Approximate K-Coverage Configuration in WirelessSensor Networks

Prasan Kumar Sahoo
Dept. of Information Management
Vanung University, Taiwan
Email: pksahoo@mail.vnu.edu.tw

Meng-Chun Wueng,I-Shyan Hwang
Dept. of Computer Scienceand Engineering
Yuan Ze University, Taiwan
Chunghwa Telecom Co. Ltd., Taiwan

***Abstract*—The K-coverage configuration is widely exploited to guarantee the surveillance quality of applications on wireless sensor networks. To prolong the system lifetime, a sensor node is determined to sleep if its sensing range is already K-covered. Many K-coverage configuration algorithms in literature cannot satisfy the requirements of high quality and low cost simultane-ously. In this paper, we propose an efficient K-coverage eligibility algorithm, which determines the eligibility of each sensor node at very low cost. The distinct feature of the ACE algorithm is to discover the regions with lower coverage degree of each sensor node. Experimental results show that the accuracy of the ACE algorithm is guaranteed to be higher than 90%, while its computational cost is only 11% of a well-known deterministic algorithm. The ACE algorithm is suitable to be used for a long-term monitoring task on wireless sensor networks.**

I. Introduction

Advances in micro-sensor and wireless communication tech-nologies enable small and inexpensive sensor nodes to perform cooperative tasks for important applications, such as surveil-lance, target tracking, military tasks, and hazardous environ-ment exploration. With the consideration that sensor nodes may exhibit faulty behavior, related fault tolerant technologies are investigated to guarantee the quality of applications on sensor networks. The faulty behavior of sensor nodes may result from many situations, such as a faulty decision from the signal processing in a senor node due to noise ([6]), environmental interference, battery depletion, or malfunctions due to low-cost hardware. Based on the fact that an individual sensor node is not reliable, a higher degree of coverage is necessary to mask the faults of sensor nodes and to obtain a higher confidence in detection [2], [10]. Therefore, the K-coverage configuration was proposed to preserve that each location in an area is covered by at least K active sensors [7], [12]. Many coverage-preserving scheduling schemes were further proposed to guarantee the required coverage degree while minimizing the number of active sensor nodes ([4], [11], [1], [3], [8]).

For the K-coverage configuration, a fundamental problem is how to determine that the monitored area is K-covered. Xing et al. [11] have proved that this problem can be transformed to calculate the coverage degree of each sensor node within the monitored area. Furthermore, the coverage degree of each node can be obtained by tracing all points, which are intersected by its neighbors, within the sensing range. To reduce the power consumption, the authors further propose a K-coverage eligibility (KE) algorithm. A sensor node can be determined to be *ineligible* to stay active if all intersection points within its sensing range are already K-covered by its neighbors. Therefore, the number of active sensor nodes can be reduced while the surveillance quality still can be guaranteed. Although the deterministic K-coverage eligibility algorithm can accurately determine the eligibility of each sensor node, the computational cost is $O(n^3)$where n is the number of the neighbors within twice the sensing range of each node.

In this paper, we propose an efficient Approximate K-Coverage Eligibility (ACE) algorithm that can correctly deter-mine the eligibility of each sensor node with low cost. The distinct feature of the ACE algorithm is that we classified the neighbors of each sensor node into R neighbors and R–2R neighbors, which are defined in Section 3. Instead of calculating the coverage degree of all intersection points within the sensing range of a node, the ACE algorithm only requires to focus on the candidate intersection points surrounding the lower coverage regions based on the characteristics of the R neighbors and R–2R neighbors. Therefore, the computa-tional cost of the ACE algorithm can be highly reduced. However, since the algorithm aims to discover the regions with lower coverage degree not the minimal coverage degree, the accuracy may be decreased. Although the accuracy of the ACE algorithm cannot be guaranteed as 100%, according to the experimental results, the correct percentage is larger than 90% as the number of the deployed sensor nodes increases. Furthermore, the computational cost is only 11% of that of KE algorithm [11]. With the consideration that wireless sensor networks have scarce energy resource, it is acceptable to have less than 10% locations in the monitored area are under K-covered ([12], [9], [11]).

The rest of this paper is organized as follows. In the next section, we briefly review the related work in the literature. Section 3 presents the design issues of ACE. Simulation results are presented and discussed in Section 4. Finally, we conclude this paper in Section 5.

II. RELATED WORK

Similar to the goal of our research, Huang et al. [5] attempted to reduce the computational cost of the K-coverage configuration. They proposed a K-perimeter-covered (KPC) algorithm to calculate the coverage degree of each sensor node by tracing the perimeter segments covered by its neighbors. Since this algorithm does not need to consider the coverage within the sensing range of a node, the computational cost can be effectively reduced. However, the accuracy on deter-mining the eligibility for each sensor node is thus decreased. Furthermore, the KPC algorithm ignores that the sensor nodes located near the monitored edges have some invalid perimeter segments, and the coverage degree of the invalid perimeter segments should not be calculated. Therefore, a sensor node located near the monitored edges may be determined to be eligible to become active, but its sensing range within the monitored edge is already K-covered in reality. As more and more sensor nodes are deployed, the accuracy of the KPC algorithm will be further decreased.

Compared with the KPC algorithm, the ACE algorithm determines the eligibility of a sensor node by calculating the coverage degrees of the intersection points surrounding the lower coverage regions. The computation cost of ACE is highly reduced. For the sensor nodes located near the monitored edges, only the coverage degrees of the intersection points within the monitored edges are calculated. Therefore, the accuracy of the ACE algorithm is guaranteed.

III. ACE ALGORITHM

In this research, all sensor nodes with identical sensing range, *R*, are assumed to be location-aware, and no other sensor node locates at the same position in the monitored area. For calculating the coverage degree of a sensor node, we define that an arbitrary point *p* is covered by a sensor node *s* if their Euclidian distance is less than the sensing range *R*, that is, $d(s,p)<R$. With the physical consideration of signal decay of a sensor node, a point which is located exactly at the sensing range of a sensor may not be detected correctly. Therefore, it is reasonable to assume that *p* is not covered by *s* if $d(s,p)=R$. On the other hand, we classify the neighbor set of each node into two groups, called R neighbors and R– 2R neighbors.

Definition 1: R neighbors and R–2R neighbors. The R neighbors of sensor node *i* are defined as R neighbors(i)= $\{j \mid j \in N, j = i, d(i,j)<R\}$, where *N* is the set of sensor nodes located in the monitored area and *d(i,j)* represents the distance between nodes *i* and *j*. The R–2R neighbors of *i* are defined as R–2R neighbors(i)= $\{j \mid j \in N, j = i, R \le d(i,j)<2R\}$.

A. Design Issues of ACE

There are two reasons to classify the neighbor set of a sensor node into two groups and calculate their coverage degree individually. Taking Figure 1 as an example, the first reason is that while farther from the target sensor node *i*, R–2R neighbors tend to form an area with lower coverage degree inside the sensing range of the node. Even if R–2R neighbors are very close to the sensor node, the node will not be fully covered by all R–2R neighbors based on the assumption that a point is not covered when it is located exactly at the sensing range of a node. Hence, when a sensor node has only R–2R neighbors, the coverage degree of the node can be determined immediately, that is 1. The second reason is that the number of R neighbors is bounded by the sensing range of a node.

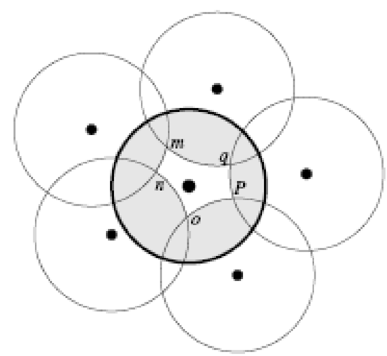

Fig. 1. (a) The central area surrounded by *m, n, o, p, q* has lower coverage degree.

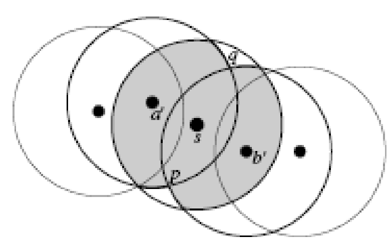

Fig. 2. The eligibility of node *s* can be determined by tracing only *p* and *q*.

Moreover, in many cases even if a sensor node has R neighbors and R–2R neighbors, the eligibility of the node can be determined by only tracing the intersection points of R neighbors, as illustrated in Figure 2. Therefore, if we can classify the neighbors of a sensor node into two groups, and calculate the coverage degree of the points intersected by R neighbors first, the computational cost in many cases can be bounded by the number of R neighbors.

The methods of calculating the coverage degree of sensor nodes can be classified into three cases. For the first case, when one R neighbor cannot cover the entire sensing range of a sensor node, and the R–2R neighbors form a lower coverage region in the center of the node, then the lower coverage regions will be surrounded by the R neighbors and other R–2R neighbors. The regions can be discovered by finding out the points with the minimal coverage degree intersected by the R neighbor and the R–2R neighbors. Even if the R neighbor does not have any intersection with the R–2R neighbors, the minimal coverage is the sensor node itself. The eligibility of the node can still be determined.

For the second case, although the intersection points of

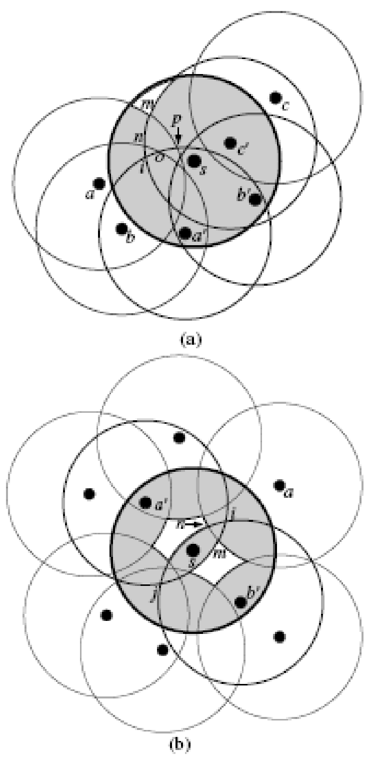

Fig. 3. Two cases of the lower coverage region formed by the candidate neighbors.

R neighbors cannot be used to determine the eligibility of the node, in most cases, the eligibility of the node can still be decided by tracing the points intersected by any two R neighbors and R–2R neighbors. During the processing, the algorithm terminates when the coverage degree of any intersection point is less than the required K-coverage degree. In ACE, we do not trace the points intersected by any two R–2R neighbors. Since the operation not only incurs lots of computations, but also cannot find out the intersection points with lower coverage degree quickly.

The third case is more complicated. To clearly explain the case, we introduce several keywords. The intersection points of any two R neighbors covered by the fewest R neighbors are called *candidate intersection points* and the two R neighbors are called *candidate R neighbors*. The R–2R neighbors that cover the *candidate intersection points* are represented as *can-didate R–2R neighbors*. Besides, the *decision points* mean the points intersected by the *candidate R neighbors* and the *candi-date R–2R neighbors*. In this case, the *candidate intersection points* are covered by several *candidate R–2R neighbors*. The lower coverage regions usually can be found by tracing the *decision points* and the *candidate intersection points*. Taking Figure 3(a) as an example, the intersection point *i* is the *candidate intersection point* covered by the *candidate R– 2R neighbors*, that is, *a* and *b*. The lower coverage region

Pseudocode 1 The main steps of ACE

Step 1: Each sensor collects the neighbor information,and then classifies its neighbors into R neighbors and R–2R neighbors.**Step 2**: Each sensor performs ACE to determine the eligibil-ity by checking the relationship of its R neighbors and R–2R neighbors**Step 3**: For a sensor *s*, if it has only R–2R neighbors butno any R neighbor, then its coverage degree must be 1. Theprocess is terminated.**Step 4**: If *s* has both R neighbors and R–2R neighbors, thenACE traces the *candidate intersection points*.**Step 5**: If the *candidate intersection points* are covered bysome *candidate R–2R neighbors* ACE traces the region withlower coverage degree is surrounded by *m* which has the minimal coverage degree among *m*, *n*, *o*, *p*, *q*, *i* which are intersected by *a'*, *c'*, *a*, and *b*. In this way, the coverage of the R–2R neighbors within the sensing range of the R neighbors will just increase the cover-age degree of the sensor node, as the coverage of *c* in Figure 3(a). The coverage degree in this overlap is usually higher than the required K-coverage degree. Therefore, to determine the eligibility of a node with low computational cost, when the lower coverage regions surrounded by the R neighbors are found, we focus that how the R–2R neighbors cover the founded regions. If the *candidate R–2R neighbors* do not fully cover the region, as Figure 3(a), a new lower coverage region will be formed by the *candidate R neighbors* and the *candidate R–2R neighbors*. Therefore, we only need to trace their intersection points and find out the points with the minimal coverage degree. On the other hand, if the *candidate R–2R neighbors* fully cover the regions, since the coverage of R–2R neighbors on the sensing range of a node is limited, the lower coverage regions will be surrounded in the center by the *candidate R neighbors* and the *candidate R–2R neighbors*, as the regions surrounded by *m* and *n* in Figure 3(b).

B. Algorithm Description

Pseudocode 1 presents the major operations of ACE. When all sensor nodes are deployed in a monitored area, they are initially in the active state and start to collect the neighbor information within twice their sensing ranges. Each sensor node *i* divides its neighbor set into *R neighbors(i)* and *R–2R neighbors(i)*. If a sensor node *i* only has *R– 2R neighbors(i)* but no *R neighbors(i)*, its coverage degree is 1. The eligibility of *i* is determined immediately without any further computation. This is one of the benefits we classify the neighbor set of a sensor node into two groups. If *i* has both *R neighbors(i)* and *R–2R neighbors(i)*, the al-gorithm first finds out the *candidate intersection points*. If the *candidate intersection points* are not covered by any *R– 2R neighbors(i)*, the eligibility of *i* is determined directly from the *candidate intersection points*. On the other hand, if the *candidate intersection points* are covered by the *candidate*

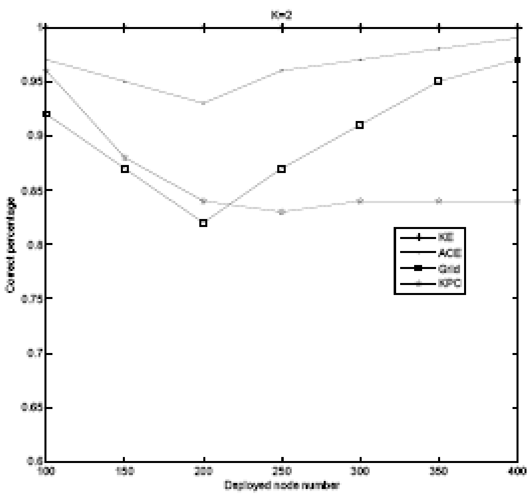

Fig. 4. The correct percentages of the four algorithms.

R–2R neighbors(i), the region with lower coverage degree needs to be discovered. To reduce the computational cost, ACE only calculates the coverage degree of the *decision points* intersected by the *candidate R neighbors(i)* and the *candidate R–2R neighbors(i)*. In this algorithm, the worse case is that the *candidate intersection point* is not found. The algorithm needs to trace the points intersected by all *R neighbors(i)* and *R–2R neighbors(i)*. Compared to other algorithms, the overall computational cost is still reduced because the points intersected by any two *R–2R neighbors(i)* are not traced. During the overall processing, the algorithm terminates when the calculated coverage degree of a sensor node is less than the required *K* coverage degree.

IV. PERFORMANCE EVALUATION

We evaluated the performance of the ACE algorithm on NS-2 in terms of accuracy and computational cost. Three related algorithms are also implemented to compare with the ACE algorithm, including the KE algorithm ([11]), the KPC algorithm ([5]), and the Grid algorithm. For the K-coverage configuration, the simple Grid algorithm is usually used to approximately determine the coverage degree of a monitored area. In our evaluation, the sensing area of each sensor node is divided into 1m × 1m grids. The coverage degree of each grid is obtained by calculating how many active sensor nodes cover the center of the grid. The eligibility of each sensor node can thus be determined by tracing all grids within its sensing range. The simulation environment is a 50m × 50m square space, and the sensing range of all deployed sensor nodes is 5m. Each result is the average of five runs with different random network topologies. All algorithms terminate when the coverage degree of a sensor node is less than or equal to the required K-coverage degree.

The accuracy and efficiency of the four algorithms were illustrated in Figures 4 and 5. The KE algorithm precisely

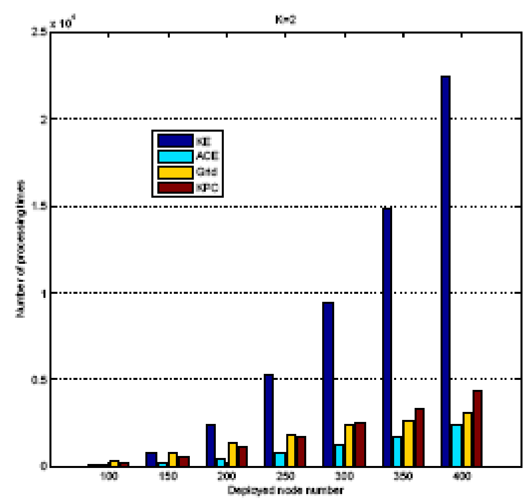

Fig. 5. The number of processing times of the four algorithms.

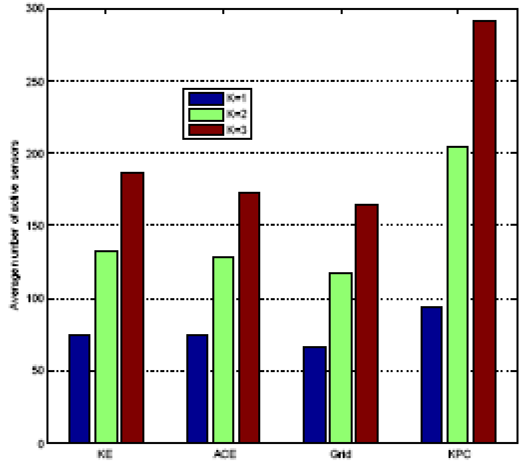

Fig. 6. The average number of active sensors of the four algorithms.

determines the eligibility of each sensor node. However, the computational cost of the KE algorithm is considerably high as the deployed sensor nodes increase. Although the KPC algorithm effectively reduces the complexity of the KE algorithm, its accuracy cannot be guaranteed. Many sensor nodes are considered to have lower coverage degrees and lots

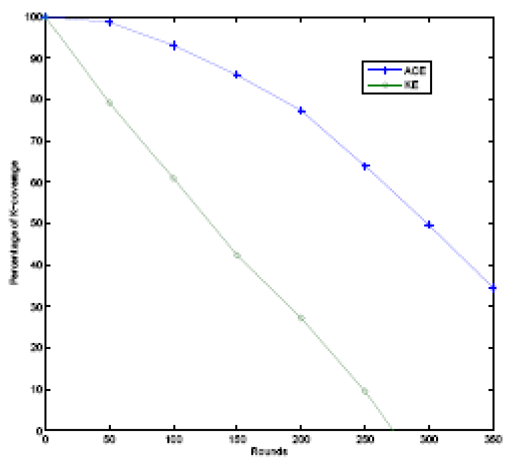

Fig. 7. The system lifetime of the four algorithms.

number of the deployed nodes increases, its computational cost can be bounded by the number of the grids. Compared to the performance of the KPC and Grid algorithms, the proposed ACE algorithm has the highest correct ratio and lowest computational cost. This is because the ACE algorithm classifies the neighbors of each sensor node into R neighbors and 2R neighbors. The eligibility of each node is determined by tracing only the intersection points surrounding the lower

of redundant nodes need to be active. When there are more and more sensor nodes deployed, the number of nodes near the monitored edges also increases, so that the error ratio of the algorithm is highly increased. In Grid algorithm, the major issue is how to determine the grid size, because both the accuracy and the computational cost are affected by the size. Figure 5 shows that when the number of the deployed nodes is less than 150, the computational cost of Grid algorithm is higher than the other three algorithms. However, as the

TABLE I
THE AVERAGE NUMBER OF NODES MAKING WRONG DECISIONS BEFORE PERFORMING THE OFF-DUTY RULE IN A ROUND.

Degree	Algorithms	Error cases	100 nodes	150 nodes	200 nodes	250 nodes	300 nodes	350 nodes	400 nodes
K=1	ACE	under K-covered	4	6	3	3	3	2	3
		over K-covered	-	-	-	-	-	-	-
	Grid	under K-covered	12	19	17	12	9	5	4
		over K-covered	-	-	-	-	-	-	-
	KPC	under K-covered	9	9	5	2	2	0	0
		over K-covered	9	17	24	34	39	45	49
K=2	ACE	under K-covered	3	8	15	13	9	6	5
		over K-covered	-	-	-	-	-	-	-
	Grid	under K-covered	8	20	35	33	27	16	14
		over K-covered	-	-	-	-	-	-	-
	KPC	under K-covered	1	10	14	12	4	2	1
		over K-covered	3	7	19	30	43	55	64
K=3	ACE	under K-covered	1	7	17	19	21	14	10
		over K-covered	-	-	-	-	-	-	-
	Grid	under K-covered	1	14	31	40	41	38	24
		over K-covered	-	-	-	-	-	-	-
	KPC	under K-covered	1	4	10	18	18	9	3
		over K-covered	0	2	9	19	31	50	68

degree regions rather than all intersection points within the whole sensing range. Hence, the ACE algorithm can guarantee a high quality of surveillance while prolonging the system lifetime.

In the K-coverage configuration, all sensors, which are eligible to sleep, will perform the off-duty rule in the self-organization phase to select some sensors stay active in the following sensing phase. Figure 6 shows the average number of active sensors derived by these four algorithms. Except the KE algorithm, the ACE, Grid, and KPC algorithms may have error cases in the self-organizing phase. Such error cases are classified into the under K-covered case (in which the location is not covered by at least K sensor nodes) and the over K-covered case. Table 1 shows the average number of sensor nodes making wrong decisions by executing the ACE, Grid, and KPC algorithms, compared with the results derived by the KE algorithm. Since the ACE, Grid, and KPC algorithms may not discover the minimal coverage degree of each sensor node, all of them cause some locations in the monitored area are not K-covered. Finally, Figure 7 illustrates the system lifetime of KE and ACE algorithms until the area is under desired K-coverage.

V. CONCLUSIONS

In this paper, we proposed an efficient K-coverage eligibility (ACE) algorithm, which accurately determines the eligibility of sensor nodes at low cost. In ACE, the neighbors of each sensor node can be classified into R neighbors and R– 2R neighbors. Based on the characteristics of the two groups, the lower coverage regions of each sensor node can be discov-ered efficiently. Therefore, only some candidate intersection points surrounding the lower coverage regions need to be traced. Simulation results demonstrate that the accuracy of the ACE algorithm is higher than 90%, but the computation cost of ACE is only 11% of the KE algorithm. We believe that the ACE algorithm enables low-cost sensor nodes to monitor events over a long duration.

REFERENCES

[1] H. M. Ammari and S. K. Das. On the Design of K-covered Wireless Sensor Networks: Self-versus Triggered Sensor Scheduling. In *Proc. of IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks and Workshops*, pages 1–9, 2009.

[2] J.-F. Chamberland and V. V. Veeravalli. How Dense Should a Sensor Network be for Detection with Correlated Observations. *IEEE Transactions on Information Theory*, 52(11):5099–5106, 2006.

[3] M. Hefeeda and M. Bagheri. Randomized K-coverage Algorithms for Dense Sensor Networks. In *Proc. of IEEE INFOCOM*, pages 2376–2380, 2007.

[4] C.-F. Hsin and M. Liu. Network Coverage Using Low Duty-Cycled Sensors: Random & Coordinated Sleep Algorithms. In *Proc. of the ACM International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 433–442, 2004.

[5] C.-F. Huang and Y.-C. Tseng. The Coverage Problem in a Wireless Sensor Network. *Mobile Networks and Applications*, 10(4):519–528, 2005.

[6] M. Ilyas and I. Mahgoub. *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC, 2004.

[7] S. Kumar, T. H. Lai, and J. Balogh. On k-coverage in a Mostly Sleeping Sensor Network. In *Proc. of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 144–158, 2004.

[8] Y. Li and S. Gao. Designing K-Coverage Schedules in Wireless Sensor Networks. *Journal of Combinatorial Optimization*, 15(2):127–146, 2008.

[9] Y. Liu and W. Liang. Approximate Coverage in Wireless Sensor Networks. In *Proc. of the IEEE Intersectional Conference on Local Computer Networks (LCN)*, pages 68–75, 2005.

[10] T.-Y. Wang, L.-Y. Chang, D.-R. Duh, and J.-Y. Wu. Fault-Tolerant Decision Fusion via Collaborative Sensor Fault Detection in Wireless Sensor Networks. *IEEE Transactions on Wireless Communications*, 7(2):756–768, 2008.

[11] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks. *ACM Transactions on Sensor Networks*, 1(1):36–72, Aug. 2005.

[12] H. Zhang and J. Hou. On Deriving the Upper Bound of α-Lifetime for Large Sensor Networks. In *Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 121–132, 2004.