Efficient Self Protection Algorithms for Static Wireless Sensor Networks

Yu Wang* Xiang-Yang Li† Qian Zhang‡
*University of North Carolina at Charlotte, Charlotte, North Carolina, USA
†Illinois Institute of Technology, Chicago, Illinois, USA; Microsoft Research Asia, Beijing, China
‡Hong Kong University of Science and Technology, Hong Kong, China

***Abstract*—Wireless sensor networks have been widely used in many surveillance applications. Due to the importance of sensor nodes in such applications, certain level of protection needs to be provided to them. We study the *self protection* problem for static wireless sensor networks in this paper. Self protection problem focuses on using sensor nodes to provide protection to themselves instead of the target objects or certain target area, so that the sensor nodes can resist the attacks targeting on them directly. A wireless sensor network is p-self-protected, if for any wireless sensor there are at least p active sensors that can monitor it. The problem finding minimum p-self-protection is NP-complete and no efficient self protection algorithms have been proposed. In this paper, we provide efficient centralized and distributed algorithms with *constant approximation ratio* for minimum p-self-protection problem. In addition, we design efficient distributed algorithms to not only achieve p-self-protection but also maintain the connectivity of all active sensors. Our simulation confirms the performances of proposed algorithms.**

I. Introduction

A wireless sensor network (WSN) consists of a set of sensor nodes which spread over a geographical area. These nodes are able to perform processing as well as sensing and are additionally capable of communicating with each other. Due to its wide-range potential applications such as battlefield, emergency relief, environment monitoring, and so on, sensor network has recently emerged as a premier research topic. Since WSN has been used for many surveillance and military applications operating in hostile environments, it is necessary to provide certain level of protection or fault tolerance to the sensor network so that it can resist the attacks from outsides. In WSNs, sensors can be put in non-active status to save energy, and only active sensors perform the sensing tasks. Obviously, the denser and more active the sensors are, the better the protection for the objects or the better fault tolerance for the network. Many research activities on sensor networks are focused on how to balance the quality of protection [1]–[3] or fault-tolerance [4]–[6] or both [7], [8] with energy consumption of the sensors.

The previous research on the quality of protection is mainly focusing on coverage problems of sensor networks which study how to determine the minimum set of sensors for covering every location in the target field. Different coverage models and methods are surveyed by Cardei and Wu [9]. The coverage problem concentrates on protection of every location or certain objects in the target field. However, since the sensors themselves are also important and critical objects in the network, they also need certain level of coverage and hence protection. Thus, *self protection* problem is also an important protection problem in WSNs. Self protection problem focuses on using sensor nodes to provide protection to themselves instead of the objects or the area, so that they can resist the attacks targeting on them directly. A wireless sensor network is p-self-protected, if at any moment, for any wireless sensor (active or non-active), there are at least p active sensors that can monitor it. This is also different from the fault-tolerance problem. Since the fault-tolerance problem focuses on providing high connectivity of the network instead of providing high level protection.

The minimum 1-self-protection problem have been studied in [10]. The authors proved that finding minimum 1-self-protection is NP-complete and gave a centralized method with $2(1 + \log n)$ approximation ratio and two randomized distributed algorithms. Here n is the total number of sensors in the network. In this paper, we consider the p-self protection problem which is much more complex than the 1-self protection problem. The main contributions of this paper are follows: (1) we provide efficient centralized and distributed algorithms with *constant approximation ratio* for the minimum p-self-protection problem in sensor networks; (2) we design efficient distributed algorithms to not only achieve p-self-protection but also maintain the connectivity of all active sensor nodes; (3) we conduct extensive simulations to verify the performances of proposed algorithms.

II. System Model and Problem Formulation

System Model: Consider a sensor network consisting of a set V of n sensors distributed in a two-dimensional plane. Each sensor node has an omni-directional antenna with a fixed *transmission range*. Each sensor also has certain sensing capabilities and can cover all nodes within its *sensing range*. As in literatures, we assume that all sensors have the same transmission range and sensing range. The transmission range and the sensing range can be equal or not equal to each other. All sensors have distinctive identities (denoted by ID

The work of Yu Wang was supported in part by the National Science Foundation under Grant NeTS-NOSS-0721666 and by funds provided by the University of North Carolina at Charlotte. The work of Qian Zhang was supported in part by National 863 Program of China under Grant No. 2006AA01Z228, the Key Project of Guangzhou Municipal Government Guangdong/Hong Kong Critical Technology grant 2006Z1-D6131, and the HKUST Nansha Research Fund NRC06/07.EG01.

hereafter). To save the energy, sensors can be put into sleep (called *non-active* status). A sensor is called *active*, if it can carry out protections currently; otherwise it is called a *non-active* sensor. We then formulate the sensor network as a sensing graph $G(V, E)$ where V is the set of sensor nodes (both active and non-active) and E is the set of directed links $\overrightarrow{uv}$ between any two sensor u and v if v is inside the sensing range of u. We use n to denote the number of sensors.

The Problem: To formally define the *minimum self protection* problem, we need first define p-self-protected:

Definition 1: A wireless sensor network is p-**self-protected**, if, for any wireless sensor (active or non-active), there are at least p active sensors that can monitor it.

Definition 2: **Minimum** p-**Self-Protection** is a selected subset (denoted by MSP_p) of V to be set as active sensors such that the sensor network is p-self-protected and the number of active nodes ($|MSP_p|$) is minimized.

It is proved in [10], by connecting to the minimum set cover problem, that the minimum 1-self protection problem is NP-complete. Since the minimum 1-self protection problem is a special case of the minimum p-self protection problem, this indicates that the minimum p-self protection problem is also NP-complete. Notice that the following fact is obvious.

Fact 1: The minimum degree of the sensing graph is at least p is a necessary and sufficient condition for the existence of a p-self-protection in sensor networks.

Proof: First of all, if a node u does not have at least p sensors that can cover it, the sensor network clearly cannot provide p-protection to node u. This shows the necessary condition for p-self-protection. When every node has at least p sensors that can sense it, then a trivial solution that activates all sensors clearly provides p-self-protection to all nodes. This shows the sufficient condition. ■

Other Definitions: A subset of vertices in a graph G is an *independent set* if for any pair of vertices, there is no edge between them. It is a *maximum independent set* (MIS) if no other independent set has more vertices. A subset S of V is a *dominating set* if each node u in V is either in S or is adjacent to some node v in S. Clearly, any MIS is a dominating set. A dominating set with minimum cardinality is called *minimum dominating set* (MDS). A subset C of V is a *connected dominating set* (CDS) if C is a dominating set and C induces a connected subgraph.

III. Providing Self Protection

A. Centralized Method with Constant Approximation Ratio

A centralized method with $2(1+\log n)$ approximation ratio for the minimum 1-self-protection problem is given in [10]. Basically, they proved that the cost of the minimum 1-self-protection is at most twice of the cost of the minimum dominating set. Then, by applying the $(1 + \log n)$ approximation algorithm [11] for minimum dominating set, they achieved $2(1 + \log n)$ approximation. Their method is not easy to be extended to address p-self-protection problem. However, the $\log n$ approximation method for minimum p-self-protection can be directly derived from the approximation algorithm for *set multicover problem* [12] where each sensor need to be covered p times. In [12], there exists $(1+\log n)$ approximation algorithm for the set multicover problem.

For minimum 1-self-protection, it is also easy to get constant approximation ratio when sensing radius of all nodes are the same. This can be done by computing a *maximal independent set* (MIS) and then choose one neighbor for each node in the MIS. All nodes in MIS and their selected neighbors will be set active. It clearly is 1-self-protected since every node outside MIS is protected by a node in MIS and every node in MIS is protected by its neighbor selected. Remember, any MIS is a dominating set. The ratio of this simple method is at most 10 since for each node there are at most 5 neighboring nodes chosen in MIS [13] while there is at least one neighboring node at the optimal solution MSP_1 for minimum 1-self-protection. Thus, MIS is at most 5 times of the optimal solution. In addition, we select one node to cover every node in MIS, thus the total number of nodes selected in this method is at most 10 times of the optimal.

Algorithm 1 General Method for Minimum p-Self-Protection

```
1: Assign each node $v$ a unique rank $r(v) \in [1, n]$ and let
   $k = 1$.
2: while $k \leq p$ do
3:     Generate a MIS $M_k$ based on the rank of all nodes: a
       node is selected to the MIS if it has the largest rank
       among all its neighboring nodes.
4:     Assign a node that is not selected in MIS a rank $r(v) +
       k \times n$. For a node that has already been selected to some
       MIS, its rank will not change.
5:     $k = k + 1$.
6: end while
7: For each node $u$ that is selected in $M_i$, $1 \leq i \leq p$, we find
   a neighboring node $v$ if node $u$ has less than $p$ neighboring
   nodes in $\bigcup_{i=1}^{p} M_i$. We use $v$ to protect $u$.
8: Let $M$ be the union of all $M_i$ and all nodes $v$ that are
   used to protect nodes in $M_i$.
```

For the general p-self-protection problem, we describe our new approximation algorithm as Algorithm 1. Here, the updating of rank in Step 4 is designed for preventing the selected MISs in the early rounds to be used again in later rounds of MISs. Notice that since we assume that each node has at least p neighboring nodes, in Step 7 there always exists a neighboring node v that is not selected when u has less than p neighboring nodes in $\bigcup_{i=1}^{p} M_i$. Obviously, the time complexity of this algorithm is $O(n)$. We now prove that this algorithm is a 10 approximation too.

Theorem 2: The set M by Algorithm 1 is a valid p-self-protection, and has size at most 10 times of the optimum solution MSP_p when sensing radius of all nodes are the same.

Proof: First, the validation of the p-self-protection is obvious. For every node $u \notin \bigcup_{i=1}^{p} M_i$, it is protected by at least p MIS nodes since each round of MIS M_i has one node protecting it. Notice that during the process, the nodes already in the MIS selected before will *not* be selected to produce new MIS due to the rank. For all node $u \in \bigcup_{i=1}^{p} M_i$, it has at least $p-1$ protectors from $\bigcup_{i=1}^{p} M_i$ since it has been protected by MIS nodes in every round except the round it is selected as MIS. If u has only $p-1$ neighbor nodes in $\bigcup_{i=1}^{p} M_i$, the algorithm will add one node in Step 7 to protect u. Thus all nodes are perfectly protected by at least p active sensor nodes.

Then, we prove the approximation ratio. Remember that for each node there are at most 5 neighboring nodes chosen in each round MIS M_i, thus for each node, there are at most $5 \cdot p$ nodes selected in $\bigcup_{i=1}^{p} M_i$. For the optimal solution MSP_p of the minimum p-self-protection, there is at least p neighboring nodes active for protection. Thus, the selected MIS nodes in $\bigcup_{i=1}^{p} M_i$ is at most 5 times of the optimal solution MSP_p. Plus the one additional node added in Step 7 for each MIS node with $p-1$ protectors, the total number of nodes selected by this method is at most 10 times of the optimal. ■

B. *Distributed Method with Constant Approximation Ratio*

Centralized solution is good for sensor networks with centralized control center. However, in many applications, there is no centralized control and all sensors are self-organized. Thus, each sensor needs to make decisions based on limited information, and a simple distributed self-protection algorithm is needed. Our distributed algorithm (See Algorithm 2) is extended from the centralized one (Algorithm 1). We assume each node u maintains the following information of itself and its direct neighbors $N(u)$ in sensing graph:

- $ID(v)$, the distinctive ID of node v
- $p(v)$, the protection level of node v shows node v is already covered by $p(v)$ sensors in MIS.
- $k(v)$, the round counter of node v indicates node v is in which round of MIS construction (i.e. index i in M_i).
- $s(v)$, the status of node v shows the current role of node v, which could be one of $Undecided$, M_i, $Active$, and $Nonactive$. The union of all nodes marked $Active$ in the end of the execution of Algorithm 2 are the protection set, again denoted by M.

We also use three kinds of messages to exchange the necessary information among neighbors:

- Protect(x,y), node x uses this message to tell its neighbors that it becomes a MIS in y-th round (i.e., in M_y) and will provide protection of them. It is also used by the nodes selected to protect those MIS nodes with less than p-protection in the end of p rounds, such node x will send Protect(x,-1) to all its neighbors to claim protection.
- ReqProtection(x,y), those MIS nodes x with less than p-protection in the end of p rounds will select a neighbor y to provide protection to itself, and send y this message.
- Notice(x,y), node x uses this message to tell all its neighbors that there is an update happened at node x. Update event y can be K++, $Active$ and $Nonactive$. If y =K++, it means $k(x)$ increases by one, otherwise it means the status of node x changed to y.

The basic idea of the distributed algorithm is as follows. Initially, all nodes are in the first round and in $Undecided$ status. Since each node u has the information of its neighbors, it knows which round they are performing. Assume node u is in round r. If node u has the largest ID among all non-MIS nodes in the same round with u, it will become a node in M_r, send message Protect(u,r) to its neighbor, and enter round $r+1$. All its neighbors received the Protect message will also enter round $r+1$. Until node u and all its neighbors finish p rounds (i.e., $k(u) = p+1$ and $k(v) = p+1$ for all $v \in N(u)$), node u can begin making decision whether should be marked *active* or *non-active*. Nodes in $\bigcup_{i=1}^{p} M_i$ will be marked *active* while nodes with $Undecided$ become *non-active*. But for those MIS nodes with less than p-protection in the end of p rounds, each of them will randomly select a non-active node to protect itself and send message ReqProtection to notify that node. When the node receives this ReqProtection, it will become *active* and also notify its neighbors.

It is easy to prove the following theorem regarding the performance of this distributed algorithm. The proof is similar to the centralized one, thus we omit it here.

Theorem 3: The set M by Algorithm 2 is a valid p-self-protection, and has size at most 10 times of the optimum solution MSP_p when sensing radius of all nodes are the same.

Theorem 4: The message complexity of this distributed algorithm is $O(n)$.

Proof: We count the messages by different types: (1) messages Protect are only sent once by each node in M, thus there is at most n such messages; (2) the number of messages ReqProtection is also limited by n since only those MIS nodes with less than p-protection in the end of p rounds use them; (3) messages Notice(u,K++) can be sent at most pn times since $k(u)$ is updated at most p times for each node; (4) the number of messages Notice(u,Active) and Notice(u,Nonactive) is at most n since each node sends once in the end of p rounds. Thus, the total number of messages used by this algorithm is bounded by $O(n)$. ■

IV. Self-Protection and Connectivity

So far, we concentrate on how to select a subset of sensors to be active such that the network is p-self-protected. However, it is also important that active sensors are connected so that they can communicate with each other or report attacks to the centralized control center. In this section, we study how to select a subset of active sensors such that it forms a connected network topology providing p-self-protection. We assume that the transmission range is equal to the sensing range.

Efficient distributed algorithms for constructing connected dominating sets to form a virtual backbone were well studied [14], [15]. A subset C of V is a *connected dominating set* (CDS) if C is a dominating set and C induces a connected subgraph. Consequently, the nodes in C can communicate with each other without using nodes in $V - C$. A connected dominating set with minimum cardinality is the *minimum connected dominating set* (MCDS). Finding the MCDS is NP-complete, but a constant approximation ratio can be easily achieved when the underlying graph is a unit disk graph, i.e., all sensors have the same transmission ranges. One efficient

Algorithm 2 Distributed Algorithm for Minimum p-Self-Protection at node u

```
1: Initialization: let protection level $p(u) = 0$, status $s(u) = Undecided$, round $k(u) = 1$.
   {Line 2-8: if node $u$ is ready to become a MIS}
2: if $s(u) = Undecided$ then
3:     if there exists some $v \in N(u)$ such that $k(u) = k(v)$ and $ID(u) > ID(v)$ for all such $v$ then
4:         $u$ becomes a MIS in $M_{k(u)}$, i.e., $s(u) = M_{k(u)}$
5:         $u$ sends message Protect(u,k(u))
6:         $k(u) = k(u) + 1$
7:     end if
8: end if
   {Line 9-21: if node $u$ has finished $p$-rounds}
9: if $k(u) = p + 1$ and $k(v) = p + 1$ for all $v \in N(u)$ then
10:    if $s(u) = M_i$ such that $i \in [1, p]$ then
11:        if $p(u) < p$ then
12:            randomly select one neighbor $v$ whose status $s(v) = Nonactive$.
13:            send message ReqProtection(u,v) to $v$
14:        end if
15:        $s(u) = Active$
16:        send message Notice(u,Active)
17:    else if $s(u) = Undecided$ then
18:        $s(u) = Nonactive$
19:        send message Notice(u,Nonactive)
20:    end if
21: end if
   {Line 22-33: node $u$ is noticed being protected}
22: if receive message Protect(x,y) then
23:    $p(u) = p(u) + 1$
24:    if $k(u) = y$ then
25:        $k(u) = k(u) + 1$
26:        send message Notice(u,K++)
27:    end if
28:    if $y = -1$ then
29:        update the local copy of $s(x) = Active$
30:    else
31:        update the local copy of $s(x) = M_y$ and $k(x) = y+1$
32:    end if
33: end if
   {Line 34-39: node $u$ is asked to protect node $x$}
34: if receive message ReqProtection(x,y) then
35:    if $u = y$ then
36:        $s(u) = Active$
37:        $u$ sends message Protect(u,-1)
38:    end if
39: end if
   {Line 40-46: update the information from node $x$}
40: if receive message Notice(x,y) then
41:    if $y =$K++ then
42:        update the local copy of $k(x) = k(x) + 1$
43:    else
44:        update the local copy of $s(x) = y$
45:    end if
46: end if
```

way [13] to build connected dominating set is first selecting a maximal independent set (which is also a dominating set), then for each MIS node finding some *connectors* (or called *gateways*) to connect them into a backbone.

To achieve both connectivity and p-self-protection, we can apply the algorithm finding connectors for MIS in [13] on the first round MIS M_1 generated in Algorithm 2, so that these connectors can connect M_1 into a CDS. In the end of the algorithm, we will also set these connectors *active*, i.e., they also belong to the final set M. Notice that [13] proved that the total number of connectors introduced is at most constant factor of the number of MIS nodes. Thus, the approximation ratio of M for MSP is still a constant. Due to space limit, we do not review the detail algorithm for finding the connectors. The reader can find it in [13] (as Algorithm 1 there).

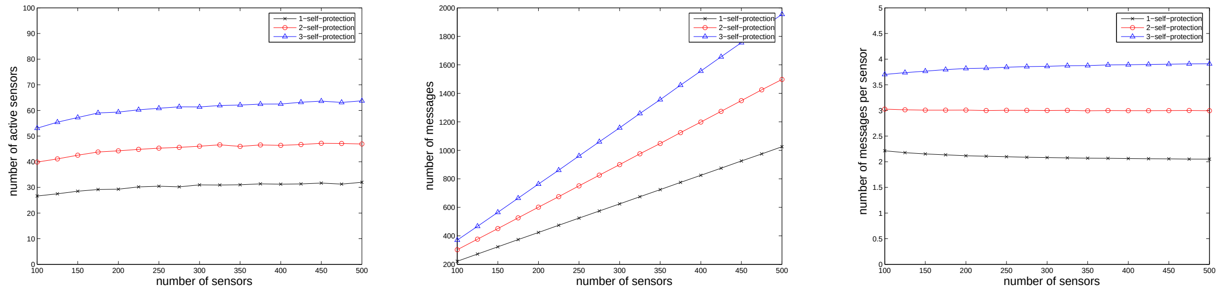
Generally, we would like to design a method to find a set of active sensors that can provide both p-self-protection and k-connectivity backbone for routing such that the size of the set is within a constant factor of the optimum. In the remainder of this section, we provide a general theorem about a general method that can achieve both p-self-protection and k-connectivity simultaneously. Our general method will first apply the best method (say with approximation ratio α_1) to find a backbone $\mathcal{B}$ that is k-connected, and apply the best method (say with approximation ratio α_2) to find a set $\mathcal{P}$ of active sensors that form p-self-protection. We then return $\mathcal{B}+\mathcal{P}$ as the solution.

Theorem 5: The size of the set of sensors $\mathcal{B}+\mathcal{P}$ is within a factor $\alpha_1 + \alpha_2$ times of the optimum set of active sensors that can provide p-self-protection and a k-connected backbone.

Proof: Since the optimum solution OPT provides p-self-protection, we have the size $|\mathcal{P}| \leq \alpha_2 |OPT|$. Since OPT also provides a backbone (not necessarily itself) that is k-connected, we have $|\mathcal{B}| \leq \alpha_1 |OPT|$. This finishes the proof due to $|\mathcal{B}| + |\mathcal{P}| \leq (\alpha_1 + \alpha_2)|OPT|$. ■

V. Simulations

In this section, we conduct extensive simulations on random networks to study the performances of our proposed algorithms. In our experiments, we randomly generated a set V of n wireless sensors and the induced sensing graph $G(V)$, then tested the connectivity and the minimum degree of $G(V)$. If it is connected and the minimum degree is larger or equal to the desired self protection level p, we construct our proposed distributed algorithm (in Section III) on $G(V)$ to select the active sensor sets supporting p-self protection and measure the total number of active sensors in these sets. Then, we apply our algorithm in Section IV to construct the connected backbone among all active sensors and provide p-self protection. In the experimental results presented here, n wireless sensors are randomly distributed in a $500m \times 500m$ square, and the sensing range and transmission range are all set to $100m$. We tested all algorithms by varying n from 100 to 500, where 50 vertex sets are generated. The average is computed over all these 50 vertex sets.



(a) average number of active sensors (b) average number of messages (c) average number of messages/sensor

Fig. 1. Results for p-self-protection ($p = 1, 2, 3$) when number of sensors increases from 100 to 500.

A. Self Protection

First, we apply Algorithm 2 to provide p-self protection to the sensor networks generated randomly. We set $p = 1, 2$ and 3. The results are plotted in Fig. 1. Fig. 1(a) shows the average number of active sensors generated by Algorithm 2. It is clear that higher self-protection level p requires more active sensors. However, for certain level p, the number of active sensors increases very slightly and slowly when the number of sensors increases. For example, for the network with 500 sensors, only 30 of them need to be activated to achieve 1-self-protection which is similar for the network with 100 sensors. Fig. 1(b) and 1(c) show the number of messages used by Algorithm 2. Notice that even the number of total messages used increases with the number of sensors, the number of messages per sensor keeps almost stable at the same low level. This confirms our message complexity analysis result $O(n)$ in Section III-B.

B. Self Protection with Connectivity

In Section IV, we studied how to select the active sensors such that the network is p-self protected and all active sensors form a connected backbone. We implement and test two methods to do so. The first method (method 1) first builds a connected dominating set (by selecting a MIS M_1 and finding connectors to connect 3-hop away sensors in M_1), then selects $p - 1$ rounds of MIS (M_i, $i \in [2, p]$), and activates one neighbor for MIS sensors with less than p protectors. The second method (method 2) first runs Algorithm 2 to achieve p-self-protection, then finds connectors to connect 3-hop away MIS sensors who are not connected by other MIS sensors yet. Fig. 2 shows the numbers of active sensors for both 1-self protection with connectivity and 2-self protection with connectivity. Notice that to achieve connectivity we need to keep more sensor active. Method 2 outperforms method 1 by activating less sensors. The reason is that many MIS sensors in M_1 are already connected by MIS sensors in later rounds since method 2 find the connectors after p-rounds of MIS. It is also clear in Fig. 2 that 2-self-protection needs more active sensors than 1-self-protection. Finally, the size of the backbone increases slightly when the network becomes denser.

VI. Conclusion

A wireless sensor network is p-self-protected, if at any moment, for any wireless sensor, there are at least p active sensors that can monitor it. The problem of finding minimum p-self-protection is NP-complete. In this paper, we gave both centralized and distributed methods that can find a p-self-protection set whose size is within at most 10 times of the optimum. We also presented efficient methods that can achieve both self protection and connectivity simultaneously.

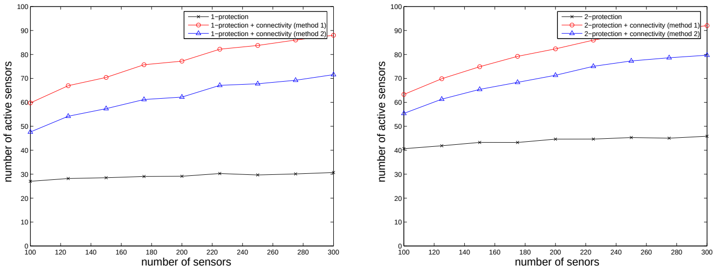


Fig. 2. Number of active sensors for p-self-protection with connectivity (Left: $p = 1$; Right: $p = 2$) when number of sensors increases.

References

[1] C. Gui and P. Mohapatra, “Power conservation and quality of surveillance in target tracking sensor networks,” in *ACM MobiCom*, 2004.

[2] S. Meguerdichian, F. Koushanfar, M. Potkonjak, *et al.*, “Coverage problems in wireless ad-hoc sensor network,” in *IEEE INFOCOM*, 2001.

[3] S. Kumar, T. H. Lai, and J. Balogh, “On k-coverage in a mostly sleeping sensor network,” in *ACM MobiCom*, 2004.

[4] X.-Y. Li, P.-J. Wan, Y. Wang, *et al.*, “Robust deployment and fault tolerant topology control for wireless ad hoc networks,” *J. on Wireless Communications and Mobile Computing*, vol.4, no.1, pp.109–125, 2004.

[5] M. Hajiaghayi, *et al.*, “Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks,” in *MobiCom*, 2003.

[6] N. Li and J.C. Hou, “FLSS: a fault-tolerant topology control algorithm for wireless networks,” in ACM MobiCom, 2004.

[7] Z. Zhou, S. Das, and H. Gupta, “Fault tolerant connected sensor cover with variable sensing and transmission,” in *IEEE SECON*, 2005.

[8] X. Bai, S. Kuma, D. Xua, *et al.*, “Deploying wireless sensors to achieve both coverage and connectivity,” in *ACM MobiHoc*, 2006.

[9] M. Cardei and J. Wu, “Energy-efficient coverage problems in wireless ad hoc sensor networks,” *Comp. Comm.*, vol.29, no.4, pp.413–420, 2006.

[10] D. Wang, Q. Zhang, and J. Liu, “Self-protection for wireless sensor networks,” in *IEEE ICDCS*, 2006.

[11] D.S. Johnson, “Approximation algorithms for combinatorial problem,” *Journal on Computer System Science*, vol. 9, pp. 256–278, 1974.

[12] V.V. Vazirani, *Approximation algorithms*, Springer, 2001.

[13] K. Alzoubi, X.-Y. Li, Y. Wang, P.-J. Wan, and O. Frieder, “Geometric spanners for wireless ad hoc networks,” *IEEE Trans. on Parallel & Distr. Sys.*, vol. 14, no. 4, pp. 408–421, 2003.

[14] S. Basagni, “Distributed clustering for ad hoc networks,” in *IEEE Int’l Symp. on Parallel Architectures, Algorithms, and Networks*, 1999.

[15] B. Das and V. Bharghavan, “Routing in ad-hoc networks using minimum connected dominating sets,” in *IEEE ICC*, 1997.