A Lightweight Scheme for Node Scheduling in Wireless Sensor Networks

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

[1] School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu 610054, P.R. China
wing.lm@gmail.com
[2] Internet and Mobile Computing Lab, Department of Computing, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, P.R. China
[3] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

Abstract. The coverage problem in wireless sensor networks (WSNs) is to determine the number of active sensor nodes needed to cover the sensing area. The purpose is to extend the lifetime of the WSN by turning off redundant nodes. In this paper, we propose a mathematical method for calculating the coverage fraction in WSNs. According to the method, each active node can evaluate its sensing area whether covered by its active neighbors. We assume that the network is sufficiently dense and the deployed nodes can cover the whole monitored area. In this scenario, if a node's sensing area is covered by its active neighbor nodes, it can be treated as a redundant node. Based on this idea, we propose a lightweight node scheduling (LNS) algorithm that prolongs the network lifetime of the sensor network by turning off redundant nodes without using location information. Simulation study shows that LNS scheme can save considerable energy for data gathering while meeting the desired coverage fraction imposed by application.

1 Introduction

Technology advances in sensors, embedded systems, and low power-consumption wireless communications have made it possible to manufacture tiny wireless sensor nodes with sensing, processing, and wireless communication capabilities. The low-cost and low power-consumption sensor nodes can be deployed to work together to form a wireless sensor network. The sensor nodes in a sensor network are able to sense surrounding environment, carry out simple processing tasks, and communicate with the neighboring nodes within its transmission range. By means of the collaborating among sensor nodes, the sensed and monitored environment information (e.g. temperature, humidity) is transmitted to the base station for future processing. A large-scale wireless sensor network can consist of tens of thousands of tiny sensor nodes, with high density of sensors (up to 20 nodes/ m^3 [1]). The high density of sensors may result in comparatively large energy consumption due to conflicting in

J. Indulska et al. (Eds.): UIC 2007, LNCS 4611, pp. 579–588, 2007.

accessing the communication channels, maintaining information about neighboring nodes, and other factors. A widely-used strategy for reducing energy consumption while at the same time meeting the coverage requirement is to turn off redundant sensors by scheduling sensor nodes to work alternatively [2-6]. The coverage problem in wireless sensor networks (WSNs) is to determine the number of active sensor nodes needed to cover the sensing area.

A widely used strategy [7-11] is to determine the active nodes by using the location information about the sensor nodes and their neighborhood. However, relying on complicated hardware equipments such as the GPS (Global Positioning System) or the directional antenna will greatly increase the hardware cost of sensor nodes and energy consumption. At the same time, the message transmission and calculation of the locations and directions will also consume the energy of a node. Therefore, it is desirable for a solution to the coverage problem that independs on location information.

Most applications may not require the full area coverage and can tolerance a small part of uncovered area at certain intervals. If the active nodes in a sensor network can maintain reasonable area coverage, the requirements of most applications can be met. Coverage can be regarded as one metric of the quality of service of a sensor network that evaluates its monitoring capability [12]. If the coverage fraction is below a certain threshold, the sensor network cannot work normally. propose a simple method that can, in a statistical sense, evaluate the coverage fraction which meets the coverage requirement of the application without location information. In this paper, we propose a mathematical method for calculating coverage fraction in WSNs without the use of location information and then propose a lightweight node scheduling algorithm. Unlike most existed solutions [2, 9-11] which focus on providing full coverage, the proposed scheme gives a solution for fractional coverage. As achieving full coverage can be very costly and energy consuming, fractional coverage is an energy efficient approach which meets the requirements of many applications such as the ones in [5] [13]. Moreover, our work is more general when comparing to those ones using theoretical methods for coverage analysis without using location information, which is a special case of our work.

The rest of the paper is organized as follows. In section 2, we present sensor network models and preliminary definitions. In section 3, we propose a mathematical model for calculating the coverage fraction in a random deployed network. In section 4, a lightweight node scheduling algorithm is proposed based on the above mathematical model. Simulation results are provided and discussed in section 5. Section 6 concludes the paper.

2 Preliminary

In this paper, we use $p(x, y)$ to represent the location of node v_i, and use $v_i(x, y)$ to represent an arbitrary point in v_i's communication range. The communication range of v_i is denoted by $C(v_i)$, and the sensing range of v_i is denoted by $S(v_i)$. To facilitate the discussion, we introduce following definitions:

Definition 1. Neighboring area $\aleph(v_i(x, y))$. For an arbitrary point in node v_i's communication range, its neighboring area is defined as

$$\aleph(v_i(x, y)) = \left\{\forall(x, y) \in C(v_i) \middle| \left((x - v_i(x))^2 + (y - v_i(y))^2 \le r^2\right)\right\} \tag{1}$$

Definition 2. Centre area $C'(v_i)$. The centre area of node v_i is defined as

$$C'(v_i) = \left\{\forall(x, y) \middle| \left((x - p(x))^2 + (y - p(y))^2\right) \le (R - r)^2\right\} \tag{2}$$

According to the above definition, we use $C(v_i) - C'(v_i)$ to indicate non-centre area.

Definition 3. Desired coverage fraction Q. The desired coverage fraction indicates the ratio of the area covered by all active nodes to the entire monitored area. It can be defined as

$$Q = \frac{\left|\bigcup S(v_i) \cap A\right|}{|A|} \tag{3}$$

Intuitively, for a sensor node v_i, if the fraction of $C(v_i)$ covered by its active neighbor nodes equal to the desired coverage fraction Q imposed by the application, node v_i can be regarded as a redundant node; otherwise, node v_i should keep working until it has enough active neighbor nodes. Therefore, the goal of our proposed mathematical method is to achieve the desired coverage fraction with a minimum number of active nodes. As this method indicates the relationship between the desired coverage fraction and the minimum number of active nodes, each active node can calculate the number of active neighbor nodes needed for the desired coverage fraction, and then determine its working or sleep status.

3 Fractional Coverage Analyses for Dense Wireless Sensor Networks

For an arbitrary point $v_i(x, y)$ within the communication range of node v_i, it is covered if there exists at least one sensor node deployed in its neighboring area $\aleph(v_i(x, y))$. We assume that all sensor nodes are randomly deployed in the entire monitored area with the uniform distribution. As a result, for a sensor node v_i, the distribution of its neighbor nodes in the communication range of v_i can also approximate to the uniform distribution. Therefore, the probability that a neighbor sensor node of v_i is deployed in the neighboring area of point $v_i(x, y)$, say q, equals to $\frac{\left|\aleph(v_i(x, y))\right|}{\left|C(v_i)\right|}$. The neighboring area of node v_i may have two cases. If the point $v_i(x, y)$ locates in $C^{'}(v_i)$, the size of neighboring area $\left|\aleph(v_i(x, y))\right|$ equals to πr^2 ; otherwise, if the point $v_i(x, y)$ lies in $C(v_i) - C^{'}(v_i)$, the size $\left|\aleph(v_i(x, y))\right| \le \pi r^2$.

Since the distribution of nodes can be regarded as a Poisson process in the region $C(v_i)$, the probability that an arbitrary point $v_i(x, y)$ is covered equals to the probability that at least one sensor falls in its neighboring area, we can easily obtain the probability of $v_i(x, y)$ being covered as follows:

$$\begin{aligned} p_c(v_i(x, y)) &= C_n^1 q(1-q)^{n-1} + C_n^2 q^2 (1-q)^{n-2} + \cdots\cdots + C_n^n q^n \\ &= 1 - (1-q)^n \end{aligned} \tag{4}$$

According to Eq. (4), for any two points in $C(v_i)$, if the sizes of their neighboring area are equal, the probabilities of being covered are also equal. Obviously, for each point within the communication range of node v_i, if the distance between this point and the node v_i is identical, the size of its neighboring area is also identical, i.e., the probability of being covered is identical. For each point that lies within the marginal region $C(v_i) - C^{'}(v_i)$, the size of its neighboring area is less than πr^2 . Especially, when the point lies on the edge of C, its neighboring area is the smallest one, and thus its probability of being covered is also the smallest one. For $\forall v_i(x, y) \in C^{'}(v_i)$, the neighboring area of node v_i is equal to πr^2 . Therefore, the probability of an arbitrary node deployed in the neighboring area of $v_i(x, y)$ $(v_i(x, y) \in C^{'}(v_i))$ equals to $\frac{r^2}{R^2}$. If $R >> r$, the average size of the neighboring areas of all points within the communication range of node v_i can approximate to πr^2 . According to [13], the desired coverage fraction can be calculated by the following equation:

$$E_{cf} = 1 - e^{-\lambda \pi r^2} \tag{5}$$

According to Eq. (4), a designer can use Eq. (5) to approximately calculate at which node density a large-scale sensor network can meet the desired coverage fraction. If the sensing range r is long enough compared to the communication rang R, the border effect cannot be ignored. In such a case, using Eq. (5) to calculate the desired coverage fraction will lead to a deviation beyond tolerance. In order to have an accurate estimation on the relationship between the minimum number of active nodes and the desired coverage fraction, we have to calculate the average size of the neighboring areas of all points within the communication range of node v_i .As shown in Fig.1, the point $v_i(x, y)$ lies in $C(v_i) - C^{'}(v_i)$, and we use l to denote the distance between the point $v_i(x, y)$ and location of node v_i. Obviously, $l > R - r$. According to the Definition 1, the point $v_i(x, y)$'s neighboring area is the shadowed region shown in Fig.1. We can easily obtain the value $\left|\aleph(v_i(x, y))\right|$ by the equation as follows:

$$\left|\aleph\left(\forall v_i(x, y)\middle|\{(x, y) \in (C(v_i) - C^{'}(v_i))\}\right)\right| = 2\left(\int_{l-r}^{\frac{R^2-r^2+l^2}{2l}} \sqrt{r^2 - (y-l)^2}\,dy + \int_{\frac{R^2-r^2+l^2}{2l}}^{R} \sqrt{R^2 -}\right.$$

$$= \frac{1}{2}\pi\left(r^2 + R^2\right) + r^2 \arcsin\frac{R^2 - r^2 - l^2}{2l}\sqrt{r^2 - \frac{\left(R^2 - r^2 - l^2\right)^2}{4l^2}}$$

$$- R^2 \arcsin\frac{R^2 - r^2 + l^2}{2lR} - \frac{R^2 - r^2 + l^2}{2l}\sqrt{R^2 - \frac{\left(R^2 - r^2 + l^2\right)^2}{4l^2}} \tag{6}$$

Eq. (6) is a general expression for calculating the neighboring area of an arbitrary point lied in $C(v_i) - C^{'}(v_i)$. According to Eq. (6), for an arbitrary point lied in $C(v_i) - C^{'}(v_i)$, the size of neighboring area is only determined by l. Observing $Eq.6$, for the average size of neighboring area of all points in $C(v_i) - C^{'}(v_i)$, we have

$$\left|\aleph(v_i(x, y))^{C(v_i)-C^{'}(v_i)}\right| = \frac{\iint_{C(v_i)-C^{'}(v_i)} \aleph(v_i(x, y))\,d\sigma}{\left|C(v_i) - C^{'}(v_i)\right|}, \tag{7}$$

where we use $\aleph(v_i(x, y))^{C(v_i)-C^{'}(v_i)}$ to denote the average size of neighboring area of all points in $C(v_i) - C^{'}(v_i)$.

Hence, the average size of neighboring area of all points in $C(v_i)$ can be calculated by the following equation:

$$\left|\aleph(v_i(x, y))^{C(v_i)}\right| = \frac{\left|\aleph(v_i(x, y))^{C(v_i)-C^{'}(v_i)}\right|\left(R^2 - (R-r)^2\right) + r^2(R-r)^2}{R^2} \tag{8}$$

Observing Eq. (4) and Eq. (8), despite that the border effect cannot be ignored, it is reasonable to consider that the probability of each point in $C(v_i)$ being covered is equal to $1 - (1 - \frac{\left|\aleph(v_i(x, y))^{C(v_i)}\right|}{\left|C(v_i)\right|})^n$, where n denotes the number of selected active neighbor sensor nodes of v_i . Therefore, the expectation of the covered region of $C(v_i)$ is given by

$$E_{the\ size\ of\ covered\ region\ of\ C(v_i)} = \iint_{C(v_i)} \left(1 - \left(1 - \frac{\left|\aleph(v_i(x,y))^{C(v_i)}\right|}{C(v_i)}\right)^n\right) d\sigma$$
$$= \pi R^2 \left(1 - (1 - \frac{\left|\aleph(v_i(x,y))^{C(v_i)}\right|}{|C(v_i)|})^n\right) \tag{9}$$

Obviously, the expectation of coverage fraction is given by

$$E_{the\ fraction\ of C(v_i)\ being\ covered} = 1 - (1 - \frac{\left|\aleph(v_i(x,y))^{C(v_i)}\right|}{|C(v_i)|})^n \tag{10}$$

Given the desired coverage fraction, we can derive from Eq. (10) the minimum number of active neighbor sensor nodes that ensures the desired coverage fraction even when node v_i itself turns off. Assume that the number of active neighbor nodes is K . For any node v_i , if there are more than K neighbor sensor nodes within $C(v_i)$, the desired coverage fraction of the entire monitored area can be guaranteed. Based on this idea, we propose a lightweight node scheduling scheme (LNS) in the next section.

4 Lightweight Node Scheduling Scheme (LNS)

We have known from the previous section that given a desired coverage fraction, the required number of active neighbor nodes can also be calculated. This information can be pre-set in each node before deployment if the desired coverage fraction is fixed in the entire lifetime of the network, or it can be broadcasted by a base station.

LNS is a local distributed algorithm, in which each sensor node periodically broadcasts the $Working\_Msg$ to its active neighbor nodes with a static interval time WT . If the number of its active nodes is larger than the analytical result K , it broadcasts the $Sleep\_Msg$ to all its neighbor nodes. Before a sensor turning off itself, it will back-off a random time $RT(RT_{\min} < RT < RT_{\max})$. Using the back-off strategy, the big sensing hole can be avoided by reducing the probability that several neighbor nodes sleep simultaneously [2]. However, if the node perceives that the number of its active nodes is smaller than the threshold K in the duration of back-off time, it will keep working until it has enough active neighbor nodes. Hence, we have $WT < RT_{\min}$.

Here we set $RT_{\min} = 2WT$ to guarantee the $Working\_Msg$ of all active neighbor nodes can be received in duration RT .In order to perform well, each node needs to maintain a neighborhood table to store the information about its neighbors such as node's ID and residual energy. The ID indicates the unique identification of the neighbor nodes. Without losing generality, we use an integer value to label a node's identification like TinyOS [14]. After a node v has received the

```
// Each node executes this process after deployment
while state = working
  if have enough neighbors
    state ⟵ back-off
    break
  endif
  if ((current time – starting time) mod WT) =0
    broadcast Working_Msg
  endif
  if received Msg from neighbor NT[i]
    update NT[i] according to the content of Msg
  endif
endwhile
while state = back-off
  if Timer of RT has not expired
    if received Msg from neighbor NT[i]
      update NT[i] according to the content of Msg
    endif
  else
    if have enough neighbors then
      state ⟵ sleeping
      broadcasting Sleeping_Msg
    else
      state ⟵ working
    endif
  endif
endwhile
while state = sleeping
  if timer of sleeping time expired
    state ⟵ working
    broadcast Working_Msg
  endif
endwhile
```

Fig. 1. Pseudo code of node scheduling

$Working\_Msg$ or $Sleep\_Msg$, v will update its neighborhood table according to the content of message. For example, if v receives the $Working\_Msg$ from its neighbor node which is labeled with sleeping state in its neighborhood table, v will update the neighbor node's state as working state and record the start working time of this neighbor in v's neighborhood table. For another example, if v receives the $Sleep\_Msg$ from a neighbor, according to the content of the $Sleep\_Msg$, v will record the starting time for back-off and the duration of back-off time RT of this neighbor in its neighborhood table, and change the state of this neighbor as back-off. After $RT+WT$ time, if v does not receive the $Working\_Msg$ from this neighbor, it will update the state of this neighbor as sleeping; otherwise, it will update the state as working. Before falling asleep, v will reset its neighborhood table. After falling asleep for a period of time ST (sleeping time), v will wakeup and update its neighborhood table as mentioned above. In order to prolong the network lifetime, we use the residual energy as a prime parameter for computing the random back-off time RT . Therefore, the more residual energy a node has the higher chance the node will work longer. The pseudo code of LNS is shown in Fig.1.

5 Simulation Study

5.1 Simulation Parameters

In the simulation experiments, we mainly focus on evaluating the effectiveness of LNS, and the relationship between the number of active nodes and the desired coverage fraction. The evaluation of other aspects of LNS will be conducted in future work. The simulation parameters are identical with [15]. Unless otherwise specified, every simulation result shown below is the average of 200 independent experiments where each experiment uses a random seed to generate a uniform topology of the sensor network. For simplicity, we assume that the wireless channel is ideal that the probability of signal collision and interference in the channel is ignorable. We also assume that the radio transmitter and radio amplifier are the main energy consumers of a sensor node. Furthermore, we assume that sleeping energy consumption is zero. In the simulation, we use the same radio model shown in [15, 16] for the radio hardware energy dissipation. We argue that such assumptions can not drastically impact the simulation results, although are not always true in a real system.

5.2 Simulation Results

a. Effectiveness of LNS

In this subsection, we have evaluated the effectiveness of LNS. As shown in Fig.2, LNS can effectively meet the desired coverage fraction required by applications. However, LNS cannot provide a perfect matching between the real coverage fraction and the desired coverage fraction, because LNS is an algorithm that schedules nodes in a statistical way. Note that providing the perfect desired coverage fraction is very difficult or even unrealistic in a real application.

b. Relationship between the number of active nodes and the desired coverage fraction

As shown in Fig.3, no matter what the density of network is, the actual number of active nodes is only determined by the desired coverage fraction after executing LNS. This feature is similar to other scheduling algorithms [8-10], but LNS can achieve the

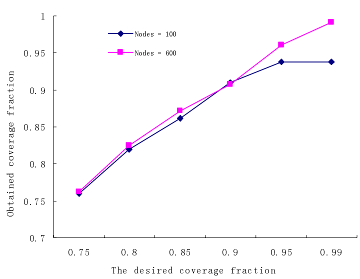

Fig. 2. Effectiveness of LNS

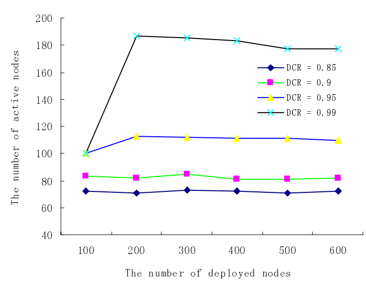

Fig. 3. The relationship between the number of active nodes and desired coverage fraction

desired coverage fraction without using any location information. However, if the number of deployed nodes cannot provide the desired coverage fraction even when all nodes are in active, LNS plays no roles. Fig.3 shows that LNS cannot provide 95% coverage fraction, when the number of deployed nodes equals to 100. Fig.3 also demonstrates that higher desired coverage fraction requirements need more active nodes, and thus more energy will be consumed. For example, if the 99% coverage fraction is required, the number of active nodes should be more than 180 after executing LNS. Therefore, it is a very important issue in node scheduling algorithms that how to achieve a better tradeoff between the desired coverage fraction and the lifetime of network according to the application requirements.

6 Conclusion

In this paper, we propose a lightweight node scheduling algorithm that has several nice features. LNS is very simple and totally localized and distributed, so it can apply in a large scale sensor network. The performance of LNS is independent of the location information of the sensor node. As a result, it can not only save considerably energy for obtaining and maintaining the location information, but also reduce the cost of sensor node. According to the desired coverage fraction required by application, LNS can dynamically adjust the density of active sensor nodes so that it will significant prolong the network lifetime. Furthermore, LNS neither requires the synchronization of sensor nodes, nor does it require the operation in a round-by-round fashion. Therefore, LNS can easily apply to a real application.

Acknowledgement

The work is partly supported by the Hong Kong Polytechnic University under the ICRG grant A-PF77, China NSF grants (60573131, 60673154) Jiangsu Provincial NSF grant (BK2005208), and China 973 projects (2006CB303004, 2002CB312002). The Conference Participation is supported by Nokia Bridging the World Program.

References

1. Shih, E., Cho, S., Ickes, N., Min, R., Sinha, A., Wang, A., Chandrakasan, A.: Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In: proceedings of the seventh annual international conference on Mobile computing and networking (MobiCom 01), Italy, pp. 272–287 (2001)
2. Tian, D., Georganas, N.D.: A Coverage-preserved Node Scheduling scheme for Large Wireless Sensor Networks. In: Proceedings of First International Workshop on Wireless Sensor Networks and Applications (WSNA'02), Atlanta, USA, pp. 32–41 (2002)
3. Liu, C., Wu, K., Xiao, Y., Sun, B.: Random Coverage with Guaranteed Connectivity: Joint Scheduling for Wireless Sensor Networks. IEEE Transactions on Parallel and Distributed Systems (TPDS) (to appear)

4. Ye, F., Zhong, G., Lu, S., Zhang, L.: PEAS: A robust energy conserving protocol for long-lived sensor networks. In: Proceedings of the 23nd International Conference on Distributed Computing Systems (2003)
5. Wu, K., Gao, Y., Li, F., Xiao, Y.: Lightweight Deployment-Aware Scheduling for Wireless Sensor Networks. ACM/Kluwer MONET Journal, Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks 10(6), 837–852 (2005)
6. Hsin, C., Liu, M.: Network coverage using low duty-cycled sensors: Random & coordinated sleep algorithm. In: Proceedings of the Third International Symposium On Information Processin In Sensor Networks (IPSN 2004) (2004)
7. Slijepcevic, S., Potkonjak, M.: Power efficient organization of wireless sensor networks. In: Glisic, S. (ed.) Proc. of the IEEE Conf. on Communications, pp. 472–476. IEEE Press, Helsinki (2001)
8. Cardei, M., MarCallum, D., Cheng, X., Min, M., Jia, X., Li, D., Du, D.: Wireless sensor networks with energy efficient organization. Journal of Interconnection Networks 213–229 (2002)
9. Xu, Y., Heidemann, J., Estrin, D.: Geography-Informed energy conservation for ad hoc routing. In: Proc. of the 7th Annual ACM/IEEE Int'l Conf. on Mobile Computing and Networking, pp. 70–84. ACM Press, New York (2001)
10. Zhang, H., Hou, J.C.: Maintaining scheme coverage and connectivity in large sensor networks. Wireless Ad Hoc and Sensor Networks: An International Journal 89–123 (2005)
11. Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., Gill, C.D.: Integrated coverage and connectivity configuration in wireless sensor networks. In: Proc. of the 1st Int'l Conf. on Embedded Networked Sensor Systems, pp. 28–39. ACM Press, New York (2003)
12. Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.: Coverage problems in wireless ad-hoc sensor networks. In: Proc of the IEEE INFOCOM. Anchorage, pp. 1380–1387. IEEE Press, Anchorage (2001)
13. Liu, B., Towsley, D.A.: study on the coverage of large-scale sensor networks. In: Proc. of the 1st IEEE Int'l Conf. on Mobile Ad-Hoc and Sensor Systems, IEEE Press, Los Alamitos (2004)
14. Culler, D.E., Hill, J., Buonadonna, P., Szewczyk, R., Woo, A.: A Network-Centric Approach to Embedded Software for Tiny Devices. In: Henzinger, T.A., Kirsch, C.M. (eds.) EMSOFT 2001. LNCS, vol. 2211, pp. 114–130. Springer, Heidelberg (2001)
15. Heinzelman, W.R.: An application-specific protocol architecture for wireless microsensor networks. IEEE Trans. on Wireless Communications 1(4), 660–670 (2002)
16. Younis, O., Fahmy, S.: Distributed clustering in Ad-hoc sensor networks: A hybrid, energy-efficient approach. In: Proc. of the IEEE INFOCOM, IEEE Computer Society Press, Los Alamitos (2004)