

# Optimizing Sensing Quality with Coverage and Lifetime in Wireless Sensor Networks

Prof.K.Karuppasamy, M.E., (Ph.D.),  
The Head of the Department,  
Department of Information Technology,  
RVS College of Engineering & Technology,  
Coimbatore.

Dr.V.Gunaraj, M.E., Ph.D.,  
The Principal,  
RVS College of Engineering & Technology,  
Coimbatore.

## Abstract

*The coverage requirement and lifetime constraint cannot be satisfied only by using the density of the sensor. Coverage has to be traded for network lifetime. In this paper, we study about scheduling the sensors to maximize their coverage during a specified network lifetime, our objective is to maximize the spatial temporal coverage by scheduling sensors activity after they have been deployed. We first present a centralized algorithm design whose approximation factor is proved to be and then, propose an optimizing protocol. In optimizing protocol, individually the nodes optimize their schedules without conflict with one another. Theoretical and simulation results show that optimizing protocol substantially outperforms other schemes in terms of network lifetime, coverage redundancy, convergence time.*

**Index Term** – Wireless sensor network, Coverage

**Keywords** – Sensor scheduling, Distributed protocol, Parallel algorithm.

## 1. Introduction

In wireless sensor networks, there is a trade-off between network lifetime and sensor coverage. To achieve a better coverage, more sensors have to be active at the same time, then more energy would be consumed and the network lifetime is reduced. On the other hand, if more sensors are put into sleep to extend the network lifetime, the coverage will be adversely affected. The trade-off between network lifetime and sensor coverage cannot be simply solved at the deployment stage, because it is hard to predict

the network lifetime requirement, which depends on the application and may change as the mission changes. For example, in a surveillance application, the initial mission is to monitor the battle field for 6 hours. As the battle goes on, the commander finds that the battle may have to last for 10 hours. Then, the mission of the sensor network is changed, which requires the network to last for 10 hours. Since it may not be possible to deploy more sensors, some sensors have to sleep longer during each duty cycle to extend the network lifetime. As a result, sensor coverage needs to be traded for network lifetime. The coverage issue in sensor networks has been studied extensively [1], [2], [3], [4], [5], [6], where scheduling algorithms are proposed to maximize the network lifetime while maintaining some predefined coverage degree. However, if the same coverage degree is maintained all the time, the lifetime requirements may not be satisfied as network condition and mission change. For example, the sensor density may drop over time and the coverage requirement may vary according to the application's demand. Different from existing works, we study how to schedule sensor nodes to maximize coverage under the constraint of network lifetime. This reverse formulation is especially useful when the number of nodes is not enough to maintain the required coverage degree for a specified time period, as shown in the above example. In this paper, we aim to resolve the conflict between the static status of sensor deployment and the dynamic nature of mission requirements. As mission dynamically changes, the lifetime and coverage requirement may not be satisfied at the same time. Then, the coverage needs to be traded for the network lifetime. Our work is thus complementary to the existing work, which

can be applied when the sensor density is sufficient to sustain both the lifetime and coverage requirement. To fulfill this goal, we have to consider the coverage in both spatial and temporal domain. In particular, we define a new spatial-temporal coverage metric, in contrast to the traditional area coverage. The spatial-temporal coverage of each small area is defined as the product of the area size and the length of the period during which the area is covered. Then, our objective becomes how to schedule the sensor's on period to maximize the global spatial-temporal coverage, calculated as the sum of individual spatial-temporal coverage over all the areas. This new formulation arises naturally from the mission critical applications with the network lifetime constraint and differentiates itself from most existing works which only consider the spatial domain.

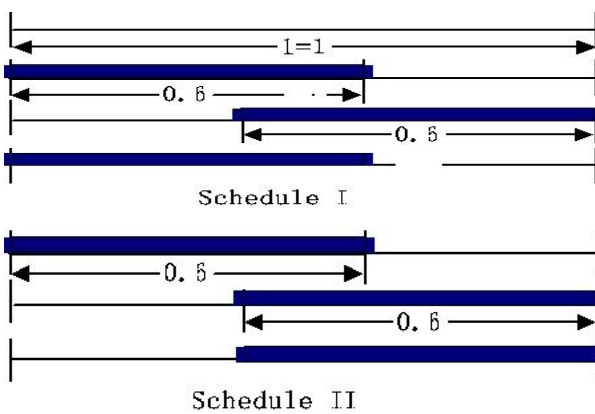
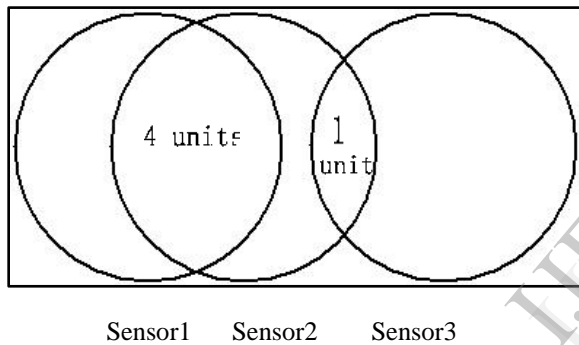


Fig 1. A Surveillance example with three sensors

Our contribution in this paper can be summarized as follows:

- First, we formalize the sensor scheduling problem in the spatial and temporal dimension with the objective to maximize

the spatial-temporal coverage with network lifetime constraint. We further prove that it is equivalent to minimize the coverage redundancy under certain conditions.

- Second, we propose a distributed heuristic, the optimizing protocol (POP), where nodes not only optimize their schedules on their own but also converge to local optimality without conflict with one another.

**2. Problem Formulation**

When the sensor density is not sufficient to satisfy both the lifetime and coverage requirements, the coverage has to be traded for lifetime. In such a case, the sensors have to make their best efforts to provide the coverage while meeting the lifetime constraint. To achieve this, we divide the network lifetime  $L$  into cycles and turn on each sensor within each cycle for a period proportional to its battery life. We further designate that the same schedule repeats in each cycle, such that the sleep schedule can be implemented, e.g., using the Power Saving Mode of 802.11. Then, the purpose of the scheduling is to place the on-periods within each cycle, such that the total spatial-temporal coverage can be maximized. We formalize it as a maxCov problem and then, transform it to a minRed problem in Section 2.2 whose objective is to minimize the overall coverage redundancy.

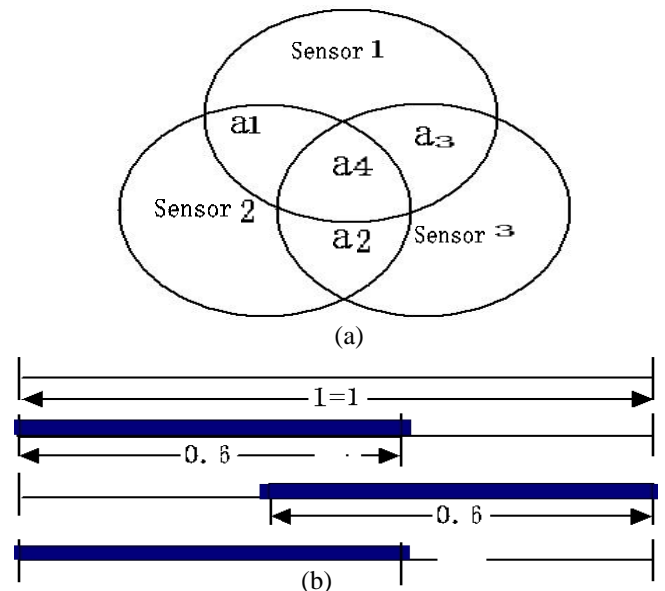


Fig 2. An example to illustrate how to calculate the redundancy for k-redundant elementary regions

## 2.1 Maximize the Spatial-Temporal Coverage Problem MaxCov

Given a unit-disk graph with  $n$  nodes, the battery life of each sensor  $B_i; i_1^1, 1 \dots n$ , and a mission lifetime of  $L$ , where  $B_i \leq L$ , we want to calculate an “on” schedule per cycle for each sensor such that the overall spatial-temporal coverage is maximized. To quantify the overall spatial-temporal coverage (or coverage, for short), we first define elementary region as the minimum region formed by the intersection of a number of sensing disks. Notice that different points belonging to the same elementary region are covered for the same length of time. Therefore, the spatial-temporal coverage of each elementary region can be calculated as the product of its area size and the length of time during which the region is covered by at least one sensor. Note that for each elementary region, the area size is fixed after the sensors are deployed, but the coverage time varies depending on the different sensor schedule. Further, define the  $k$ -redundant elementary region as the elementary region formed by the intersection of  $k$  sensors, where  $k \leq 2$ . For example, in Fig. 2a, there are seven elementary regions and four of them are redundant elementary regions, whose area sizes are  $a_1 \frac{1}{4} a_2 \frac{1}{4} a_3 \frac{1}{4} a_4 \frac{1}{4} 1$ .  $a_1, a_2$ , and  $a_3$  refer to the two-redundant elementary regions and  $a_4$  refers to the three-redundant elementary region. The non redundant elementary regions are covered by only a single sensor, such as those elementary regions other than  $a_1, a_2, a_3$ , and  $a_4$ . Since the coverage time of the non redundant elementary region is the same as the “on” period of that sensor, its spatial-temporal coverage is constant irrespective of the sensors’ schedule. Therefore, to devise a better “on” schedule per cycle for each sensor, we only need to focus on the redundant elementary regions to maximize their total spatial-temporal coverage. Given the schedule in Fig. 2b, the spatial-temporal coverage of the two-redundant elementary region can be calculated similar to that of Fig. 1. For example, the spatial temporal coverage for  $a_1$  is the product of the area of  $a_1$  and the time during which  $a_1$  is covered by either  $s_1$  or  $s_2$ , or both, i.e.,  $1 \times 1 = 1$ . Similarly, the coverage for  $a_2$  and  $a_3$  is  $1 \times 1 = 1$  and  $1 \times 0.6 = 0.6$  respectively. For the three redundant elementary region  $a_4$ , we need to find out the length of time during which it is covered by at least one of the three sensors, which is 1 time unit in Fig. 2. Therefore, the total spatial-temporal coverage over all the redundant elementary regions in Fig. 2 is  $1+1+0.6+1 = 0.6$ . In general, we can formalize the problem in the form of mathematical programming. Before giving the

formulation, we first define some notations that will be used throughout the paper.

## 2.2 Minimize the Coverage Redundancy

In this section, we consider the coverage maximization problem from another perspective and propose a new formulation. In the previous section, the objective is to maximize the total spatial-temporal coverage, which desires the total coverage time of each redundant elementary region to be as large as possible. Alternatively, we can achieve the same goal by minimizing the schedule overlap of the sensors that monitor the same redundant elementary region. Toward this direction, we propose another metric, spatial-temporal coverage redundancy, whose value depends on the area size, the overlapping “on” periods, and the number of sensors that monitor the area in each period. With the concept of spatial-temporal coverage redundancy (or coverage redundancy, for short), the problem of “maximizing coverage under the constraint of network lifetime” becomes “minimizing the coverage redundancy under the constraint of network lifetime” (called minRed problem). We can prove that the two objectives are equivalent under certain conditions. We first use Fig. 2 as an example to illustrate how to calculate the coverage redundancy of the redundant elementary regions. For instance, the coverage redundancy for  $a_1$  is the area of  $a_1$  times the schedule overlap of  $s_1$  and  $s_2$ , i.e.,  $1 \times 0.2 \frac{1}{4} 0.2$ . Similarly, the redundancy for  $a_2$  and  $a_3$  is 0.2 and 0.6, respectively. The coverage redundancy of  $a_4$  consists of two parts, i.e., the part of time when  $a_4$  is covered by exactly two sensors, and the part of time when it is covered by exactly three sensors. Intuitively, the two parts should have different contribution to the coverage redundancy, because more resources will be wasted as more sensors overlap in time. To reflect this, we assign different weight to different periods during which the same region is monitored by different number of sensors. In particular,  $a_4$  is solely monitored by  $s_1$  and  $s_2$  for 0 unit of time, by  $s_1$  and  $s_3$  for 0.4 unit of time, by  $s_2$  and  $s_3$  for 0 unit of time, all of which are assigned weight 1. On the other hand,  $a_4$  is solely monitored by  $s_1, s_2$ , and  $s_3$  for 0.2 units of time, and it is assigned weight 2. Then, the total coverage redundancy is the weighted sum of the product of area size and time overlap over all the redundant elementary regions.

**Theorem 1:** With the same graph, network lifetime requirement, and battery constraints, the objective to maximize the total spatial-temporal coverage is equivalent to minimize the total spatial-temporal

coverage redundancy when setting the weight factor to be  $w(j)=j-1$ . We first rewrite the objective of total coverage, decomposing the coverage time  $T_i$  into a multitude of sub periods according to the different coverage degree. This theorem implies that maximizing the total coverage  $C$  is equivalent to minimizing the total coverage redundancy  $R$ .

### 3. Distributed Algorithm Design

From the above discussion, we know that in a complex network of large scale, it is computationally infeasible to enumerate each elementary area  $a_i$  and list each period during which area  $i$  is covered by exactly  $j$  sensors. Therefore, in the distributed design, we focus on the pair-wise sensors and let each node minimize its own local coverage redundancy, defined as the sum of pair-wise redundancy with its neighbors. Although the global optimal is computationally infeasible to achieve, we can design a class of algorithms in which each node is able to achieve the local optimal if certain conditions can be satisfied. The basic idea is to let each node first generate a random schedule independently. Then, each node adjusts its schedule individually to minimize the local coverage redundancy with its neighbors, until everyone converges to its local optimality. The seemingly simple idea has several challenges.

- How to do the local optimization?
- Does it have polynomial time algorithms to achieve the local optimal?
- If each sensor adjusts the schedule individually, is the algorithm able to converge?
- How to eliminate conflicts caused by simultaneous adjustments of the neighboring nodes?

The following sections will address these challenges one by one.

#### 3.1 Local Optimization:

Each node has its own reference cycle. The cycles at different nodes are not required to be synchronized. Each node only needs to know the relative position of its neighbor's on-period. This can be easily achieved via exchange of hello packets with its neighbors. In our solution, we only focus on some crucial points, which could jointly determine the redundancy at every possible value.

#### 3.2 Convergence Property:

In our distributed algorithm, each node locally optimizes its own schedule as long as its schedule

does not remain locally optimal. Since altering a node's schedule can affect the redundancy of its neighbors, the schedule adjustment at different nodes may conflict with each other and the adjustment process may never end. For example, if two neighboring nodes adjust their own schedules at the same time, they may not be aware that their neighbor's schedule has been changed and cannot achieve local optimality. Next, we provide guidelines to guarantee that each node can converge to its local optimality.

**Theorem:** Given a graph  $G$  and arbitrary schedules, a distributed algorithm will terminate in a finite number of steps and after termination, each node's schedule will converge to the local optimality, if, no neighboring nodes optimize their schedules at the same time; each node's local adjustment continues as long as its local objective can be improved for at least a predefined threshold.

### 4. Distributed Protocol Design

Theorem tells us that for a distributed protocol to converge, all three conditions have to be satisfied. Before presenting our distributed protocol, let's see two simple algorithms: . Random Algorithm: each node generates a random schedule individually. Serial Optimization Algorithm: each node first generates a random schedule, based on which the schedule is locally optimized one by one. This serial optimization process is repeated until no improvement can be made beyond the predefined threshold. Each of the above algorithms has its pros and cons. The random algorithm is simple, distributed, and has no message complexity. It can serve as a baseline for comparison. The serial optimization algorithm uses the Line Traversal Algorithm as a functional module to ensure that every node can achieve its local optimality, but it is centralized. In addition, for the serial algorithm to converge, much iteration are needed until no improvement can be made. Therefore, the serial algorithm takes a long time to terminate. To retain the merit of the serial algorithm and remedy its weakness, we propose a optimizing protocol.

#### 4.1 Line Traversal Algorithm

The basic idea of optimizing protocol is to let many nodes locally optimize their schedules (using Line Traversal Algorithm) in parallel, so that it can converge much faster than the serial algorithm. According to Theorem 4, a set of no neighboring nodes can adjust their own schedules simultaneously without causing any conflict. From the algorithmic

point of view, to search for such a set of no neighboring nodes is equivalent to finding an Independent Set, which is defined as a subset of nodes among which there is no edge between any two nodes. The set is a maximal independent set if no more edges can be added to generate a bigger independent set.

#### 4.2 Maximal Independent Set Algorithm

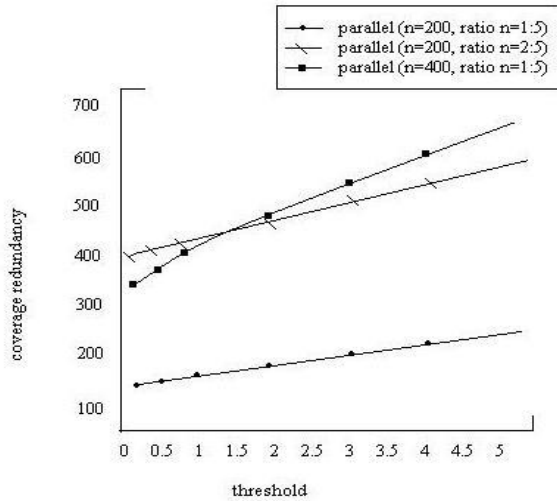
To find the maximal independent set, each node independently determines whether it belongs to the set by comparing its weight with its neighbors. If it has the best weight in the neighborhood, it elects itself as belonging to the set, and then, no other neighbors can be chosen. In general, the algorithm can be denoted as maximal independent set weight; criteria, where the weight can be id, degree, energy, etc., and the criteria can be either smallest or largest. The criteria are used to interpret the meaning of best weight, i.e., the smallest or the largest. Algorithm 2 lists the pseudo code of the optimizing protocol which can be implemented in a distributed manner. For clarity of presentation, we first introduce the protocol in a centralized manner, and then, give guidance to its distributed operation. Initially, all nodes are unlabeled. Then, each node individually determines whether it belongs to the maximal independent set by comparing its weight with the neighbors. The labeled nodes locally optimize their schedules, after which the maximal independent set algorithm will continue to run among the remaining unlabeled nodes. We term the time a round if during this period, an maximal independent set is found and local optimization is executed in parallel at the nodes of the maximal independent set. Several rounds comprise an iteration during which the coalition of the maximal independent set elected can have all the nodes labeled. The maximal independent set algorithm continues to run round after round and iteration after iteration until no improvements can be made to any node's schedule. At the end of iteration, all nodes' labels are removed and a new iteration starts with the criteria reversed, i.e., "smallest" becomes "largest" and vice versa. Therefore, the iterations alternate between the increasing and decreasing order of weight in executing the maximal independent set algorithm. The criteria are reversed to facilitate the distributed operation, so that the nodes belonging to the maximal independent set in the last round of previous iteration can start a new iteration.

### 5. Performance Evaluation

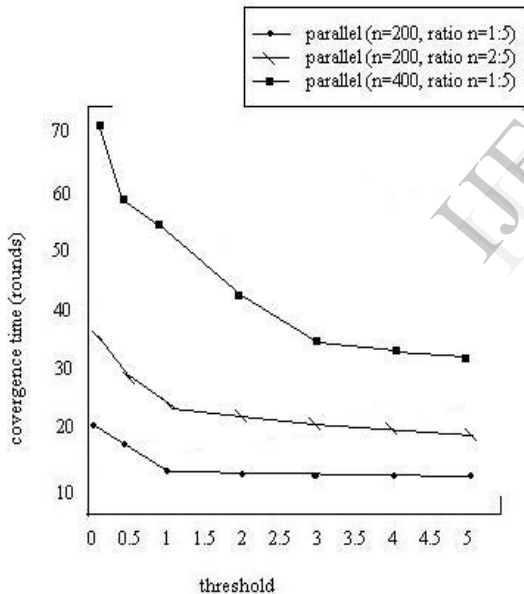
In this section, we evaluate the performance of the proposed algorithms. In the simulation, a 10 x 10 square area is considered, with  $n$  varying from 100 to 500. The sensing range is 1 unless otherwise specified. We assume the BN has two level transmission power with the transmission radii  $r$  and  $R = 2r$ , respectively. First, we deploy 100 sensor nodes randomly and the transmission radius  $r$  is set to 15 meters. For example, when  $n = 100$  and the battery/network lifetime ratio is  $\frac{3}{5}$ . Both homogeneous and heterogeneous battery states are considered. In the homogeneous case, every node has the same battery/network lifetime ratio  $v$ , but in heterogeneous case  $v_i$  is a random variable uniformly distributed in  $[v/2, 3v/2]$  with  $v$  as the average ratio. Three schemes are evaluated, namely, random, serial, and optimizing protocol, in terms of coverage redundancy, convergence time, and event detection probability. As the global coverage/coverage redundancy is infeasible to compute, we use the sum of local coverage redundancy as an approximation. The randomized event is considered whose location of occurrence is uniformly distributed in time and space, and whose length of occurrence  $e$  is normalized as the event/cycle ratio. The event detection probability is calculated by simulating 1000 randomized events. To compare with the existing schemes, we implement an extended version of the Coverage Configuration Protocol, which is shown to outperform other schemes in most of the scenarios. While the objective of the original Coverage Configuration Protocol is to select the minimum number of sensors to provide the full coverage, we extended it to a continuously operational case where the sensor node may die of limited battery. After a sensor dies, each sleeping sensor needs to decide whether it should be activated to remedy the coverage hole based on the eligibility. We evaluate Coverage Configuration Protocol in terms of coverage redundancy and network lifetime. The network lifetime is defined as the period during which half of the nodes fail.

#### 5.1 Determine the optimization threshold

The threshold of improvement made at each step. It determines how accurate the algorithm can approach the local optimality and how fast the algorithm can converge.



**Fig. 5. Relationship between the coverage redundancy (homogeneous)**



**Fig. 6. Relationship between the convergence time (homogeneous)**

From Figs. 5, 6, it can be seen that affects the coverage redundancy and the convergence time in different ways. As threshold increases, the redundancy will rise but the convergence time goes down. In other words, the objectives of redundancy and convergence time conflict with each other from the perspective. To make the coverage redundancy better, a smaller should be 11 used, but to improve the convergence time, a larger should be employed.

To balance coverage redundancy and convergence time, we set to be 1 in the following experiments

## 6. Discussion and Future work

In this paper, we assume that the disk sensing model is used, where the sensing range is modeled by a disk and a point is covered if and only if it falls within the sensing disk of one of the sensors. While the disk model provides valuable high-level guidelines, it may not accurately reflect the performance in reality. Recently, some researchers have started to investigate the impact of link irregularity and the corresponding non disk model on the performance of the sensor networks. For example, the work in employs an empirical approach to estimate the sensing ranges. A probability model is used in to depict the coverage property of the sensor network, where the coverage probability of a point depends on the distance from the monitoring sensors. To adapt the optimizing protocol to the non disk model, we can leave the big framework intact but change the method to calculate the local coverage redundancy. The algorithm still executes in iterations, but during each iteration, each node calculates the pair-wise coverage redundancy.

An example to illustrate the optimizing protocol specific non disk model. Taking the probability model as an example, the local coverage redundancy of node  $s_0$  can be calculated in the disk model. The new calculation is based on the polar coordinate system, with the middle point of the line connecting the pair-wise neighboring sensors as the pole. In particular, Probability is calculated based on the specific model, and  $s_0s_i$  follows. In general, extension of the coverage property to the non disk model is still an open issue in many situations. We leave the complete design and evaluation to the future work. Another issue worth of further investigation is the connectivity property of the sensor network. Although in this paper, we consider network lifetime as a constraint and connectivity is not our focus, achieving continuous connectivity is still valuable for the data delivery. It has been proved that when the communication range is at least twice the sensing range, the full coverage implies the connectivity of the sensor network. However, in our paper, we study the scenario where the sensors may not be sufficient enough to sustain both coverage and lifetime, so sometimes, coverage has to be traded for lifetime, resulting in the partial coverage. As far as we know, the condition under which the connectivity can be achieved in the partially covered sensor network is still an open issue. Although we did not solve it in this paper, we point out that this is an

interesting issue for the future research and have proposed a remedy solution in our previous work . We design a new set of routing protocols for the data delivery over the intermittently connected network. In an intermittently connected network, the network may not be physically connected at all instants, but the data can still be delivered to the destination in a store-and-forward fashion.

## 7. Conclusion

As mission-driven sensor networks usually have stringent lifetime requirement, sometimes coverage has to be traded for network lifetime. In this paper, we studied how to schedule sensor active time to maximize the spatial temporal coverage while meeting the lifetime constraint. The distributed parallel optimization protocol can ensure each node to converge to local optimality without conflict with each other. The computational complexity of optimizing protocol is only per node, where  $d$  is the maximum node degree, and its message complexity, which is linear with the number of nodes. Theoretical and simulation results showed that optimizing protocol substantially outperforms other schemes in terms of coverage redundancy, convergence time, network lifetime, and event detection probability.

## 8. References

- [1] "Maximal Lifetime Scheduling for Sensor Surveillance Systems with K Sensors to One Target" - IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 17, NO. 12, DECEMBER 2006
- [2] "Design and Analysis of Sensing Scheduling Algorithms under Partial Coverage for Object Detection in Sensor Networks" - IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 3, pp. 334-350, Mar. 2007
- [3] "An Integrated Protocol for Maintaining Connectivity and Coverage under Probabilistic Models for Wireless Sensor Networks" - Ad Hoc & Sensor Wireless Networks Vol. 7, pp. 295-323, 2009
- [4] "Exploring In-Situ Sensing Irregularity in Wireless Sensor Networks" - IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 4, pp. 547-561, Apr. 2010.
- [5] M. Cardei, M. Thai, Y. Li, and J. Wu, "Energy-Efficient Target Coverage in Wireless Sensor Networks," Proc. IEEE INFOCOM, 2005.
- [6] S. Basagni, "A distributed algorithm for finding a maximal weighted independent set in wireless networks," in *11th International Conference on Parallel and Distributed Computing and Systems (PDCS)*, 1999.
- [7] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, and C.-W. Yi, "Data fusion improves the coverage of wireless sensor networks," in *ACM Mobicom*, 2009.
- [8] C. Liu and G. Cao, "Minimizing the cost of mine selection via sensor networks," in *IEEE INFOCOM*, 2009.
- [9] J. Hwang, T. He, and Y. Kim, "Exploring in-situ sensing irregularity in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, 2009.
- [10] M. Hefeeda and H. Ahmadi, "An integrated protocol for maintaining connectivity and coverage under probabilistic models for wireless sensor networks," *Ad Hoc & Sensor Wireless Networks*, 2009.
- [11] L. Su, C. Liu, H. Song, and G. Cao, "Routing in intermittently connected sensor networks," in *IEEE ICNP*, 2008.
- [12] Y. Zou and K. Chakrabarty, "A distributed coverage and connectivity centric technique for selecting active nodes in wireless sensor networks," *IEEE Tran. Computer*, 2005.
- [13] S. Kumar, T. H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," in *ACM MOBICOM*, 2004.
- [14] S. Funkey, A. Kesselman, F. Kuhn, and Z. Lotker, "Improved approximation algorithms for connected sensor cover," *Wireless Networks*, 2007.
- [15] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsly, "Efficient energy management in sensor networks," *Ad hoc and sensor networks*, 2005.
- [16] G. S. Kasbekar, Y. Bejerano, and S. Sarkar, "Lifetime and coverage guarantees through distributed coordinate-free sensor activation," in *ACM Mobicom*, 2009.
- [17] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying wireless sensors to achieve both coverage and connectivity," in *ACM MOBIHOC*, 2006.
- [18] M. Cardei, M. Thai, Y. Li, and J. Wu, "Energy-efficient target coverage in wireless sensor networks," in *IEEE INFOCOM*, 2005.
- [19] C. Liu and G. Cao, "An multi-poller based energy-efficient monitoring scheme for wireless sensor networks," in *IEEE INFOCOM miniconference*, 2009.
- [20] "Distributed monitoring and aggregation in wireless sensor networks," in *IEEE INFOCOM*, 2010.