Efficient Self Protection Algorithms for Wireless Sensor Networks

Yu Wang
Dept. of Computer Science
University of North Carolina at Charlotte
Charlotte, NC 28223, USA
Email: ywang32@uncc.edu

Xiang-Yang Li
Dept. of Computer Science
Illinois Institute of Technology
Chicago, IL 60616, USA
Email: xli@cs.iit.edu

Qian Zhang
Dept. of Computer Science
Hong Kong Univ. of Science & Tech.
Hong Kong, China
Email: qianzh@cs.ust.hk

***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 need to be provided to them. In [1], Wang, Zhang and Liu first formally introduced the *self protection* problem in wireless sensor networks. 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. [1] proved that the problem finding minimum 1-self-protection is NP-complete, and gave a centralized method with $O(\log n)$ approximation ratio. Here n is the total number of sensors in the network. In this paper, we further study the p-self-protection for wireless sensor networks and discuss several aspects that have not been considered or can not be addressed in [1]. We provide efficient centralized and distributed algorithms with *constant approximation ratio* for minimum p-self-protection problem in sensor networks with either homogeneous or heterogeneous sensing radius. 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 sensor network 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. With coordination among these sensor nodes, the sensor network together achieves a larger sensing task both in urban environments and in inhospitable terrain. Due to its wide-range potential applications such as battlefield, emergency relief, environment monitoring, surveillance system, and so on, wireless sensor network (WSN) has recently emerged as a premier research topic. The sheer numbers of sensors, the limited resources on each sensor, and the expected dynamics in these environments present unique challenges in the design of WSNs.

Since wireless sensor network has been used for many surveillance applications [2], [3] and military applications operating in hostile environments, it is necessary to provide certain level of protection or fault tolerance to the senor 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 [3]–[7] or fault-tolerance [8]–[10] or both [11]–[14] 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 [15]. 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. Recently, Wang, Zhang and Liu [1] first formally introduced another important protection problem, called *self 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 with fault-tolerance problem. Since fault-tolerance problem focuses on providing high connectivity of the network (k-connectivity) instead of protection, while self protection problem does not care about connectivity issues.

In [1], Wang, Zhang and Liu proved that finding minimum 1-self-protection is NP-complete by connecting it with the well-known NP-complete problem, minimum set cover problem. Then they gave a centralized method with $2(1 + \log n)$ approximation ratio, using approximation algorithm for minimum dominating set, and two randomized distributed algorithms for the minimum 1-self protection problem. Here n is the total number of sensors in the sensor network. In this paper, we further study the minimum p-self-protection for wireless sensor networks which is much more complex than minimum 1-self protection problem. We not only improve the results in [1] but also discuss several aspects that have not been considered or can not be addressed in [1]. The main contributions of this paper are follows: (1) we provide efficient centralized and distributed algorithms with *constant approximation ratio* for minimum p-self-protection problem in sensor

networks when all sensors have the same sensing radius; (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 prove our centralized and distributed algorithms can also achieve *constant approximation ratio* for sensor networks with heterogeneous sensing radius; (4) our simulation confirms the performances of proposed algorithms.

The remainder of this paper is organized as follows. In Section II, we introduce the formal definition of the self-protection problem and the system model we used. In Section III, we present our new centralized and distributed algorithms which can achieve constant approximation ratio for the self protection problem. In Section IV, we further study how to achieve both self protection and connectivity. In Section V, we show how to achieve constant approximation ratio for self protection in sensor networks with heterogeneous sensing radius. Section VI discusses some possible improvements and variations of proposed methods. Section VII presents our simulation results and Section VIII provides an overview of the prior literature related to protection in sensor networks. Finally, a brief conclusion of our research work is highlighted in Section IX.

II. System Model and Problem Formulation

System Model: Sensors have size, weight, and cost restrictions, which impact resource availability. Thus, sensor nodes usually have limited battery resources and limited processing and communication capabilities. Consider a sensor network consisting of a set V of n wireless sensor nodes distributed in a two-dimensional plane. Each wireless sensor node has an omni-directional antenna, so that a single transmission of a node can be received by all nodes within its vicinity which is a disk centered at the node. We call the radius of this disk the *transmission range* (or *communication range*, denoted by r_t) of this sensor node. Two nodes within each other's transmission ranges can communicate directly, while two far away nodes can communicate through multi-hop wireless links by using intermediate nodes to relay the message. Each sensor node also has certain sensing or monitoring capabilities. We assume that a sensor can cover all nodes inside its sensing area which is defined by the disk centered at the sensor with radius r_s. We call r_s *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. In practice, the sensing range is usually larger than the transmission range. We also assume that all wireless sensor nodes have distinctive identities (denoted by ID 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.

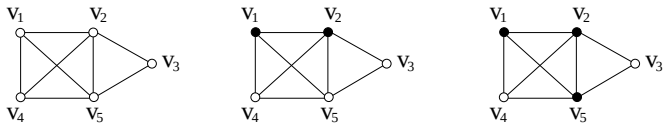
The Problem: To formally define the *minimum self protection* problem, we need first define p-self-protected:

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.

Notice that our definition is slightly different with the one in [1] where they defined being p-self-protected only needs $p-1$ active monitoring sensors. In their paper, they focused on 2-self-protection where each sensor only needs *one* active sensor to monitor it, which is called 1-self-protection by our definition in this paper. We will study the more general p-self-protection problem.

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.

Figure 1 shows examples of the minimum p-self-protection. Five sensors v_1 to v_5 form a sensing graph as shown in Figure 1(a). Subset $\{v_1, v_2\}$ achieves minimum 1-self-protection and subset $\{v_1, v_2, v_5\}$ achieves minimum 2-self-protection as shown in Figure 1.



(a) sensing graph (b) 1-self-protection (c) 2-self-protection

Fig. 1. Illustrations of minimum p-self-protection.

It is proved in [1], 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, since for each sensor we need at least p neighbors in the sensing graph to be the candidates.

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: Two definitions we will use later are *maximum independent set* (MIS) and *minimum dominating set* (MDS). 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* 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. Nodes from S are called dominators, while nodes not in S are

called dominatees. Clearly, any maximal independent set is a dominating set. A dominating set with minimum cardinality is called *minimum dominating set*. 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

In this section, we first give a centralized method to decide which set of nodes are active to provide p-self-protection, and show that this method can achieve constant approximation ratio for minimum p-self-protection problem. Later, we extend it to an efficient distributed method.

A. Centralized Method with Constant Approximation Ratio

In [1], Wang, Zhang and Liu gave a centralized method with $2(1+\log n)$ approximation ratio for the minimum 1-self-projection problem. Basically, they proved that the cost of the minimum 1-self-projection is at most twice of the cost of the minimum dominating set. Then, by applying the $(1 + \log n)$ approximation algorithm [16] for minimum dominating set, they achieved $2(1 + \log n)$ approximation. Their method is not easy to be extended to address p-self-projection problem. However, the $\log n$ approximation method for minimum p-self-projection can be directly derived from the approximation algorithm for *set multicover problem* [17] where each sensor need to be covered p times. In [17], 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 [21] 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.

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.

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.

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. For this kind of large self-organized sensor networks, it is preferred to design simple distributed method to address the self protection problem.

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 of them.
- 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 this message to y.
- 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 mark *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 notice that node. When the node receives this ReqProtection, it will become *active* and also notice 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 nodes in M, thus there is at most n such messages; (2) the number of messages ReqProtection is also limited by n since only

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 $v \in N(u)$ 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** $s(u) = M_i$ 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 send 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**

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-protection. However, in reality, it is also important that these active sensors are connected so that they can communicate with each other or they can report the centralized control center when attacks happen. Therefore, in this section, we study how to select a subset of sensors to be active such that all active sensors form a connected network topology providing p-self-protection. Notice that talking about network connectivity we need to consider the transmission range of each node. Here, 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 [18]–[20]. 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 way [21] 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 applying the algorithm finding connectors for MIS in [21] 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 [21] 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 [21] (as Algorithm 1 there).

Generally, we would like 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. Self-Protection for Sensor Networks with Heterogeneous Sensing Radius

In previous section, we assume that all sensors in the network have the same sensing radius. In this section, we will consider the sensor networks where the sensing radius of all nodes are heterogeneous and show our algorithms (Algorithm 1 and Algorithm 2) still achieve constant approximation ratios for such networks. Let each sensor u has the sensing range $r_s(u) \in [R_{\min}, R_{\max}]$. Here $R_{\max}$ and $R_{\min}$ are the maximum and the minimum sensing ranges in the network respectively. Let $\gamma = R_{max}/R_{min}$.

Theorem 6: The protection set M generated by Algorithm 1 or Algorithm 2 has size at most $12 \cdot (3\lceil \log_2 \gamma \rceil + 2)$ times of the optimum solution MSP_p when sensing radius of all nodes are heterogeneous and belong to $[R_{\min}, R_{\max}]$.

Proof: Remember for homogeneous case we prove the approximation ratio by showing that for each node there are at most 5 neighboring nodes chosen in each round MIS M_i. Here, we will show that for each node, there are at most $6 \cdot (3\lceil \log_2 \gamma \rceil + 2)$ nodes selected in each round MIS M_i. Since in each round M_i is an independent set, we only need to show that the number of independent neighbors for every node is bounded by $6 \cdot (3\lceil \log_2 \gamma \rceil + 2)$. The proof is based on a novel space partition method (Method 1) introduced in [22]. For a node v, Method 1 divides its sensing area into a constant set of regions. As shown in Figure 2(b), obviously, the number of triangle regions in each cone is $3h - 2$, where $h = 1 + \lceil \log_2 \gamma \rceil$ ($2^{h-2} < \gamma \leq 2^{h-1}$). Plus the cap region, the number of regions in each cone is at most $(3\lceil \log_2 \gamma \rceil + 2)$. Since we divide sensing range into six cones, the total number of regions is at most $6 \cdot (3\lceil \log_2 \gamma \rceil + 2)$. Lemma 7 (also Lemma 7 in [22]) shows any two nodes in a same region are connected to each other. Thus, any independent set in v's neighborhood has at most $6 \cdot (3\lceil \log_2 \gamma \rceil + 2)$ nodes.

We proved that, for each node, there are at most $6 \cdot (3\lceil \log_2 \gamma \rceil + 2)$ nodes selected in each round MIS M_i generated by our algorithms. Thus for each node there are at most $6p \cdot (3\lceil \log_2 \gamma \rceil + 2)$ nodes selected in $\bigcup_{i=1}^{p} M_i$. For the optimal solution MSP_p for 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 $6 \cdot (3\lceil \log_2 \gamma \rceil + 2)$ times of the optimal solution. Plus the one additional node added in the end of p rounds of MIS for each MIS node with $p - 1$ protectors, the total number of nodes in M selected

by our methods is at most $12 \cdot (3\lceil \log_2 \gamma \rceil + 2)$ times of the optimal. ■

Notice that actually we can improve the performance bound to $12 \cdot (3\lceil \log_2 \gamma' \rceil + 2)$ where $\gamma' = \max_{uv \in E} \frac{r_s(u)}{r_s(v)}$.

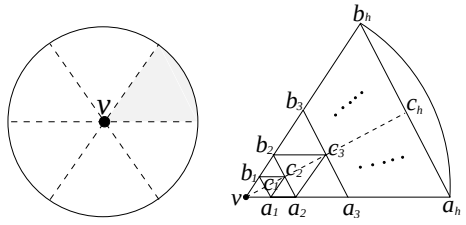


Fig. 2. Novel partition of the sensing area of node v: (a) dividing the sensing area to six cones; (b) further space partition in each cone.

Method 1: **Partition Sensing Ranges**

1: Each node v divides its sensing area into six equal cones as shown in Figure 2(a).
2: Then node v divides each cone centered at v into a limited number of triangles and caps, as illustrated by Figure 2(b), where $\|va_i\| = \|vb_i\| = \frac{1}{2^{h-i}} r_v$ and c_i is the mid-point of the segment $a_i b_i$, for $1 \le i \le h$. Here, $h = 1 + \lceil \log_2 \gamma \rceil$.
3: The triangles $\triangle va_1b_1$, $\triangle a_ib_ic_{i+1}$, $\triangle a_ia_{i+1}c_{i+1}$, $\triangle b_ib_{i+1}c_{i+1}$, for $1 \le i \le h-1$, and the cap $\overset{\frown}{a_nb_n}$ form the final space partition of each cone. For simplicity, we call such a triangle or the cap as a *region*.

Lemma 7: [22] Any two nodes u, w that co-exist in any one of the generated regions are directly connected, i.e., $\|uw\| < \min(r_s(u), r_s(w))$.

VI. Discussions

A. Further Improvements

In this subsection, we discuss several techniques that may improve the performance of our proposed algorithms.

A possible more efficient method could be as follows. Notice that the purpose of selecting MIS is to provide certain protections to nodes that are not selected into MIS. However, this may not be necessary after some rounds for some nodes when it already has p protections from selected active nodes. For example, by just one round MIS, it is possible that some node may already have upto 5 active sensors selected in the MIS. Thus, for each node u, we again use $p(u)$ to denote the protection level (*i.e.*, the number of active sensors that can sense this node) that it already has achieved via previously activated sensors from MISs. Then we have the following modified method (Algorithm 3).

Another possible improvement is that instead of random selection of a sensor to cover each active sensor in MIS, we can use a smarter method to select the nodes to protect the MIS nodes with less than p protectors in the last steps of our algorithms. Notice that the problem of adding protection to these MIS nodes is a set cover problem: each node in MISs (that has less than p-protections) is an element and each non-MIS node defines a set whose elements are all adjacent MIS nodes (with less than p-protections). To minimize the number of selected nodes in this step, we can apply the approximation algorithm for minimum set cover problem, which has several methods with approximation ratio $O(\log d)$ [23], where d is the maximum set size. Notice that for any node, there is only at most 5 neighboring MIS node, *i.e.*, $d \le 5$ for one single round MIS. Since we may have at most p rounds of MISs at the last step of our method, we have $d \le 5p$. Thus, given MISs, the additional sensors found using greedy set cover method is within $\log p$ of the smallest number of sensors needed to make this MISs set with p-self-projection property.

If we only consider the centralized algorithm for minimum 1-self-protection problem, we can produce a better solution by using the PTAS (polynomial time approximation scheme) for MIS. For example, we can use the PTAS proposed by [24] to approximate the maximum independent set when sensing radius are the same in network. Notice that the PTAS runs in time polynomial of n and can achieve $1+\epsilon$ approximation for any additional parameter $\varepsilon > 0$ for MIS. Thus, it implies a $2(1+\epsilon)$ solution for the minimum 1-self-protection problem.

Algorithm 3 Modified Method for Minimum p-Self-Protection

1: Assign each node v a unique rank $r(v) \in [1, n]$ and let $k = 1$. And assign $p(v) = 0$ for every node v.
2: **while** exist node u with $p(u) < p$ **do**
3: Let V_k be the set of nodes with $p(v) < p$, *i.e.*, nodes in V_k needs additional protections. Let U_k be the set of nodes that either is in V_k or that can sense a node from V_k, *i.e.*, U_k is the set of nodes that can provide protections to nodes in V_k.
4: Generate a MIS M_k based on the rank of all nodes in U_k: a node from U_k is selected to the MIS if it has the largest rank among all its neighboring nodes from V_k and it is not marked. Mark all nodes in M_k.
5: Assign every node that is not selected in MIS a rank $r(v) + k \cdot n$. For a node that has already been selected to some MIS, its rank will not change.
6: Update the protection $p(v)$ for every node v in V_k as $p(v) = p(v)$+number of neighboring nodes in M_k.
7: $k = k + 1$.
8: **end while**
9: For each node u that is selected in M_i, $1 \le i \le p$, 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.
10: Let M be the union of all M_i and all nodes v that are used to protect nodes in M_i.

If we only consider the centralized algorithm for minimum 1-self-protection problem, we can produce a better solution by using the PTAS (polynomial time approximation scheme) for MIS. For example, we can use the PTAS proposed by [24] to approximate the maximum independent set when sensing radius are the same in network. Notice that the PTAS runs in time polynomial of n and can achieve $1+\epsilon$ approximation for any additional parameter $\varepsilon > 0$ for MIS. Thus, it implies a $2(1+\epsilon)$ solution for the minimum 1-self-protection problem.

B. Implementation Issues

After the generation of the set of active nodes to achieve p-self-protection, dynamic maintenance of this set via updates or rotations of active/non-active roles is also an important issue during the implementation in sensor networks, since each sensor node has limited power and resources.

To balance the energy consumption, one simple method is generating certain number of p-self-protection sets and rotating the active set among these sets. Notice that our proposed

methods generate unique p-self-protection set M, however by changing the criteria of selecting the MIS we still can get several different sets M. For example, in centralized methods we can use different ranking. In localized methods, we can use criteria other than ID to select MIS nodes, such as node degree or remaining energy. Assume that, we can generate k sets M^i ($i \in [1,k]$) each of which can guarantee the p-self-protection of the network. Then how to schedule the rotations of these k sets to maximize the life time of the sensor network is also an interesting problem. Assume that set M^i will be activated for t_i seconds and each sensor v_j ($j \in [1,n]$) has limited energy can support it active for at most T_j seconds. Let $f(i,j)$ indicates whether sensor $v_j \in M^i$, $f(i,j) = 1$ if $v_j \in M^i$, otherwise $f(i,j) = 0$. Thus, the maximum life-time scheduling is equivalent to solve the linear programming $\max \sum_{i=1}^{k} t_i$ with constrains $\sum_{i=1}^{k} t_i \cdot f(i,j) \leq T_j$ for all $v_j \in V$. The solution of t_i ($i \in [1,k]$) is the size of active time lot of each p-self-protection set M^i.

Another technique to balancing the energy consumption is considering the energy as the *priority criterion* for the selection of MIS and performing our algorithm periodically with a pre-set time. In other words, we let the node with most energy remaining have higher priority to become MIS (i.e. to be active) since the active nodes will consume more energy than those non-active nodes. After certain time, the network rerun our algorithms to select a new active set based on the current energy information. The update processing is performed periodically. This way insures the energy balance throughout the network. Energy-based clustering methods have also been studied in [25]–[27] where they consider the remaining energy or energy consumption rate as the criterion. In [27], Wang *et al.* studied how to efficiently construct MIS and MCDS for weighted sensor networks.

VII. 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. Figure 3 shows two sets of examples ($n = 100$ and 300, $p = 1$ and 2) of the active sets and the backbones generated by our proposed algorithms.

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 for each case to smooth the possible peak effects. The average are computed over all these 50 vertex sets. Notice, the parameter setting of our experiments here is just for demonstrations. We have tried other various settings, the results and performances are stable, due to space limit, we can not present all of them here.

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 Figure 4. Figure 4(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. This is also illustrated in Figure 3 ((b) and (c), (g) and (h)). 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. Figures 4(b) and 4(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.

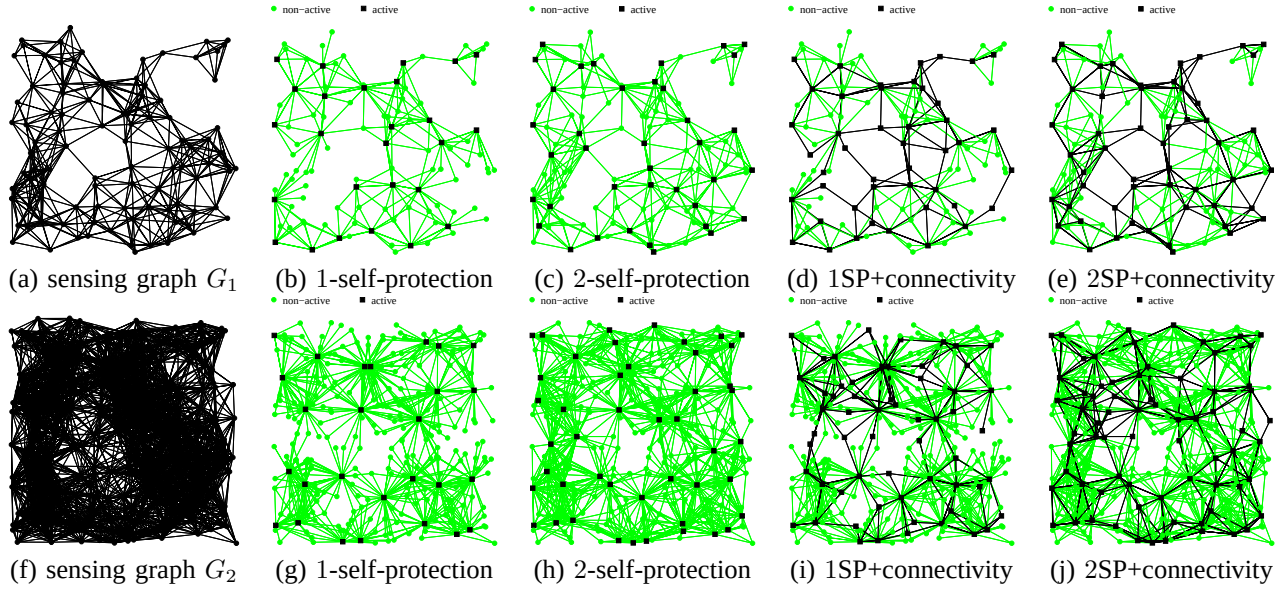
B. Self Protection with Connectivity

In Section IV, we studied how to select the active sensors such that the network is p-self protection and all active sensors form a connected backbone. Figures 3(d), 3(e), 3(i) and 3(j) illustrate the active sensors and the formed 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. Figure 5 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 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 Figure 5 that 2-self-protection need more active sensors than 1-self-protection. Finally, the size of the backbone increases slightly when the network becomes denser.

VIII. Related Work

Wireless sensor network has drawn a lot of attention recently due to its unique capability and the wide spectrum. Many research activities on sensor networks are focused on how to balance the quality of protection [4]–[7] (coverage) or fault-tolerance [8]–[10] or both [11]–[14] with energy consumption of the sensors.

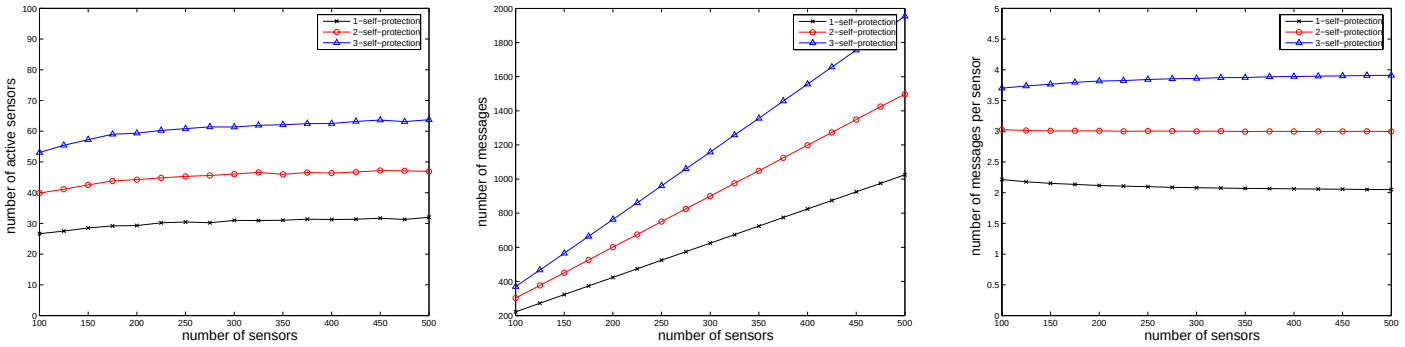
Sensor coverage is a key design issue in many sensor network applications. Cardei and Wu [15] provided a complete



(a) sensing graph G_1 (b) 1-self-protection (c) 2-self-protection (d) 1SP+connectivity (e) 2SP+connectivity

(f) sensing graph G_2 (g) 1-self-protection (h) 2-self-protection (i) 1SP+connectivity (j) 2SP+connectivity

Fig. 3. Active sets generated by our self-protection algorithms for sensing graph G_1 with 100 sensors and sensing graph G_2 with 300 sensors. Here, black squares are active nodes and gray dots are non-active nodes. Black links in (d)(e)(i)(j) are links in the backbone keeping the active sensors connected.



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

Fig. 4. Results for p-self-protection ($p = 1, 2, 3$) when number of sensors increases from 100 to 500.

survey on sensor coverage problem. The most studied coverage problem is the area coverage problem, where the main objective of the sensor network is to cover (monitor) an area, i.e., every point in the area should be covered or k-covered by sensors. Kumar *et al.* [6] studied k-coverage problem in sensor networks, and proposed a sleep/active schedule to minimize energy consumption. In [7], they considered barrier coverage where the sensors can be used as barriers. They defined the concept of k-barrier coverage (crossing a barrier of sensors will always be detected by at least k active sensors) and provided efficient algorithms to determine whether a given belt region is k-barrier covered or not. In [4], [5], the authors defined the maximal breach path and the maximal support path to measure the quality of coverage, and studied efficient methods to solve coverage problem under such measurements.

Fault tolerance is another key challenge in sensor networks. To make fault tolerance possible, network topology must have k-connectivity or multiple paths between any two wireless devices. [8], [28] studied how to set the transmission radius to achieve the k-connectivity with certain probability for a random network, while [9], [29] studied how to find small transmission range for each node such that the resulted communication graph is k-connected. [10] and [12] proposed localized algorithms to build k-connected topologies.

Until recently, coverage and connectivity problems have been studied together in sensor networks. Xing *et al.* [14] designed a integrated coverage configuration protocol to provide both certain degrees of coverage and connectivity guarantees. Zhang and Hou [11] proposed a decentralized density control algorithm to maintain sensing coverage and connectivity in high density sensor networks. Both [14] and [11] proved that if the transmission range is at least twice the sensing range, complete 1-coverage of a convex area implies connectivity among the working set of nodes. Recently, Bai *et al.* [13] studies the optimal deployment pattern to achieve both 1-coverage of an area and 2-connectivity of the sensors. Zhou *et al.* [12] proposed a set of distributed algorithms to achieve both k-connected and k-covered sensor network by using localized

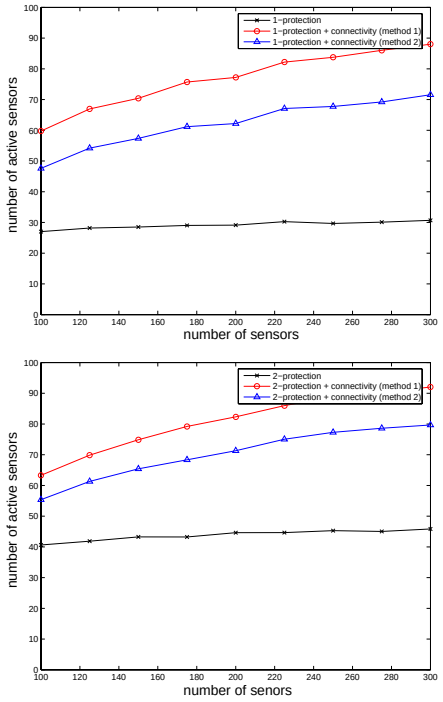


Fig. 5. Number of active sensors for p-self-protection with connectivity (Upper: $p = 1$; Lower: $p = 2$) when number of sensors increases.

Voronoi graph and extended relative neighborhood graph.

Notice that the p-self-protection problem studied here and in [1] is different with both k-coverage and k-connectivity problems. It focuses on providing p-protection to sensor nodes themselves.

IX. Conclusion

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. Wang *et al.* [1] proved that the problem finding minimum 1-self-protection is NP-complete, and gave a centralized method with $O(\log n)$ approximation ratio. 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 when the sensing ranges of all sensors are uniform. When sensing ranges are heterogeneous, we proved that our methods can find a p-self-protection set with approximation ratio $O(\log_2 \gamma)$ where γ is the ratio of the maximum sensing range over the minimum sensing range in the network. We also presented efficient methods that can achieve both self protection and connectivity simultaneously.

A number of interesting and important questions that we did not address here are left for future research. The first question is to find a small set of sensors that itself is k-connected backbone and provides p-self-protection. The second question is when the current sensors cannot provide p-self-protection, how to add the smallest number of sensors such that the new network provides p-self-protection. The third question is to find a good approximation algorithm for scheduling the active sensors such that the lifetime of the network is maximized while the active sensors always provide p-self-protection.

References

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

[2] T. He, S. Krishnamurthy, *et al.*, "Energy-efficient surveillance system using wireless sensor networks," in *ACM MobiSYS*, 2004.

[3] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," in *: ACM MobiCom*, 2004.

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

[5] X.-Y. Li, P.-J. Wan, and O. Frieder, "Coverage in wireless ad-hoc sensor networks," in *IEEE ICC*, 2002.

[6] S. Kumar, T. H. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *ACM MobiCom*, 2005.

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

[8] 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.

[9] M. Hajiaghayi, N. Immorlica, and V.S. Mirrokni, "Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks," in *ACM MobiCom*, 2003.

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

[11] H. Zhang and J.C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Wireless Ad Hoc and Sensor Networks: An International Journal*, vol. 1, no. 1-2, pp. 89–123, 2005.

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

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

[14] 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 Trans. Sen. Netw.*, vol.1, no.1, pp.36–72, 2005.

[15] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad hoc sensor networks," *Computer Communications Journal (Elsevier)*, vol. 29, no. 4, pp. 413–420, 2006.

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

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

[18] S. Basagni, "Distributed clustering for ad hoc networks," in *IEEE Int'l Symp. on Parallel Architectures, Algorithms, and Networks*, 1999.

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

[20] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks," *IEEE Trans. on Parallel & Distr. Sys.*, vol.13, no.1, pp.14–25, 2002.

[21] K. Alzoubi, X.-Y. Li, Y. Wang, P.-J. Wan, and O. Frieder, "Geometric spanners for wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Processing*, vol. 14, no. 4, pp. 408–421, 2003.

[22] X.-Y. Li, W.-Z. Song, and Y. Wang, "Localized topology control for heterogeneous wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 1, pp. 129–153, 2006.

[23] V. Chvátal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[24] H.B. Hunt III, M.V. Marathe, *et al.*, "NC-approximation schemes for NP- and PSPACE -hard problems for geometric graphs," *Journal of Algorithms*, vol.26, no.2, pp.238–274, 1998.

[25] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *IEEE 33rd Hawaii International Conference on System Sciences*, 2000.

[26] M. Chatterjee, S.K. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[27] Y. Wang, W. Wang, and X.-Y. Li, "Efficient distributed low cost backbone formation for wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 7, pp. 681–693, 2006.

[28] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *ACM MobiHoc*, 2002.

[29] R. Ramanathan and R. Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *IEEE INFOCOM*, 2000.