Efficient Algorithms for the p-Self-Protection Problem in Static Wireless Sensor Networks

Yu Wang, *Member*, *IEEE*, Xiang-Yang Li, *Member*, *IEEE*, and Qian Zhang, *Senior Member*, *IEEE*

Abstract—Wireless sensor networks have been widely used in many surveillance applications. Due to the importance of sensor nodes in such applications, a certain level of protection needs to be provided to them. We study the *self-protection* problem for static wireless sensor networks in this paper. The 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 them directly. A wireless sensor network is p-self-protected if at any moment, for any wireless sensor (active or nonactive), there are at least p active sensors that can monitor it. The problem of finding the 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 a *constant approximation ratio* for the minimum p-self-protection problem in sensor networks with either a 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 the proposed algorithms.

Index Terms—Self-protection, coverage, independent set, distributed algorithms, wireless sensor networks.

1 Introduction

A sensor network consists of a set of sensor nodes, which spread over a geographical area. These nodes are able to perform processing and 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 of potential applications such as battlefield, emergency relief, environment monitoring, surveillance system, and so on, wireless sensor networks (WSNs) [1] have 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 WSN has been used for many surveillance applications [2], [3] and military applications operating in hostile environments, it is necessary to provide a 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 nonactive 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], [4], [5], [6], [7] or fault tolerance [8], [9], [10], [11] or both [12], [13], [14], [15] 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 [16]. Dietrich and Dressler [17] also provide a survey on network lifetime, which includes various coverage problems defined in sensor networks. 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 a certain level of coverage and, hence, protection. Thus, the *self-protection* problem is also an important protection problem in WSNs. The 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 them directly. A WSN is p-self-protected if at any moment, for any wireless sensor (active or nonactive), 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 (k-connectivity) instead of providing high-level protection, while the self-protection problem does not care about connectivity issues.

Notice that the minimum self-protection problem was first introduced by Wang et al. [18], [19]. In [18], Wang et al. formulated the p-self-protection problem in sensor networks but then only focused on the study of the 1-self-protection problem. They proved that finding minimum 1-self-protection is NP-complete by reducing the minimum set cover problem. Then, they gave a centralized method with $2(1+\log n)$ approximation ratio

- *Y. Wang is with the Department of Computer Science, University of North Carolina, Charlotte, NC 28223-0001. E-mail: yu.wang@uncc.edu.*
- *X.-Y. Li is with the Department of Computer Science, Illinois Institute of Technology, 10 W. 31st St., Chicago, IL 60616-3793. E-mail: xli@cs.iit.edu.*
- *Q. Zhang is with the Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. E-mail: qianzh@cs.ust.hk.*

Manuscript received 31 Mar. 2007; revised 20 Sept. 2007; accepted 28 Dec. 2007; published online 10 Jan. 2008.
Recommended for acceptance by M. Ould-Khaoua.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2007-03-0101.
Digital Object Identifier no. 10.1109/TPDS.2008.13.

and two randomized distributed algorithms for the minimum 1-self-protection problem. Here, n is the total number of sensors in the sensor network. Their centralized method is based on a minimum dominating set algorithm in [21], while the two randomized algorithms can only achieve 1-self-protection with certain probability. Later, in an extended version [19], the same authors introduced the maximum disjoint self-protection problem, which finds the maximum number of disjoint sets of sensors such that each set can provide self-protection. They proved that it is also NP-complete. Different from [18] and [19], in this paper, we focus on the minimum p-self-protection problem, which is much more complex than the 1-self-protection problem. We propose a set of centralized and distributed algorithms with *constant approximation ratios* for the minimum p-self-protection problem. Our proposed distribution algorithms can efficiently select the active sensors to self-protect the network with small overhead.

The main contributions of this paper are summarized as follows:

1. We provide efficient centralized and distributed algorithms with *constant approximation ratio* for the 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 that our centralized and distributed algorithms can also achieve a *constant approximation ratio* for sensor networks with a heterogeneous sensing radius.
4. We conduct extensive simulations to verify the performances of the proposed algorithms.

The remainder of this paper is organized as follows: In Section 2, we introduce the formal definition of the self-protection problem and the system model we used. In Section 3, we present our new centralized and distributed algorithms, which can achieve a constant approximation ratio for the self-protection problem. In Section 4, we further study how to achieve both self-protection and connectivity. In Section 5, we show how to achieve a constant approximation ratio for self-protection in sensor networks with a heterogeneous sensing radius. Section 6 discusses some possible improvements and variations of the proposed methods. Section 7 presents our simulation results, and Section 8 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 9. A preliminary conference version of this article appeared in [20]. This version contains new self-protection algorithms for sensor networks with a heterogeneous sensing radius and better overall presentation.

2 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 static sensor network consisting of a set V of n wireless sensor nodes distributed in a two-dimensional plane. Each wireless sensor node has an omnidirectional 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 multihop 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 the *sensing range.* As in the literature, 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 could be larger or smaller than the transmission range, depending on the type of sensors. 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 *nonactive* status). A sensor is called *active* if it can carry out protections currently; otherwise, it is called a *nonactive* 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 nonactive), 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 to first define p-self-protected:

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

Notice that our definition is slightly different from the one in [18], where they defined that 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.*

Fig. 1 shows examples of the minimum p-self-protection. Five sensors v_1 to v_5 form a sensing graph, as shown in Fig. 1a. 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 Fig. 1.

It is proved in [18], 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.

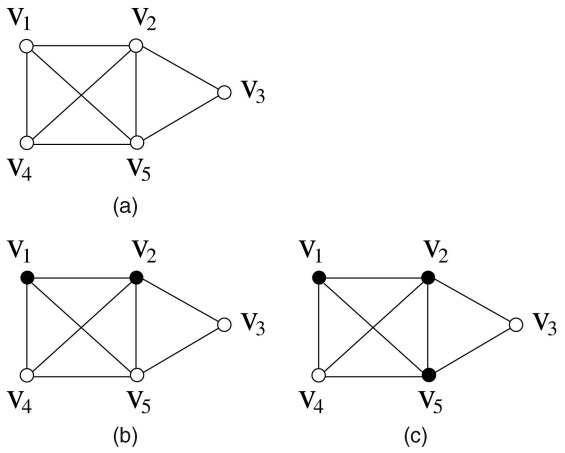


Fig. 1. Illustrations of minimum p-self-protection. (a) Sensing graph. (b) 1-self-protection. (c) 2-self-protection.

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 condition that 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 the *maximum independent set* (MIS) and the *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 an MIS if no other independent set has more vertices. Notice that an MIS is different from the *maximal independent set*. A subset of vertices is a *maximal independent set* if no additional vertices can be added into the subset while it is still an independent set. A subset S of V is a *dominating set* if each node u in V is either in S or 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 an MDS. A subset C of V is a *connected dominating set* (CDS) if C is a dominating set and C induces a connected subgraph.

3 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 a constant approximation ratio for the minimum p-self-protection problem. Later, we extend it to an efficient distributed method.

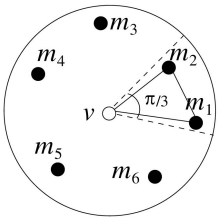


Fig. 2. For a node v, there are at most five MIS nodes in its neighborhood. Proof: Assume that there are six MIS nodes m_i, $1 \le i \le 6$. There must be two nodes (say, m_1 and m_2) falling in a cone with a $\pi/3$ angle, i.e., $\angle m_1 v m_2 \le \pi/3$. Since both m_1 and m_2 are inside the range of v and all sensors have the same range, m_1 and m_2 are in the range of each other. This is a contradiction with the definition of MIS.

3.1 Centralized Method with Constant Approximation Ratio

A centralized method with a $2(1+\log n)$ approximation ratio for the minimum 1-self-protection problem is given in [18]. Basically, they proved that the cost of the minimum 1-self-protection is at most twice of the cost of the MDS. Then, by applying the $(1+\log n)$ approximation algorithm [21] for MDS, they achieved $2(1+\log n)$ approximation. Their method is not easy to be extended to address the p-self-protection problem. However, the $\log n$ approximation method for minimum p-self-protection can be directly derived from the approximation algorithm for the *set multicover problem* [22], where each sensor needs to be covered p times. In [22], there exists $(1+\log n)$ approximation algorithm for the set multicover problem.

For the minimum 1-self-protection, it is also easy to get a constant approximation ratio when the sensing radius of all nodes is the same. This can be done by computing a *maximal independent set* (denoted by M_0) and then choosing one neighbor for each node in M_0. All nodes in M_0 and their selected neighbors will be set active, denoted by M. It clearly is 1-self-protected since every node outside MIS M_0 is protected by a node in M_0 and every node in M_0 is protected by its selected neighbor. Remember that any MIS is a dominating set. The ratio of this simple method is also bounded by a constant 10.

Theorem 1. *The set M by this simple method has a size at most 10 times of the optimum solution MSP_1 when the sensing radius of all nodes is the same.*

Proof. Consider each node v; there are at most five neighboring nodes chosen in MIS M_0 [29], since if there are six neighboring nodes of v in M_0, then at least two of them will be a neighbor of each other, which contradicts the definition of MIS. See Fig. 2 for illustration. On the other hand, there is at least one neighboring node of v at the optimal solution MSP_1 of the minimum 1-self-protection problem to guarantee the protection of v. Thus, the size of the MIS M_0 is at most five times of the size of the optimal solution MSP_1. In addition, we select one node to cover every node in M_0; thus, the total number of nodes selected in M by this method is at most 10 times of the optimal $|MSP_1|$. □

For the general p-self-protection problem, we describe our new approximation algorithm as Algorithm 1. The basic idea of the algorithm is given as follows: The algorithm first generates k MISs in k rounds. In each round, an MIS M_k is

generated based on the ranks of nodes. Here, the updating of rank in step 4 is designed to prevent the selected MISs in the early rounds to be used again in later rounds of MISs. After k MISs are generated, all nodes in these MISs will be in the active set M. For each node u inside these MISs, if it has less than p neighbors in the MISs, the algorithm adds a neighbor v into M. 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.

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 an 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.

Theorem 2. *The set M by Algorithm 1 is a valid p-self-protection and has a size at most 10 times of the optimum solution MSP_p when the sensing radius of all nodes is 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 the 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 five neighboring nodes chosen in each round of 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 are at least p neighboring nodes active for protection. Thus, the number of the selected MIS nodes in $\bigcup_{i=1}^{p} M_i$ is at most five times of the size of the optimal solution MSP_p. Adding 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. □

3.2 Distributed Method with Constant Approximation Ratio

A centralized solution is good for sensor networks with a 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 a 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 that each node u maintains the following information of itself and its direct neighbors $N(u)$ in the sensing graph:

- $ID(v)$, the distinctive ID of node v,
- $p(v)$, the protection level of node v that shows node v is already covered by $p(v)$ sensors in MIS,
- $k(v)$, the round counter of node v that indicates node v is in which round of MIS construction (i.e., index i in M_i),
- $s(v)$, the status of node v that shows the current role of node v, which could be one of $Undecided$, M_i, $Active$, and $Nonactive$ (the union of all nodes marked $Active$ at 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 an MIS in the yth round (i.e., in M_y) and will provide protection to them. It is also used by the nodes selected to protect those MIS nodes with less than p-protection at 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 at the end of p rounds will select a neighbor y to provide protection to themselves and send this message to y.
- Notice(x, y). Node x uses this message to tell all its neighbors that there is an update that happened at node x. Update event y can be K++, $Active$, and $Nonactive$. If $y = \text{K++}$, it means that $k(x)$ has increased by one; otherwise, it means the status of node x has changed to y.

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 an MIS}
2: **if** $s(u) = Undecided$ and $k(v) \geq k(u)$ for all $v \in N(u)$ **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 an 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 ∈ N(u) then
10:   if s(u) = M_i such that i ∈ [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
```

The basic idea of the distributed algorithm is given as follows: Initially, all nodes are in the first round and in the *Undecided* status. Since each node u has the information of its neighbors, it knows which round they are performing. Assume that node u is in round r and *Undecided*. If all its neighbors are already in round r and it has the largest ID among all non-MIS nodes in the same round, it will become a node in M_r, send message Protect(u, r) to its neighbor, and enter round $r+1$. All its neighbors that 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 a decision whether it should be marked *active* or *nonactive*. Nodes in $\bigcup_{i=1}^{p} M_i$ will be marked *active*, while nodes with *Undecided* will become *nonactive*. However, for those MIS nodes with less than p-protection at the end of p rounds, each of them will randomly select a nonactive 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 a size at most 10 times of the optimum solution MSP_p when the sensing radius of all nodes is 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 are 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 at 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 at the end of p rounds.

Thus, the total number of messages used by this algorithm is bounded by $O(n)$. □

4 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, in reality, it is also important that these active sensors are connected so that they can communicate with each other or they can report to 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 CDSs to form a virtual backbone were well studied [23], [24], [25], [26], [27], [28], [29]. A subset C of V is a 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 CDS 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 [29] to build a CDS is first selecting a maximal independent set (which is also a dominating set), and 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 for finding connectors for MIS in [29] on the first round MIS M_1 generated in Algorithm 2 so that these connectors can connect M_1 into a CDS. At the end of the algorithm, we will also set these connectors *active*, i.e., they also belong to the final set M. Notice that [29] proved that the total number of connectors introduced is at most a constant factor of the number of MIS nodes. Thus, the approximation ratio of M for MSP is still a constant. Due to space limitations, we do not review the details of the algorithm for finding the connectors. The reader can find it in [29, Algorithm 1].

Generally, we would like to design a method to find a set of active sensors that can provide both p-self-protection and a k-connected 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. Notice that there are several methods [30], [31] that have been proposed to construct a k-connected dominating set. For example, the method in [31] can achieve a constant approximation ratio for the minimum 2-CDS problem.

Theorem 5. *The size of the set of sensors $\mathcal{B}+\mathcal{P}$ is within a factor $\alpha_1+\alpha_2$ 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 $|\mathcal{P}| \le \alpha_2|OPT|$. Since OPT also provides a backbone (not necessarily itself) that is k-connected, we have $|\mathcal{B}| \le \alpha_1|OPT|$. This finishes the proof due to $|\mathcal{B}|+|\mathcal{P}| \le (\alpha_1+\alpha_2)|OPT|$. □

5 Self-Protection for Sensor Networks with Heterogeneous Sensing Radius

In Section 4, 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 is heterogeneous and show that our algorithms (Algorithms 1 and 2) still achieve constant approximation ratios for such networks. Let each sensor u have 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 a size at most $12 \cdot (3\lceil \log_2 \gamma \rceil + 2)$ times of the optimum solution MSP_p when the sensing radius of all nodes is heterogeneous and belongs to $[R_{\min}, R_{\max}]$.*

Proof. Remember that for the homogeneous case, we prove the approximation ratio by showing that for each node, there are at most five neighboring nodes chosen in each round of 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 of 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 [32]. For a node v, Method 1 divides its sensing area into a constant set of regions. As shown in Fig. 3b, obviously, the number of triangle regions in each cone is $3h-2$, where $h = 1 + \lceil \log_2 \gamma \rceil$ ($2^{h-2} < \gamma \le 2^{h-1}$). Adding the cap region, the number of regions in each cone is at most $(3\lceil \log_2 \gamma \rceil + 2)$. Since we divide the sensing range into six cones, the total number of regions is at most $6 \cdot (3\lceil \log_2 \gamma \rceil + 2)$. Lemma 7 [32, Lemma 7] shows that any two nodes in the 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.

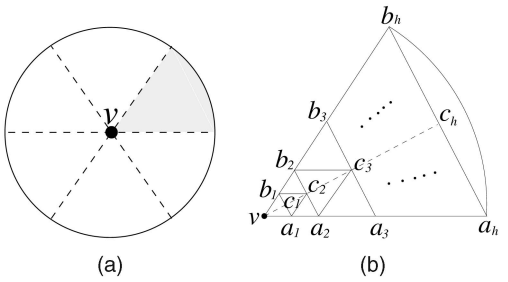


Fig. 3. Novel partition of the sensing area of node v. (a) Dividing the sensing area to six cones. (b) Further space partition in each cone.

We proved that for each node, there are at most $6 \cdot (3\lceil \log_2 \gamma \rceil + 2)$ nodes selected in each round of 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 are at least p neighboring nodes active for protection. Thus, the selected MIS nodes in $\bigcup_{i=1}^{p} M_i$ are at most $6 \cdot (3\lceil \log_2 \gamma \rceil + 2)$ times of the optimal solution. Adding the one additional node added at 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 $\gamma' = \max_{uv \in E} \frac{r_s(u)}{r_s(v)}$.

Method 1. **Partition sensing range of node v**

1: Each node v divides its sensing area into six equal cones, as shown in Fig. 3a.
2: Then, node v divides each cone centered at v into a limited number of triangles and caps, as illustrated in Fig. 3b, where $\|va_i\| = \|vb_i\| = \frac{1}{2^{h-i}} r_v$, and c_i is the midpoint of the segment a_ib_i, for $1 \le i \le h$. Here, $h = 1 + \lceil \log_2 \gamma \rceil$.
3: The triangles $\triangle va_1b_1$, $\triangle a_ib_ic_{i+1}$, $\triangle a_ia_{i+1}c_{i+1}$, and $\triangle b_ib_{i+1}c_{i+1}$, 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 [32]. *Any two nodes u and w that coexist in any one of the generated regions are directly connected, i.e., $\|uw\| < \min(r_s(u), r_s(w))$.*

6 Discussions

6.1 Further Improvements

In this section, 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 the 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 of MIS, it is possible that some node may already have up to five 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).

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 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 an 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.

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 the minimum set cover problem, which has several methods with approximation ratio $O(\log d)$ [33], where d is the maximum set size. Notice that for any node, there are only at most five neighboring MIS nodes, i.e., $d \le 5$, for one single round of 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 the greedy set cover method is within $\log p$ of the smallest number of sensors needed to make this MIS set with p-self-protection property.

If we only consider the centralized algorithm for the minimum 1-self-protection problem, we can produce a better solution by using the polynomial-time approximation scheme (PTAS) for MIS. For example, we can use the PTAS proposed in [34] to approximate the MIS when the sensing radius is the same in the 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.

6.2 Implementation Issues

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

To balance the energy consumption, one simple method is generating a certain number of p-self-protection sets and rotating the active set among these sets. Notice that our proposed methods generate a 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 rankings. 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 lifetime 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 that can support it to be active for at most T_j seconds. Let $f(i, j)$ indicate whether sensor $v_j \in M^i$. Then, $f(i, j) = 1$ if $v_j \in M^i$; otherwise, $f(i, j) = 0$. Thus, the maximum lifetime scheduling is equivalent to solving the linear programming $\max \sum_{i=1}^{k} t_i$ with constrains $\sum_{i=1}^{k} t_i \cdot f(i, j) \le T_j$ 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 preset time. In other words, we can let the node with the most energy remaining have a higher priority to become an MIS (i.e., to be active) instead of using nodes with the highest rank (as in Algorithm 1) or highest ID (as in Algorithm 2), since the active nodes will consume more energy than those nonactive nodes. After a certain time, the network reruns 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 [35] and [36], where they consider the remaining energy or energy consumption rate as the criterion. On the other hand, considering that the energy cost (or drain rate) for being active at each node may be various, another variation of the minimum p-self-protection problem can be defined. Assume that each node u has a cost $c(u)$ to be active. Instead of minimizing the number of active nodes, the *minimum cost p-self-protection* problem minimizes the total cost of active nodes. The formal definition is given as follows:

Definition 3. ***Minimum cost p-self-protection*** *is a selected subset (denoted by $MCSP_p$) of V to be set as active sensors such that the sensor network is p-self-protected and the total cost of active nodes ($\sum_{u \in MCSP_p} c(u)$) is minimized.*

In [37], Wang et al. studied how to efficiently construct a low-cost MIS and MCDS for weighted sensor networks (i.e., approximation algorithms for finding an MIS or MCDS with minimized total cost). We can directly apply their method to select an MIS or MCDS in our algorithms. For more details, please refer to [37].

Notice that so far, we only consider a static sensor network where no sensor node moves and no sensor node is added or removed after the initial deployment. If the network allows node insertion and deletion, dynamic update procedures are needed for the proposed methods. Usually, there are two kinds of update methods: local update and global update. A local update will only affect the local neighborhood where the change occurs, while a global update basically reruns the construction algorithm in the whole network. There are also two ways to run the update method: on-demand update or periodical update. Most of the existing clustering algorithms using CDS or MIS are invoked periodically, while some algorithms (such as [36]) perform the updating only when it is required (i.e., on-demand). Our algorithm can adapt and combine both of these two update methods. If there is no major topology change (or no remarkable energy change for weighted methods), no global update will be performed until some preset timer expires. In other words, we perform our algorithm periodically with a preset time for dynamic networks. The time could be set quite long, depending on the frequency and type of network change. This kind of global update also insures the load balance throughout the network. However, for some major topology change (such as a large number of sensors died) or tremendous change of energy among the network (such as a big drop of energy level in many active nodes), an on-demand global update will be performed. Notice that not every topology change needs to trigger an update. It is clear that if a new node is already protected by p sensors or a nonactive node is deleted from the network, no update is needed. If a new node does not have enough protection or an active node that provides protections to other nodes is removed from the network, an update is needed. However, if the number of sensors added or removed is not large, we can just perform a local update in the neighborhood where the insertion or deletion happened. For example, if a new node has less than p active neighbors, it can simply select several nonactive neighbors to request the protection by sending the message ReqProtection. Notice that this kind of local update may hurt the approximation ratio of the method in the whole network. Therefore, it remains an open problem how to update the active set efficiently while preserving the approximation quality of the protection.

7 Simulations

In this section, we conduct extensive simulations on random networks to study the performances of our proposed algorithms. Since existing methods in [18] and [19] are only for 1-self-protection and do not have a constant approximation ratio, we do not implement their solutions for comparison here. In our experiments, we randomly generated a set V of n wireless sensors and the induced sensing graph $G(V)$ and then tested the connectivity and the minimum degree of $G(V)$. If it is connected and the minimum degree is larger than or equal to the desired self-protection level p, we construct our proposed distributed algorithm (in Section 3) 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 4 to construct the connected backbone among all active sensors and provide p-self-protection. Fig. 4 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 $500\,m \times 500\,m$ square, and the sensing range and transmission range are all set to $100\,m$. 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 is computed over all these 50 vertex sets. Notice that the parameter setting of our experiments here are just for demonstrations. We have tried other various settings, and the results and performances are stable; due to space limitations, we cannot present all of them here.

7.1 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 on different sizes of sensor networks (with 100 to 500 sensors) are plotted in Fig. 5. Fig. 5a shows the average number of active sensors generated by Algorithm 2. It is clear that a higher self-protection level p requires more active sensors. This is also illustrated in Figs. 4b, 4c, 4g, and 4h. However, for a 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. Figs. 5b and 5c show the number of messages used by Algorithm 2. Notice that even if 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 3.2.

We also test the effect of the sensing and transmission ranges on a fixed-size sensor network with 300 sensors. We set $p = 1, 2,$ and 3. The results are plotted in Fig. 6. Fig. 6a shows the average number of active sensors generated by

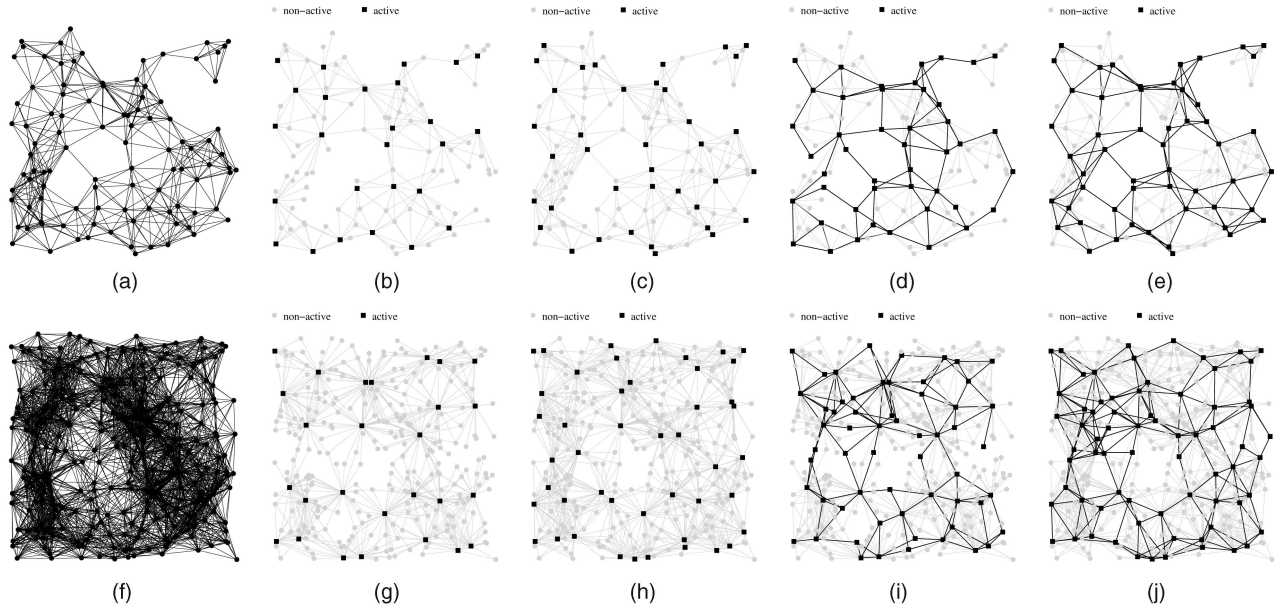


Fig. 4. Active sets generated by our self-protection algorithms for sensing graph G_1 with 100 sensors and sensing graph G_2 with 300 sensors. Here, black squares are active nodes, and gray dots are nonactive nodes. The black links in (d), (e), (i), and (j) are the links in the backbone keeping the active sensors connected. (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.

Algorithm 2. Again, a higher self-protection level p requires more active sensors. It is also interesting that the number of active sensors decreases while the range increases. In other words, for a fixed sensor network, a larger sensing range can reduce the number of active sensors needed for self-protection. It is reasonable, since an active sensor with a larger range can protect more sensors. If the range is infinite, the sensing graph becomes a completed graph, where only p active sensors is needed to achieve p-self-protection. Thus, there is a trade-off between the sensing range and the size of the active set. Figs. 6b and 6c show the number of messages used by Algorithm 2, which is stable for various ranges.

7.2 Self-Protection with Connectivity

In Section 4, we studied how to select the active sensors such that the network is p-self-protected and all active sensors form a connected backbone. Figs. 4d, 4e, 4i, and 4j 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 CDS (by selecting an MIS M_1 and finding connectors to connect three-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 three-hop-away MIS sensors who are not connected by other MIS sensors yet. Fig. 7 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 sensors active. Method 2 outperforms method 1 by activating fewer sensors. The reason is that many MIS sensors in M_1 are already connected by MIS sensors in later rounds since method 2 finds the connectors after p rounds of MIS. It is also clear in Fig. 7 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.

8 Related Work

WSNs have drawn a lot of attention recently due to their unique capability and wide spectrum. Many research activities on sensor networks are focused on how to balance the quality of protection [4], [5], [6], [7] (coverage) or fault tolerance [8], [9], [10], [11] or both [12], [13], [14], [15] with energy consumption of the sensors.

Sensor coverage is a key design issue in many sensor network applications. Cardei and Wu [16] and Dietrich and Dressler [17] provided complete surveys on the 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 [38]. Kumar et al. [6] studied the 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] and [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 the coverage problem under such measurements.

Fault tolerance is another key challenge in sensor networks. To make fault tolerance possible, the network topology must have k-connectivity or multiple paths between any two wireless devices. The authors of [8], [39], and [40] studied how to set the transmission radius to

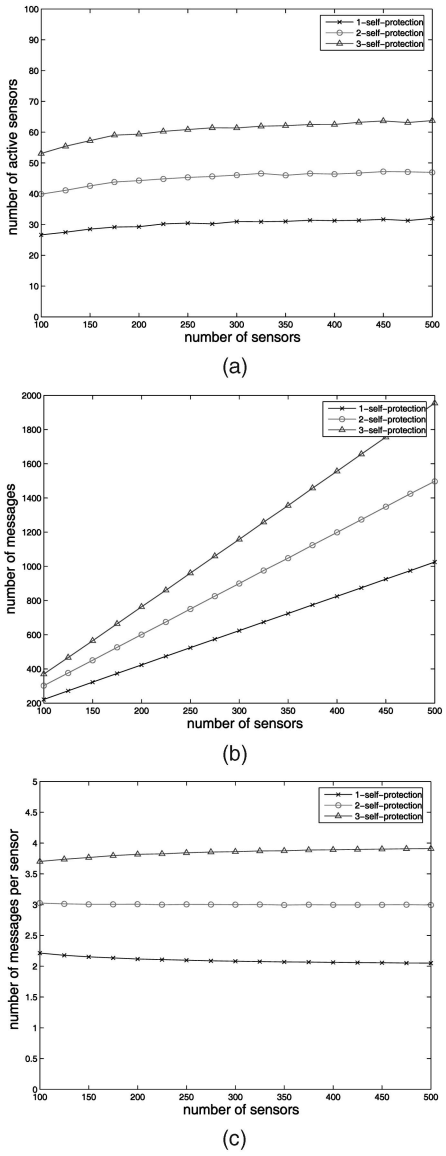


Fig. 5. Results for p-self-protection $(p = 1, 2, 3)$ when the number of sensors increases from 100 to 500. (a) Average number of active sensors. (b) Average number of messages. (c) Average number of messages per sensor.

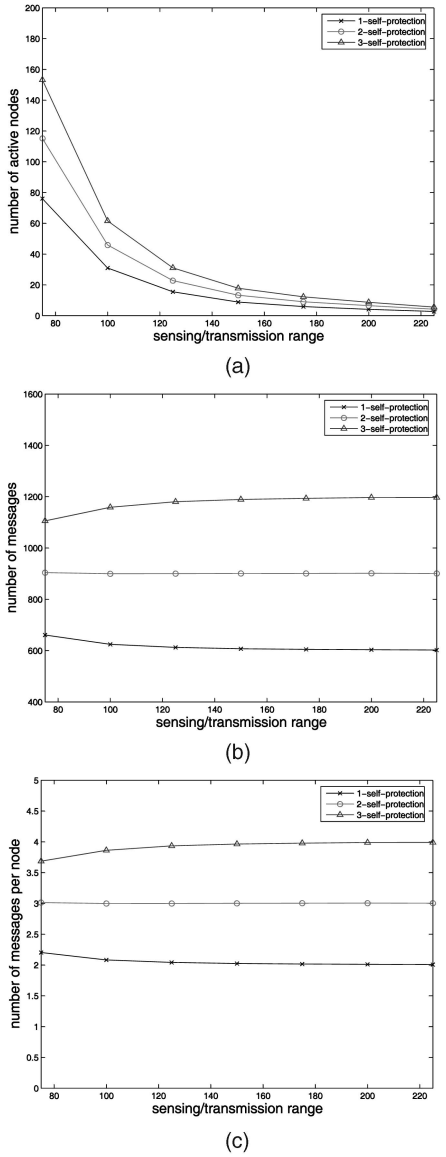


Fig. 6. Results for p-self-protection $(p = 1, 2, 3)$ when the sensing/transmission range increases from 75 to 225. (a) Average number of active sensors. (b) Average number of messages. (c) Average number of messages per sensor.

achieve the k-connectivity with certain probability for a random network, while the authors of [10] and [41] studied how to find a small transmission range for each node such that the resulting communication graph is k-connected. The authors of [11], [9], and [13] proposed localized algorithms to build k-connected topologies.

Until recently, coverage and connectivity problems have been studied together in sensor networks. Xing et al. [15] designed an integrated coverage configuration protocol to provide both certain degrees of coverage and connectivity guarantees. Zhang and Hou [12] proposed a decentralized density control algorithm to maintain sensing coverage and connectivity in high-density sensor networks. Both [15] and [12] 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. [14] has studied the optimal deployment pattern to achieve both 1-coverage of an area and 2-connectivity of the sensors. Zhou et al. [13] proposed a set of distributed algorithms to achieve a k-connected and k-covered sensor network by using a localized Voronoi graph and an extended relative neighborhood graph.

Notice that the p-self-protection problem studied here and in [18] and [19] is different from both k-coverage and k-connectivity problems. It focuses on providing p-protection to sensor nodes themselves.

9 Conclusion

A WSN is p-self-protected if at any moment, for any wireless sensor (active or nonactive), 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 when the sensing ranges of

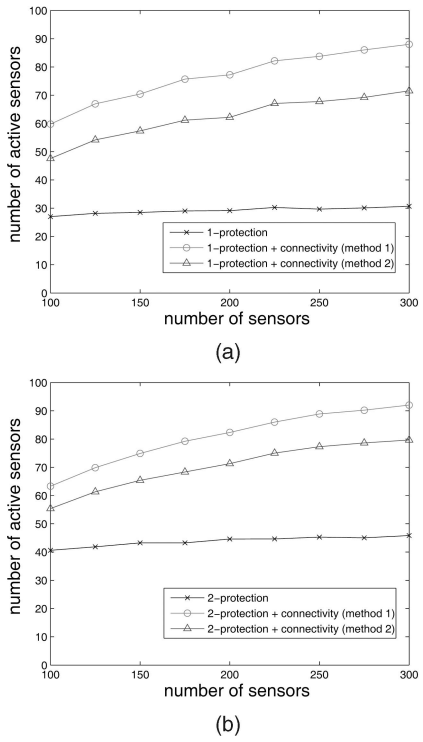


Fig. 7. Number of active sensors for p-self-protection with connectivity when number of sensors increases. (a) $p = 1$. (b) $p = 2$.

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. Our simulation confirms the performances of the proposed algorithms. We left the further study of the proposed methods in real testbeds as one of our future work.

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 a k-connected backbone and provides p-self-protection. Even though Theorem 5 gives a general framework to achieve this, there is still no simple approximation algorithm that can build a k-connected backbone efficiently. The second question is how to efficiently update the active set in mobile environments while preserving the approximation quality of the protection. The third question is that when the current sensors cannot provide p-self-protection, how do we add the smallest number of sensors such that the new network provides p-self-protection. The fourth 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.

Notice that in this paper, we did not specify what kind of sensing modality is used for providing the protection. In different sensing applications, the network may use different sensing devices (such as a camera, sonar, or radio) or combinations of devices to provide protection. The sensing range may also be different from the communication range (larger or smaller than it). Notice that even if we use radio as the sensing modality (i.e., the sensing range is the communication range), the connectivity problem is still different with the self-protection problem. A k-connected backbone will only provide k-protection for all active sensors, while k-self-protection also requires k-protection for all nonactive sensors.

Besides the sensor networking applications, we are also interested to find more potential applications for the proposed methods, such as usage in building a distributed intrusion detection system (IDS) or a dynamic trust model for large-scale ad hoc networks. In a distributed IDS, several IDS nodes need to be selected to protect and monitor parts of the network. Each IDS node may only be used for protection of nodes in a certain range (e.g., k-hop neighbors), and each node in the network needs p IDS nodes to protect. Therefore, our proposed algorithms can be used for the selection of the IDS nodes, which aims to minimize the number of IDS nodes while maintaining the p-protection for all nodes (including IDS nodes themselves). This problem is the same as the p-self-protection problem we studied above. Here, IDS nodes are just the "active" sensors, and other non-IDS nodes are the "nonactive" sensors.

ACKNOWLEDGMENTS

The work of Yu Wang was supported in part by the US National Science Foundation (NSF) under Grant CNS-0721666 and by funds provided by the University of North Carolina, Charlotte. The work of Xiang-Yang Li was partially supported by the National Basic Research Program of China (973 Program) under grant No. 2006CB30300, the National High Technology Research and Development Program of China (863 Program) under grant No. 2007AA01Z180, the RGC under Grant HKBU 2104/06E, and CERG under Grant PolyU-5232/07E. The work of Qian Zhang was supported in part by the National 863 Program of China under Grant 2006AA01Z228, by the Key Project of Guangzhou Municipal Government Guangdong/Hong Kong Critical Technology Grant 2006Z1-D6131, and by the HKUST Nansha Research Fund NRC06/07.EG01. The authors also greatly appreciate the anonymous reviewers and the editor for their constructive comments for improving the paper.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks,* vol. 38, no. 4, pp. 393-422, 2002.

[2] T. He, S. Krishnamurthy, J.A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-Efficient Surveillance System Using Wireless Sensor Networks," *Proc. Second ACM Int'l Conf. Mobile Systems, Applications, and Services (MobiSys),* 2004.

[3] C. Gui and P. Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks," *Proc. ACM MobiCom,* 2004.

[4] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage Problems in Wireless Ad-Hoc Sensor Network," *Proc. IEEE INFOCOM,* 2001.

[5] X.-Y. Li, P.-J. Wan, and O. Frieder, "Coverage in Wireless Ad-Hoc Sensor Networks," *Proc. IEEE Int'l Conf. Comm. (ICC),* 2002.

[6] S. Kumar, T.H. Lai, and A. Arora, "Barrier Coverage with Wireless Sensors," *Proc. ACM MobiCom,* 2005.

[7] S. Kumar, T.H. Lai, and J. Balogh, "On k-Coverage in a Mostly Sleeping Sensor Network," *Proc. ACM MobiCom,* 2004.

[8] X.-Y. Li, P.-J. Wan, Y. Wang, C.-W. Yi, and O. Frieder, "Robust Deployment and Fault Tolerant Topology Control for Wireless Ad Hoc Networks," *J. Wireless Comm. and Mobile Computing,* vol. 4, no. 1, pp. 109-125, 2004.

[9] M. Bahramgiri, M.T. Hajiaghayi, and V.S. Mirrokni, "Fault-Tolerant and 3-Dimensional Distributed Topology Control Algorithms in Wireless Multi-Hop Networks," *Proc. 11th IEEE Int'l Conf. Computer Comm. and Networks (ICCCN),* 2002.

[10] M. Hajiaghayi, N. Immorlica, and V.S. Mirrokni, "Power Optimization in Fault-Tolerant Topology Control Algorithms for Wireless Multi-Hop Networks," *Proc. ACM MobiCom,* 2003.

[11] N. Li and J.C. Hou, "FLSS: A Fault-Tolerant Topology Control Algorithm for Wireless Networks," *Proc. ACM MobiCom,* 2004.

[12] H. Zhang and J.C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Wireless Ad Hoc and Sensor Networks: An Int'l J.,* vol. 1, nos. 1-2, pp. 89-123, 2005.

[13] Z. Zhou, S. Das, and H. Gupta, "Fault Tolerant Connected Sensor Cover with Variable Sensing and Transmission," *Proc. Second Ann. IEEE Comm. Soc. Conf. Sensor, Mesh, and Ad Hoc Comm. and Networks (SECON),* 2005.

[14] X. Bai, S. Kuma, D. Xua, Z. Yun, and T.H. Lai, "Deploying Wireless Sensors to Achieve Both Coverage and Connectivity," *Proc. ACM MobiHoc,* 2006.

[15] 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. Sensor Networks,* vol. 1, no. 1, pp. 36-72, 2005.

[16] M. Cardei and J. Wu, "Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks," *Computer Comm. J.,* vol. 29, no. 4, pp. 413-420, 2006.

[17] I. Dietrich and F. Dressler, "On the Lifetime of Wireless Sensor Networks," Technical Report 04/06, Dept. of Computer Science, Univ. of Erlangen, Dec. 2006.

[18] D. Wang, Q. Zhang, and J. Liu, "Self-Protection for Wireless Sensor Networks," *Proc. 26th IEEE Int'l Conf. Distributed Computing Systems (ICDCS),* 2006.

[19] D. Wang, Q. Zhang, and J. Liu, "The Self-Protection Problem in Wireless Sensor Networks," *ACM Trans. Sensor Networks,* vol. 3, no. 4, p. 20, 2007.

[20] Y. Wang, X.-Y. Li, and Q. Zhang, "Efficient Self Protection Algorithms for Static Wireless Sensor Networks," *Proc. 50th IEEE Global Telecomm. Conf. (Globecom),* 2007.

[21] D.S. Johnson, "Approximation Algorithms for Combinatorial Problem," *J. Computer System Science,* vol. 9, pp. 256-278, 1974.

[22] V.V. Vazirani, *Approximation Algorithms.* Springer, 2001.

[23] S. Basagni, "Distributed Clustering for Ad Hoc Networks," *Proc. IEEE Int'l Symp. Parallel Architectures, Algorithms, and Networks (ISPAN),* 1999.

[24] B. Das and V. Bharghavan, "Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets," *Proc. IEEE Int'l Conf. Comm. (ICC),* 1997.

[25] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 13, no. 1, pp. 14-25, Jan. 2002.

[26] J. Wu and H. Li, "A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks," *Telecomm. Systems J.,* vol. 3, pp. 63-84, 2001.

[27] F. Dai and J. Wu, "An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 15, no. 10, pp. 902-920, Oct. 2004.

[28] Y. Li, T. Thai, F. Wang, C.-W. Yi, P. Wan, and D.-Z. Du, "On Greedy Construction of Connected Dominating Sets in Wireless Networks," *J. Wireless Comm. and Mobile Computing,* vol. 5, no. 88, pp. 927-932, 2005.

[29] K. Alzoubi, X.-Y. Li, Y. Wang, P.-J. Wan, and O. Frieder, "Geometric Spanners for Wireless Ad Hoc Networks," *IEEE Trans. Parallel and Distributed Processing,* vol. 14, no. 4, pp. 408-421, Apr. 2003.

[30] F. Dai and J. Wu, "On Constructing k-Connected k-Dominating Set in Wireless Networks," *Proc. 19th IEEE Int'l Parallel and Distributed Processing Symp. (IPDPS),* 2005.

[31] F. Wang, M.T. Thai, and D.-Z. Du, "2-Connected Virtual Backbone in Wireless Networks," *IEEE Trans. Wireless Comm.,* to appear.

[32] X.-Y. Li, W.-Z. Song, and Y. Wang, "Localized Topology Control for Heterogeneous Wireless Sensor Networks," *ACM Trans. Sensor Networks,* vol. 2, no. 1, pp. 129-153, 2006.

[33] V. Chvátal, "A Greedy Heuristic for the Set-Covering Problem," *Math. of Operations Research,* vol. 4, no. 3, pp. 233-235, 1979.

[34] H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, S.S. Ravi, D.J. Rosenkrantz, and R.E. Stearns, "NC-Approximation Schemes for NP- and PSPACE-Hard Problems for Geometric Graphs," *J. Algorithms,* vol. 26, no. 2, pp. 238-274, 1998.

[35] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. 33rd Hawaii Int'l Conf. System Sciences (HICSS),* 2000.

[36] M. Chatterjee, S.K. Das, and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *J. Cluster Computing,* vol. 5, no. 2, pp. 193-204, 2002.

[37] Y. Wang, W. Wang, and X.-Y. Li, "Efficient Distributed Low Cost Backbone Formation for Wireless Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 17, no. 7, pp. 681-693, Aug. 2006.

[38] S. Slijepcevic and M. Potkonjak, "Power Efficient Organization of Wireless Sensor Networks," *Proc. IEEE Int'l Conf. Comm. (ICC),* 2001.

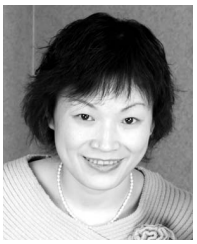
[39] M. Penrose, "On k-Connectivity for a Geometric Random Graph," *Random Structures and Algorithms,* vol. 15, no. 2, pp. 145-164, 1999.

[40] C. Bettstetter, "On the Minimum Node Degree and Connectivity of a Wireless Multihop Network," *Proc. ACM MobiHoc,* 2002.

[41] R. Ramanathan and R. Hain, "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment," *Proc. IEEE INFOCOM,* 2000.

Yu Wang received the BEng degree and the MEng degree in computer science from Tsinghua University, China, in 1998 and 2000, respectively, and the PhD degree in computer science from the Illinois Institute of Technology in 2004. He has been an assistant professor of computer science in the Department of Computer Science, University of North Carolina, Charlotte, since 2004. His current research interests include wireless networks, ad hoc and sensor networks, mobile computing, and algorithm design. He has published more than 60 papers in peer-reviewed journals and conference proceedings. He is an editorial board member of the *International Journal of Ad Hoc and Ubiquitous Computing* and an associate editor of the *International Journal of Mobile Communications, Networks, and Computing.* He has served as the program chair, the publicity chair, and a program committee member for several international conferences (such as IEEE IPCCC, IEEE GLOBECOM, IEEE ICC, IEEE INFOCOM, IEEE MASS, etc.). He is the program cochair of the First ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing (FOWANC 2008) and was the program cochair of the 26th IEEE International Performance Computing and Communications Conference (IEEE IPCCC 2007), the program cochair of the Fourth Workshop on Wireless Ad Hoc and Sensor Networks (WWASN 2007), and the program vice chair of the First International Conference on Integrated Internet Ad Hoc and Sensor Networks (InterSense 2006). He is a recipient of the Ralph E. Powe Junior Faculty Enhancement Awards from Oak Ridge Associated Universities. He is a member of the ACM, the IEEE, and the IEEE Communications Society.

Xiang-Yang Li received the BS degree in computer science and the BS degree in business management from Tsinghua University, Peoples Republic of China, both in 1995, and the MS and PhD degrees in computer science from the University of Illinois, Urbana-Champaign in 2000 and 2001, respectively. He has been an associate professor since 2006 and was an assistant professor from 2000 to 2006 in the Department of Computer Science, Illinois Institute of Technology. He has been a visiting professor with Microsoft Research Asia, Beijing, since May 2007. He also holds a visiting professorship or adjunct professorship at the following universities in China: TianJing University, WuHan University, and NanJing University. He was a member of the special class (of 20 students) in China prepared for the International Mathematics Olympics (IMO) from 1988 to 1990. His research interests span wireless ad hoc networks, game theory, computational geometry, and cryptography and network security. He has published about 80 conference papers in top-quality conferences such as ACM MobiCom, ACM MobiHoc, ACM SODA, ACM STOC, IEEE INFOCOM, etc. He has more than 40 journal papers published or accepted for publishing. He is an editor of *Ad Hoc & Sensor Wireless Networks: An International Journal.* He recently also co-organized a special issue of *ACM Mobile Networks and Applications* on noncooperative computing in wireless networks and a special issue of the *IEEE Journal of Selected Areas in Communications.* He served in various positions (such as the conference chair, local arrangement chair, financial chair, session chair, or TPC member) at a number of international conferences such as AAIM, IEEE INFOCOM, ACM MobiHoc, ACM STOC, and ACM MobiCom. He has also been invited to serve on the panel for the review of research proposals by several institutions such as the US National Science Foundation, the National Science Foundation of China, and RGC Hong Kong. He is a member of the ACM and the IEEE.



Qian Zhang received the BS, MS, and PhD degrees in computer science from Wuhan University, China, in 1994, 1996, and 1999, respectively. She joined the Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, in September 2005 as an associate professor. Before that, she was with Microsoft Research Asia, Beijing, from July 1999, where she was the research manager of the Wireless and Networking Group. She has published more than 150 refereed papers in international leading journals and key conferences in the areas of wireless/Internet multimedia networking, wireless communications and networking, and overlay networking. She is the inventor of about 30 pending patents. Her current research interests are in the areas of wireless communications, IP networking, multimedia, P2P overlay, and wireless security. She has also participated in many activities in the IETF Robust Header Compression (ROHC) WG group for TCP/IP header compression. She is an associate editor for the *IEEE Transactions on Wireless Communications*, *IEEE Transactions on Vehicular Technologies*, *IEEE Transactions on Multimedia*, *Computer Networks*, and *Computer Communications.* She has also served as a guest editor for the *IEEE Wireless Communications*, *IEEE Journal on Selected Areas in Communications*, *ACM/Springer Journal of Mobile Networks and Applications*, and *Computer Networks.* She received the TR 100 (*MIT Technology Review*) world's top young innovator award in 2004, the Best Asia Pacific (AP) Young Researcher Award elected by the IEEE Communication Society in 2004, the Best Paper Award by the Multimedia Technical Committee (MMTC) of the IEEE Communications Society, and Best Paper Awards at QShine 2006 and IEEE Globecom 2007. She received the Overseas Young Investigator Award from the National Natural Science Foundation of China (NSFC) in 2006. She is the vice chair and also the award committee chair of the Multimedia Communication Technical Committee of the IEEE Communications Society. She is also a member of the Visual Signal Processing and Communication Technical Committee and the Multimedia Systems and Application Technical Committee of the IEEE Circuits and Systems Society. She is a senior member of the IEEE.