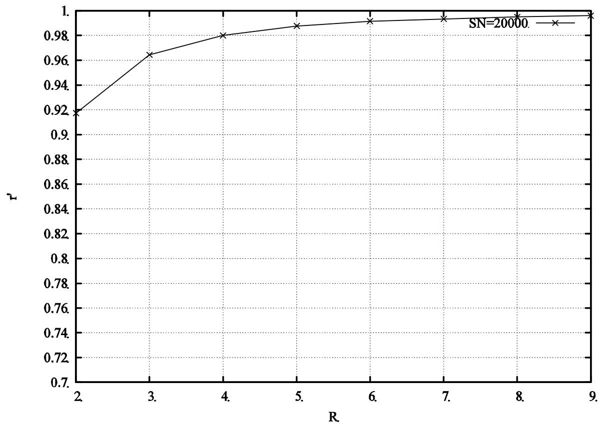


Fig. 5. Relations between energy saving coefficient and monitoring radius

4.3 Analysis of Response Time of the Half-Sleep Scheduling

Compared to the traditional sleep scheduling mechanism, the half-sleep scheduling mechanism has an advantage in response time. In order to verify this, we now give the simulate tests about the response time, where include 1,000 nodes. Fig. 6 gives the result of the traditional sleep scheduling mechanism. In the simulate tests, each node works for 10 seconds, and then go to sleep for *stime* second. The curve in Fig. 6 is got with the value of *stime* range from 1000 to 9000. At the same time, we do another nine tests about the time need to wake up a sensor when it is under the half-sleep state, and the result is shown in Fig. 7. From the above figures we can see that in the traditional

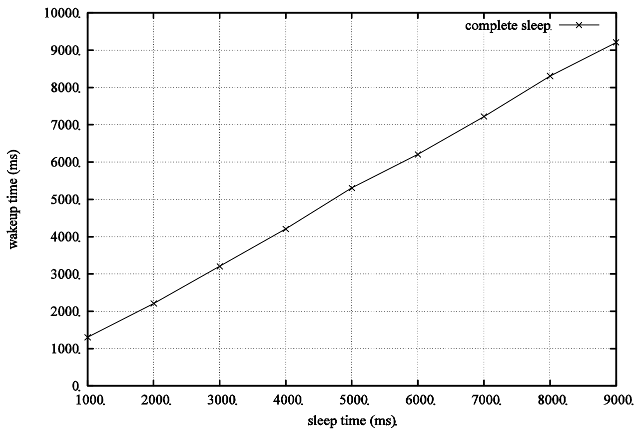


Fig. 6. Wakeup response time of traditional sleep mode

sleep scheduling mechanism, when wakeup event occurs, the response time is basically proportional to the time period of the sensor under sleep state plus certain network delay. But, in these nine tests, the response time of the proposed half-sleep scheduling mechanism is always around 300ms, which is just the network delay. From this comparison, we can find that the proposed half-sleep scheduling mechanism has the absolute advantage in the response time.

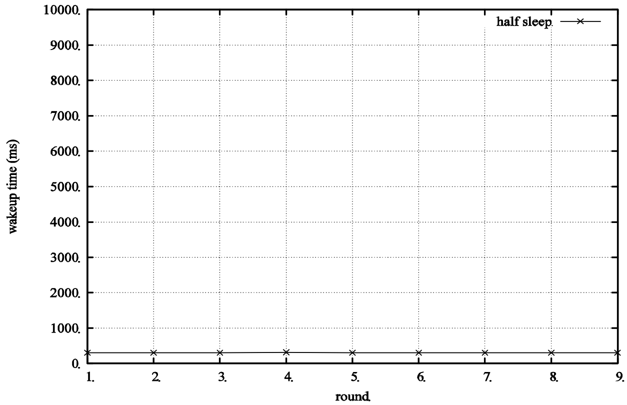


Fig. 7. Wakeup response time of half-sleep mode

5 Conclusions

Most of the sensor networks usually adopt sleep scheduling mechanism, but traditional method has the shortcoming of response time delay. In order to extend the sensor's lifetime, this paper researched deeply into the sleep scheduling mechanism in the distributed sensor network. Now, the commonly used sleep scheduling methods based on the coverage have the problem of response delay, so we proposed a sleep scheduling method based on the half-sleep state to overcome this shortcoming. Under the control of regional agent node, this method adopts a minimum coverage algorithm based on the approximate solution to select the minimum coverage node set. Experimental results show that the proposed scheduling method can both reduce power consumption of the whole network effectively and extend the lifetime of the sensor noticeably.

Acknowledgements. The work was supported by the National Natural Science Foundation of China (No. 61100066, 61262013).

References

1. Zhang, J., Deng, P., Wan, J., Yan, B., Rong, X., Chen, F.: A Novel Multimedia Device Ability Matching Technique for Ubiquitous Computing Environments. EURASIP Journal on Wireless Communications and Networking 2013, 181 (2013), doi: 10.1186/1687-1499-2013–181

2. Wan, J., Chen, M., Xia, F., Li, D., Zhou, K.: From Machine-to-Machine Communications towards Cyber-Physical Systems. Computer Science and Information Systems **10**(3), 1105–1128 (2013)
3. Wan, J., Zhang, D., Sun, Y., Lin, K., Zou, C., Cai, H.: VCMIA: A Novel Architecture for Integrating Vehicular Cyber-Physical Systems and Mobile Cloud Computing. ACM/Springer Mobile Networks and Applications **19**(2), 153–160 (2014)
4. Ren, Q., Li, J., Gao, H., Cheng, S.: A Two-Phase Sleep Scheduling Based Protocol for Target Tracking in Sensor Networks. Chinese Journal of Computers **32**(10), 1971–1979 (2009)
5. Shi, G., Liao, M.: Stochastic Sleeping for Energy-Conserving in Large Wireless Sensor Networks. Journal of Computer Research and Development **43**(4), 579–585 (2006)
6. Liu, Y., Wu, J., Chen, Z., Xiong, Z.: A Dynamic Sleep Scheduling Mechanism for Localization in Mobile Sensor Networks. Journal of Computer Research and Development **45**(8), 1330–1337 (2008)
7. Zhu, J., Li, J., Liu, Y., Gao, H.: Data-Driven Sleeping Scheduling Mechanism in Sensor Networks. Journal of Computer Research and Development **45**(1), 172–179 (2008)
8. Liu, M., Cao, J., Zheng, Y., Chen, L., Xie, L.: Analysis for Multi-Coverage Problem in Wireless Sensor Networks. Journal of Software **18**(1), 127–136 (2007)
9. Mao, Y., Liu, M., Chen, L., Chen, D., Xie, L.: A Distributed Energy-Efficient Location-Independent Coverage Protocol in Wireless Sensor Networks. Journal of Computer Research and Development **43**(2), 187–195 (2006)