A Distributed Active Sensor Selection Scheme for Wireless Sensor Networks*

Kuei-Ping Shih, Yen-Da Chen, Chun-Wei Chiang, and Bo-Jun Liu
Department of Computer Science and Information Engineering
Tamkang University
Tamshui 251, Taipei, Taiwan
Email: kpshih@mail.tku.edu.tw

Abstract

The paper proposes a distributed active sensor selection scheme, named DASS, for WSNs, under the requirement of complete coverage of a sensing field. By means of Voronoi diagram*, the sensor can find appropriate sensors to work together for the sensing tasks. DASS can find as few number of sensors as possible to be in charge of the sensing task. Simulation results show that DASS can efficiently select few sensors to cover the whole sensing field. Furthermore, the network lifetime can be protracted significantly in comparison with the state-of-the-art schemes.*

1 Introduction

Generally speaking, a wireless sensor network (WSN) is composed of a large number of sensors scattered in a hazardous or inaccessible environment. It is costly or unpractical to recharge or replace sensor's battery. Therefore, efficient energy management of sensors is much critical for the functioning and longevity of a WSN. On the other hand, a sensor network is usually deployed with high density. Thus, some sensors can turn off their power by turns in order to prolong the network lifetime. Consequently, on the premise of preserving the complete sensing coverage, selecting appropriate sensors to take over the sensing tasks and letting the others sleep are very important issues for dense-deployed WSNs. In the paper, how to select sensors to power on for the sensing tasks as well as preserve the complete sensing coverage at the same time is termed *Active Sensor Selection Problem* (*ASSP*).

Recently, many researches pay attention on *ASSP* [1–7]. The main idea to solve *ASSP* is, for some period of time, to select a set of sensors whose sensing coverage can span the whole sensing field to undertake the sensing task. The concept is very similar to find a *backbone* of sensors to cover the whole sensing range. According to the proposed schemes, they can be further categorized as follows.

- *Self Pruning*(*SP*):

  The schemes, [1–3], focus on the eligibility of redundant sensors. Due to the self-determination of active sensors, these schemes are easy to cause the imbalance of power consumption and result in coverage holes.

- *Initiator-Driven Selection* (*IBS*):

  The main concept of this category is to, from an initiator, select minimal number of working sensors to cover the whole sensing field [4,5]. Similar to the self-pruning-based schemes, it is also possible to cause the imbalance of power consumption.

- *Dynamic Backbone Selection* (*DBS*):

  The active sensors in *DBS* schemes, [6, 7], change dynamically with time during the sensing phase. However, the frequent switching between active and sleep modes will also waste energy in terms of battery chemical property. On the other hand, the occurrence of redundant sensors at each time instance is another disadvantage of these schemes.

The paper proposes a distributed active sensor selection scheme, named DASS, to solve *ASSP*. DASS is an initiator-driven static backbone selection scheme. During every initialization phase, DASS makes use of the concept of *Voronoi diagram* to select appropriate working sensors for the sensing tasks at the following sensing phase. The remaining power of sensors is also taken into account by DASS in order to balance the power consumption of sensors and prolong the network lifetime accordingly. DASS is shown to be a much more efficient scheme than the state-of-the-art schemes. Simulation results verify that DASS can efficiently and effectively select appropriate active sensors

*This work was supported by National Science Council of the Republic of China under Grants NSC 94-2524-S-032-003, NSC 94-2213-E-032-014, and NSC 94-2745-E-032-005-URD.

Authorized licensed use limited to: Tamkang University. Downloaded on March 23,2010 at 22:03:54 EDT from IEEE Xplore. Restrictions apply.

for the sensing tasks. The network lifetime of DASS can be protracted significantly as well in comparison with the existing schemes in the literature.

The rest of the paper is organized as follows. In Section 2, the problem statement is given. The proposed scheme, DASS, is described in Section 3. Section 4 provides the performance evaluations of DASS in comparison with the existing work. Section 5 concludes the paper.

2 Preliminaries

Let $\mathcal{F}$ be the sensing field or the area of interest to be sensed or monitored. Sensors, $S = \{s_j, j = 1, 2, \cdots\}$, are randomly deployed over $\mathcal{F}$, where sensors are homogeneous and stationary. That is, all sensors have the same sensing, communication, and processing capabilities. Let r and R respectively denote the sensing range and the communication range of a sensor, where r and R are far smaller than $\mathcal{F}$. The sensing area and the communication area of a sensor are circles centered at the sensor. Any point within the sensing range of a sensor can be detected by the sensor. In the meanwhile, any sensor within the communication range of another sensor can communicate with each other. The sensing area covered by s_j is denoted as $\mathcal{C}(s_j)$. $\mathcal{C}(s_j) = \{x \in \mathcal{F} | dist(x, s_j) \leq r\}$, where $dist(x, y)$ is the Euclidean distance between x and y. Basically, s_j always means sensor s_j. However, sometimes it also means the location of s_j according to the context. On the other hand, let $N(s_j)$ be the set of neighbors of s_j, which is defined as $N(s_j) = \{s_k \in S | dist(s_k, s_j) \leq R\}$.

To simplify the discussion, we assume that a sensor is location-aware and $R \geq 2r$. The assumption can ensure the connectivity of sensors when full sensing coverage is preserved [4]. In addition, one-hop neighbor information is assumed to be known in advance, including neighbors' locations, remaining power, and so on. On the other hand, as the same assumption in most of the related work [1–7], sensors are assumed to be synchronized. That is, each sensor can be aware of the beginning of a working cycle.

Let $Act(T_i)$ be the set of sensors, which should be active to sense in order to cover the whole sensing filed $\mathcal{F}$ during some sensing phase T_i, for some i. Obviously, $Act(T_i) \subseteq S$. The formal definition of $Act(T_i)$ is as follows.

Definition 1 *$Act(T_i)$ is a subset of S, which requires that the sensors in $Act(T_i)$ should complete cover the sensing field $\mathcal{F}$ and be active during the sensing phase T_i to undertake the sensing tasks. That is,*

$$\bigcup_{s \in Act(T_i)} \mathcal{C}(s) \supseteq \mathcal{F}.$$

□

In addition, let loc_j and Er_j be the location and remaining energy of sensor s_j, respectively, where $s_j \in S$.

3 DASS: Distributed Active Sensor Selection Scheme

DASS is an initiator-driven backbone selection scheme. DASS operates in a distributed manner with only one-hop neighbor information, including the location and remaining energy. The basic concept of DASS is introduced in Section 3.1. One important procedure involved in DASS, discovery of companion neighbors, is described in Section 3.2.

3.1 Basic concept of DASS

By the property of the Voronoi diagram [8], any point located within a Voronoi cell of s, $VC(s)$, is nearest to s than to any other Voronoi neighbors of s. Therefore, a Voronoi cell of a site can be regarded as a *responsibility area* to be in charge by the site. If the Voronoi cell of each sensor on the plane is within its sensing range of this sensor, it can be ensured that the plane is at least 1-cover, and vice versa. The formal description and verification of the phenomenon are shown in Theorem 1. The meaning of 1-cover is that any point in the plane is covered by at least 1 sensor. The definitions of 1-cover, or even k-cover, can be found in [9].

Theorem 1

$VC(s) \subseteq \mathcal{C}(s), \forall\, s \in S \Longleftrightarrow \mathcal{F}$ *is at least 1-covered by* S.

Proof: Assume $VC(s) \subseteq \mathcal{C}(s), \forall\, s \in S$ holds true. Let $x \in \mathcal{F}$, for some x. Since Voronoi diagram partitions $\mathcal{F}$ into Voronoi cells, $x \in VC(s)$, for some $s \in S$. By the assumption, $x \in \mathcal{C}(s)$. Hence, $\mathcal{F}$ is at least 1-cover.

We prove the reverse part by contradiction. The condition that $\mathcal{F}$ is at least 1-cover is held. We claim that $\forall\, x \in VC(s), x \in \mathcal{C}(s)$. For any $x \in VC(s)$, assume $x \notin \mathcal{C}(s)$. By the definition of Voronoi diagram, x is not covered by any other sensor. It contradicts to the condition that $\mathcal{F}$ is at least 1-cover. Therefore, $x \in \mathcal{C}(s)$. The claim is obtained. □

$Act(T_i)$, in some sense, acts like a backbone to be in charge of the sensing task for the whole sensing field (i.e. $\mathcal{F}$). The sensors in $Act(T_i)$ should cover the whole sensing field. Consequently, the following corollary is obtained.

Corollary 1

$$\begin{aligned} & Act(T_i) \text{ acts as a backbone to cover } F, \\ \Longleftrightarrow\ & \mathcal{F} \text{ is at least 1-covered by } Act(T_i), \\ \Longleftrightarrow\ & \forall\, s \in Act(T_i), VC(s_i) \subseteq \mathcal{C}(s). \end{aligned}$$

Proof It can be obtained directly from Theorem 1 and the definition of $Act(T_i)$. □

Authorized licensed use limited to: Tamkang University. Downloaded on March 23,2010 at 22:03:54 EDT from IEEE Xplore. Restrictions apply.

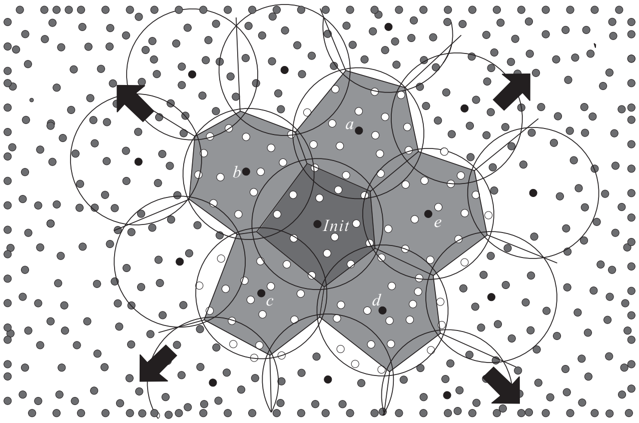


Figure 1. Basic concept of DASS.

Thus, DASS takes advantage of the results obtained from Corollary 1 to derive $Act(T_i)$. Basically, DASS operates as follows. In the beginning of the initialization phase, an initiator, $Init$, located near the center of $\mathcal{F}$ starts the finding of $Act(T_i)$. $Init$ could be predetermined by the sink or be elected by contention. Initially, $Init \in Act(T_i)$. Starting from $Init$ and scattering to the whole $\mathcal{F}$, $Act(T_i)$ is figured out to cover the whole $\mathcal{F}$ accordingly.

Take Figure. 1 as an example. Initially, all sensors, except $Init$, are in *undecided* state. On the contrary, $Init$ is in the *active* state. Based on Corollary 1, to find $Act(T_i)$, $Init$ will find a Voronoi cell, $VC(Init)$, from its neighbors first. There could exist many Voronoi cells, in terms of different neighbors. However, the Voronoi cell, $VC(Init)$, should satisfy that $VC(Init) \subseteq \mathcal{C}(Init)$. Moreover, the number of sensors involved should be as few as possible and the energy consumption should be balanced as well. The scheme to find such kind of Voronoi cell will be described in Section 3.2. In the example, the deepest gray area is $VC(Init)$, where a, b, c, d, and e are the corresponding Voronoi neighbors of $Init$. These Voronoi neighbors (a, b, c, d, and e) will be active with $Init$ as *companions* to undertake the sensing tasks collaboratively. Thus, these Voronoi neighbors are termed *companion neighbors* of $Init$. Let $N_{CP}(s)$ denote the set of companion neighbors of s, for some sensor s. In the example, $N_{CP}(Init) = \{a, b, c, d, e\}$. The state transitions of these sensors are as follows. The companion neighbors in $N_{CP}(Init)$ will be set in the *active* state. The other neighbors in $Init$'s sensing range will be set in the *sleep* state.

Afterwards, likewise, the companion neighbors in $N_{CP}(Init)$ will start finding their companion neighbors accordingly. However, if all companion neighbors start finding their companion neighbors simultaneously, it may incur conflicts among the selections of the companion neighbors. Thus, time differentiation among the companion neighbors from starting finding companion neighbors simultaneously is adopted to avoid the conflict. The detailed procedures from $Init$ scattering to the whole network is omitted due to the limitation of space. It is worth mentioning that, with the time elapsed, it is possible that the WSN is not connected any more. Some sensors may not be reached by $Init$. Thus, in the initialization phase, after timeout, a sensor still in the *undecided* state can start the scheme, DASS.

Abstractly, DASS operates as follows to obtain $Act(T_i)$.

$$Act(T_i) = Act(T_i) \cup \{N_{CP}(s), \forall s \in Act(T_i)\},$$

where, T_i is for some sensing phase T_i. Initially, $Act(T_i) = Init$. DASS utilizes the properties of the Voronoi diagram to find the companion neighbors. In order not to cause some sensors overloaded, DASS takes sensors' energy consumption into consideration. Thus, the sensors with high energy will have high possibility to be selected as companion neighbors.

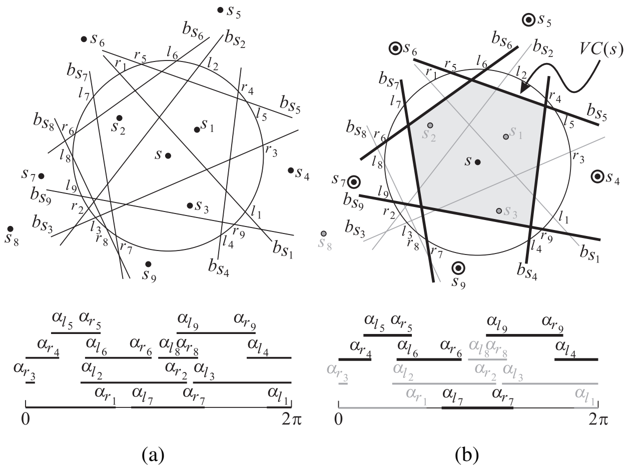
3.2 Discovery of Companion Neighbors

As mentioned above, the discovery of companion neighbors for sensor s, $N_{CP}(s)$, is an important procedure in DASS. Thus, the details of companion neighbors discovery is depicted in the section.

To find $N_{CP}(s)$, for some $s \in S$, is similar to find a $VC(s)$ from $N(s)$ such that $VC(s) \subseteq \mathcal{C}(s)$, where $N(s)$ is the set of neighbors of s. Therefore, let b_{s_j} denote the perpendicular bisector of $\overline{s_j s}$, the line segment of s_j and s, where $s_j \in N(s)$. Since the Voronoi cell formed by s and its companion neighbors should be within the sensing area of s; therefore, only the neighbors whose bisector has two intersection points with the sensing circle of s are needed to be taken into account. As a result, let the neighbors whose bisector has two intersection points with the sensing circle of s be termed *valid neighbors* of s and be denoted $N_v(s)$. Obviously, $N_v(s) \subseteq N(s)$. Thus, the problem is confined to find $VC(s)$ from b_{s_j}, $s_j \in N_v(s)$ such that $VC(s) \subseteq \mathcal{C}(s)$. An example to illustrate the companion neighbors discovery problem is shown in Figure. 2(a), where s, $N_v(s) = \{s_j, j = 1, 2, \ldots, 9\}$, and $b_{s_j}, s_j \in N_v(s)$ are shown.

Since $VC(s)$ can be viewed as a responsibility area of s, if the $VC(s)$ is as large as possible, the overlap of s with its companion neighbors can be reduced as much as possible such that the number of active sensors can be reduced as well. As a result, the companion neighbor discovery problem is shifted to find a $VC(s)$ from b_{s_j}, $s_j \in N_v(s)$, such that $VC(s)$ is as large as possible. This problem is a combinatorial optimization problem, which is known to be an *NP*-complete problem [10]. Therefore, some heuristic solutions are proposed in the paper.

Authorized licensed use limited to: Tamkang University. Downloaded on March 23,2010 at 22:03:54 EDT from IEEE Xplore. Restrictions apply.



(a) (b)

Figure 2. An example to illustrate the discovery companion neighbors. (a) The example and the perimeter covers of $p(s)$ by $b_{s_i}, \forall s_i \in N_v(s)$. (b) The companion neighbors and $VC(s)$ of s for the example shown in (a).

Let $p(s)$ denote the perimeter of the sensing circle of s. Let l_j and r_j be the two intersection points that b_{s_j} intersects with $p(s)$. That is, $b_{s_j} \cap p(s) = \{l_j, r_j\}$. Figure. 2(a) also shows l_j and r_j, for all $b_{s_j}, s_j \in N_v(s)$. The indexing is counterclockwise. Obviously, the two intersection points divide $p(s)$ into two arcs. Using the representation in [9], the shorter arc can be represented as a segment $[\alpha_{l_j}..\alpha_{r_j}]$ within $[0..2\pi]$. In other words, $p(s)$ is *perimeter covered* by b_{s_j} in angles $[\alpha_{l_j}..\alpha_{r_j}]$. Let Γ_j denote the perimeter coverage segment of $[\alpha_{l_j}..\alpha_{r_j}]$. For the example shown in Figure. 2(a), Γ_j, for all $b_{s_j}, s_j \in N_v(s)$, are illustrated in below.

Since $VC(s) \subseteq \mathcal{C}(s)$, it implies that $p(s)$ should be covered completely by the bisectors to form the $VC(s)$. Therefore, the companion neighbor discovery problem that find a $VC(s)$ from b_{s_j}, $s_j \in N_v(s)$, such that $VC(s) \subseteq \mathcal{C}(s)$ and is as large as possible is turned to, from the perimeter coverage segments ($\Gamma_j, \forall\, b_{s_j}, s_j \in N_v(s)$), find several segments which can cover the $[0..2\pi]$ and each of segments is as short as possible. To require that each segment is as short as possible is because the shorter a segment is, the larger the Voronoi cell can obtain.

As a result, the perimeter coverage segment Γ_j is given a weight $w_j, \forall\, \Gamma_j$. The weight is set as follows.

$$w_j = |\Gamma_j|, \qquad (1)$$

where $|\Gamma_j|$ stands for the length of Γ_j, which is an angle of the shorter arc resulted by b_{s_j}. According to $w_j, \forall\, \Gamma_j$, a heuristic algorithm to find segments from $\Gamma_j, \forall\, b_{s_j}, s_j \in N_v(s)$ such that these segments can completely cover $[0..2\pi]$ and each of the segments is as short as possible is illustrated in Algorithm 1. The first step of Algorithm 1 is to sort Γ_j according to their weights in ascending order (line number #3). After that, pick one Γ with the smallest weight (#5) and check whether Γ can increase the coverage of $[0..2\pi]$ (#7). If it is true, the Γ is selected. Likewise, check another Γ with the next smallest weight until $[0..2\pi]$ is completely covered or all Γs are checked. It is worth noting that after the previous selections, it may include some redundant segments. Therefore, the removal of redundant segments is necessary (#10 to #14) in order to obtain as few segments as possible. For the example shown in Figure. 2(a), the results obtained by performing Algorithm 1 are as follows, where $w_8 < w_5 < w_6 < w_7 < w_4 < w_9 < w_2 < w_3 < w_1$. After the select-and-check steps, Γ_8, Γ_5, Γ_6, Γ_7, Γ_4, and Γ_9 are selected and they can completely perimeter cover $[0..2\pi]$. In the removal of redundant segment steps, Γ_8 is removed. As a result, Γ_5, Γ_6, Γ_7, Γ_4, and Γ_9 are selected finally. Consequently, $N_{CP}(s) = \{s_4, s_5, s_6, s_7, s_9\}$. The result is illustrated in Figure. 2(b).

Algorithm 1: Companion Neighbors Discovery

```
   Input: $\Gamma_j, \forall\, b_{s_j}, s_j \in N_v(s)$
   Output: $N_{cp}(s)$, for some $s \in S$
1  begin
2      $V \leftarrow \emptyset$ ;
3      $U \leftarrow$ Sort $\Gamma_j, \forall\, b_{s_j}, s_j \in N_v(s)$ by weight $w_j$ in
       ascending order ;
4      while $[0..2\pi]$ is not completely covered and $U \neq \emptyset$ do
5          $\Gamma_{new} \leftarrow$ the first one in $U$ ;
6          $U \leftarrow U - \Gamma_{new}$ ;
7          if $\Gamma_{new}$ can increase the coverage of $[0..2\pi]$ then
           $V \leftarrow V \cup \Gamma_{new}$ ;
8      end
9      $V \leftarrow$ Sort $\Gamma_j, \forall\, \Gamma_j \in V$ by weight $w_j$ in descending
       order ;
10     while $\exists\, \Gamma \in V$, which has not been checked do
11         $\Gamma \leftarrow$ the first unchecked segment in $V$ ;
12         if Removal of $\Gamma$ will not violate the coverage of
           $[0..2\pi]$ then $V \leftarrow V - \Gamma$ ;
13         else Mark $\Gamma$ checked ;
14     end
15     $N_{cp}(s) \leftarrow s_j, \forall\, \Gamma_j \in V$ ;
16 end
```

The time complexity of Algorithm 1 is $O(n^2)$, if a sensor has at most n valid neighbors. The complete analysis of Algorithm 1 is omitted due to the space limitation.

In principle, the fewer the sensors are involved in forming a Voronoi cell, say $VC(s)$, the less the overlap of the Voronoi neighbors with s is. Therefore, from the viewpoint of s, it is the best choice if $VC(s)$ is as large as possible. However, from the global point of view, the largest $VC(s)$

Authorized licensed use limited to: Tamkang University. Downloaded on March 23,2010 at 22:03:54 EDT from IEEE Xplore. Restrictions apply.

is not a good choice since it will result in much overlap among the Voronoi neighbors. [4] have shown that the best placement of three sensors is that the distance between any two sensors is $\sqrt{3}r$, where r is the sensing range of a sensor. It will result in $|\Gamma| = \frac{\pi}{3}$. Therefore, the Γ whose length is closer to $\frac{\pi}{3}$ has a high priority to be selected. Consequently, the weight of Γ_j, w_j, is modified as follows.

$$w_j = abs(|\Gamma_j| - \frac{\pi}{3}), \qquad (2)$$

where $abs()$ is an absolute value function.

Suppose s_j is a valid neighbor of s. Actually, the meaning behind $|\Gamma_j|$ stands for the distance between the two sensors. The larger $|\Gamma_j|$ is, the closer s_j to s is. Thus, in Eq. (2), only the distance between s and s_j is considered. This will cause a phenomenon that the sensors selected in $Act(T_i)$ will not be replaced except that they exhaust their energy. It violated the subjection of balancing the energy consumption in problem statement. Consequently, the remaining energy of the sensor s_j, Er_j, is taken into consideration as well. To normalize the two factors, $|\Gamma_j|$ and Er_j, $|\Gamma_j|$ will divide $\frac{2\pi}{3}$ since $0 < |\Gamma_j| < \pi$. As a result, the weight of Γ_j, w_j, is finally modified as follows.

$$w_j = \lambda \frac{abs(|\Gamma_j| - \frac{\pi}{3})}{\frac{2\pi}{3}} + (1 - \lambda)(1 - Er_j), \qquad (3)$$

where $0 \leq \lambda \leq 1$. The effect of λ on the performance, in terms of the number of active sensors and network lifetime, will be discussed in Section 4.

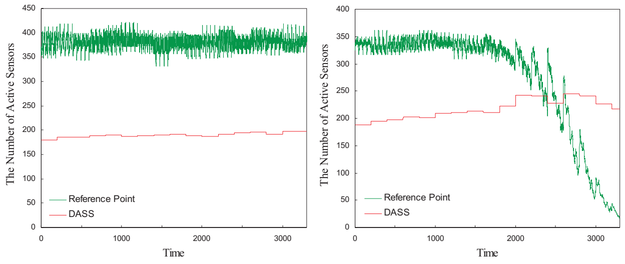
4 Performance Evaluation

To verify the effectiveness of the proposed protocol, a lot of experiments are performed in C++ language. In addition to DASS protocol, Reference Point scheme [6] is simulated. The evaluation metrics include the number of active sensors, the number of alive sensors, and coverage percentage. Sensors are randomly deployed and consume 1 unit power per time unit during sensing model. Simulation parameters are shown in Table 1.

Table 1. System Parameters

Parameter	Value
Simulation area	500 meters × 500 meters
Transmission range	60 meters
Sensing range	30 meters
Round time (used in [6])	50 time unit
Sensing Phase time	200 time unit
The initial energy of each sensor	1000 unit power
The number of sensors	900(d=1) and 2700(d=3)

Figure. 3 illustrates the variations of number of active sensors subject to the simulation time when the network density is 1 and 3, respectively. DASS can select appropriate working sensors, $Act(T_i)$, and minimize the number of $Act(T_i)$ to preserve the coverage requirement. On the other hand, Reference Point scheme does not guarantee the minimal number of working sensors. In other words, DASS performs better than Reference Point scheme in the number of active sensors.



(a) The network density is 3.　　(b) The network density is 1.

Figure 3. The variations of number of active sensors subject to the simulation time.

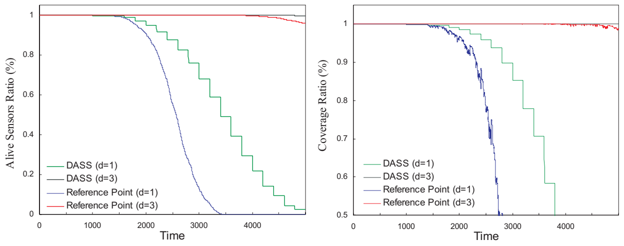
Note that, a sensor may have more neighbors in its sensing range when the network density is high. Reference Point scheme can schedule very well for each sensor to work in a short period. However, when the network density is sparse, the working time of each sensor will increase. Thus, it is easier for a sensor to exhaust its energy. As a result, the number of active sensors in Reference Point scheme may decrease with the time elapsed.

On the other hand, it is sufficient for DASS to fine a minimal $Act(T_i)$ to cover the whole sensing filed when the network density is 1. Thus, the number of active sensors may be equal no matter how the network density is 1 or 3. Note that there are more redundant sensors in Reference Point scheme, especially, when the network density is dense. Thus the number of active sensors for Reference Point scheme in Figure. 3(a) is more than that in Figure. 3(b).

Figure. 4(a) shows the variations of alive sensors ratio subject to the simulation time. According to Figure. 3, DASS performs better than Reference Point scheme in the number of active sensors. Additionally, initiator changed in each *Initialization Phase*, the different initiator can get the different minimal $Act(T_i)$ to cover the whole sensing filed. Thus, the energy of each sensor may be consumed equally and moderately. On the other hand, there are more redundant sensors in Reference Point scheme. Thus, much energy is consumed over the whole sensing field and it is easier for sensors exhausting their energy. In other words, DASS performs better than Reference Point scheme in alive sensors ratio metric.

Note that the variance of energy consumption between

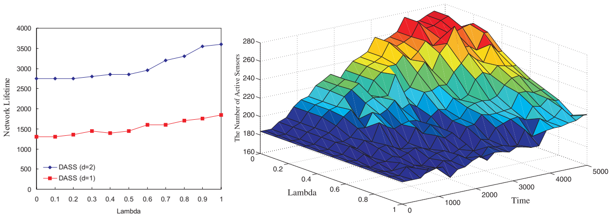
Authorized licensed use limited to: Tamkang University. Downloaded on March 23,2010 at 22:03:54 EDT from IEEE Xplore. Restrictions apply.



(a) The variations of the alive sensors ratio to the simulation time.

(b) The variations of coverage ratio to the simulation time.

Figure 4. The variations of number of alive sensors ratio and coverage ratio subject to the simulation time.



(a) The variations of network lifetime subject to lambda

(b) The number of active sensors subject to lambda and time

Figure 5. The variations of number of active sensors subject to the simulation time.

each sensor is slight in Reference Point scheme. Thus, it is easy for sensors to exhaust their energy at the same time, and the number of alive sensors in Reference Point scheme may decrease rapidly as the time pass. Figure. 4(b) shows the variations of coverage ratio subject to the simulation time. Obviously, the more alive sensors ratio is, the more the coverage ratio is. Thus, the simulation results in Figure. 4(a) are like in Figure. 4(b).

In Figure. 5, while λ is 1, the number of active sensors is optimal. As a result, the network lifetime is increasing, but the energy consuming between each sensor is not balance. On the contrary, if $lambda$ approximates to 0, we prefer to consider the remaining energy matrix to find companion neighbors instead of selecting the sensors within the optimizing location to minimize companion neighbors. In this circumstance, the more sensors will be active, such that the network lifetime is decreasing, but the energy consuming is more balenced.

5 Conclusions

The paper proposes a distributed active sensor selection scheme, named DASS, to solve *ASSP* problem. DASS is an initiator-driven backbone selection scheme, which can select appropriate sensors effectively as a backbone to cover the whole sensing field and undertake the sensing obligation. DASS utilizes the properties of Voronoi diagram to minimize number of active sensors. Moreover, DASS also takes sensors' remaining energy into consideration to balance the energy consumption of sensors. Simulation results reveal that DASS has better performance in comparison with the state-of-the-art scheme in terms of the number of active sensors and network lifetime. DASS indeed is an efficient and effective active sensor selection scheme in solving *ASSP* problem.

References

[1] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: A robust energy conserving protocol for long-lived sensor networks," in *Proceedings of the 23th IEEE International Conference on Distributed Computing Systems (ICDCS)*, May 2003, pp. 28–37.

[2] B. Carbunar, A. Grama, J. Vitek, and O. Carbunar, "Coverage preserving redundancy elimination in sensor networks," in *Proceedings of the IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2004, pp. 377–386.

[3] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SENSYS)*, 2003, pp. 28–39.

[4] H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," in *Proceedings of the NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, 2004.

[5] H. Gupta, S. R. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2003, pp. 189–200.

[6] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SENSYS)*, 2003, pp. 51–62.

[7] C.-F. Huang, L.-C. Lo, Y.-C. Tseng, and W.-T. Chen, "Decentralized energy-conserving and coverage-preserving protocols for wireless sensor networks," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005.

[8] F. Aurenhammer, "Voronoi diagrams–a survey of a fundamental geometric data structure," *ACM Computing Survey*, vol. 23, no. 3, pp. 345–405, Sept. 1991.

[9] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," in *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2003.

[10] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Pretince-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

Authorized licensed use limited to: Tamkang University. Downloaded on March 23,2010 at 22:03:54 EDT from IEEE Xplore. Restrictions apply.