Self-Protection for Wireless Sensor Networks

Dan Wang[1]*, Qian Zhang[2]†, Jiangchuan Liu[1]‡

[1] School of Computing Science,
Simon Fraser University,
Burnaby, BC, Canada, V5A 1S6,
Email: {danw, jcliu}@cs.sfu.ca

[2] Department of Computer Science,
Hong Kong Univ. of Sci and Tech
Clear Water Bay, Kowloon, Hong Kong,
Email: qianzh@cs.ust.hk

Abstract

Wireless sensor networks have recently been suggested for many surveillance applications such as object monitoring, path protection, or area coverage. Since the sensors themselves are important and critical objects in the network, a natural question is whether they need certain level of protection, so as to resist the attacks targeting on them directly. If this is necessary, then who should provide this protection, and how it can be achieved?

We refer to the above problem as self-protection, as we believe the sensors themselves are the best (and often the only) candidate to provide such protection. In this paper, we for the first time present a formal study on the self-protection problem in wireless sensor networks. We show that, if we simply focus on the quality of field or object covering, the sensors might not necessarily be self-protected, which in turn makes the system vulnerable. We then investigate different forms of self-protections, and show that the problems are generally NP-complete. We develop efficient approximation algorithms for centrally-controlled sensors. We then extend the algorithms to fully distributed implementation, and introduce a smart sleep-scheduling algorithm that minimize the energy consumption.

1 Introduction

A wireless sensor network consists of a large number of sensor nodes that perform sensing, computation, and communication. It has become an attractive modern tool for surveillance and protection applications, such as museum monitoring, military surveillance, object tracking, and intrusion detection. A key objective here is to provide enough coverage for the monitored entities; which range from individual objects to an entire area.

Obviously, the denser and more active the sensors are, the better the coverage quality we can expect, and hence, the better protection for the objects. Sensors, however, are small and uni-functional devices which are tightly constrained by non-rechargeable batteries. Sensors will die after the depletion of their energy resource and the quality of protection will thus be damaged. Many research activities on sensor networks are focusing on how to balance the quality of protection and energy consumption of the sensors.

Sensors may also die due to attacks. By sneakily dismantle a few sensors, the quality of coverage/protection can also be significantly affected. Fig. 1 shows an example of such attacks to the *maximum break path* model in a sensor network [10]. Intuitively, the maximum break path is a path traveling through the sensor network that has the least probability of being detected. More formally, define the weight of a path as the minimum distance from this path to any sensor in the network; a maximum break path is the maximum weight path from the source to the destination, as illustrated in Fig. 1 (a). Intuitively, when deploying the sensors, we should minimize the weight of the maximum break path. While elegant solutions have been devised in this context to provide quality coverage for the area, they generally assume that the sensors are not the target of attacks. This, however, creates a severe back door that can be explored by intruders. As shown in Fig 1 (b), if removing two sensors A, B in Fig 1 (a), the weight of the maximum break path can be substantially increased. Our simulations verifies that, by removing about 1% of the carefully chosen sensors, the weight of the maximum break path will increase 40%. Note that, the attack may not need to physically remove a sensor; a simple interference would work, and a smart intruder may strategically select weak sensors to amplify the effect.

Given the sensors themselves are important and critical objects in the network, we argue that they also need certain level of coverage and hence protection. We refer to the above problem as *self-protection*, as we believe the sen-

*Part of this work was done when Dan Wang was a visiting student at Microsoft Research, Asia.

†Q. Zhang's work was supported in part by DAG05/06.EG05 from Research Grant Council (RGC) of Hong Kong.

‡J. Liu's work was supported in part by a Canadian NSERC Discovery Grant 288325, an NSERC Research Tools and Instruments Grant, a Canada Foundation for Innovation (CFI) New Opportunities Grant, a BCKDF Matching Grant, and an SFU President's Research Grant.

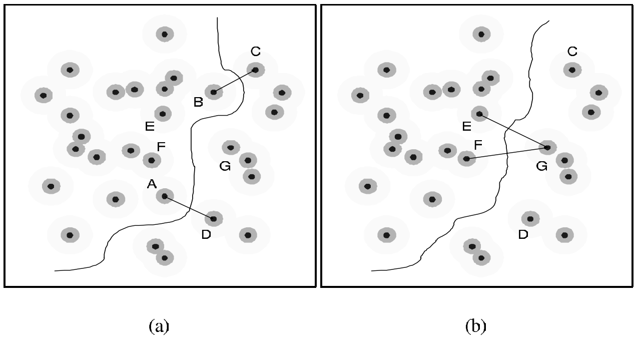

Figure 1. (a) The maximum break path in the network. The minimum weights are at (B, C) and (A, D). (b) By removing A and B, the weight of the maximum break path is significantly increased. The weight of the maximum path is remarked between lines of E, G and F, G.

sors themselves are the best (and often the only) candidate to provide protection. In a simple form, a sensor network is self-protected if all sensors are protected by at least one other active sensor. The challenges lie in three aspects: 1) We need to identify the necessary requirements for different levels of self-protections; 2) We need efficient and preferably distributed algorithms to accommodate the self-protection demands; and 3) Self-protection itself is never the ultimate objective in system design; whereas serving field/object protections is. We refer these as the *main objective(s)* of the system. An effective integration of self-protection with the protection of main objectives is needed.

In this paper, we for the first time present a formal study on the self-protection problem in wireless sensor networks. We show that, if we simply focus on enhancing the quality of field or object covering, the sensors might not necessarily be self-protected, which in turn makes the system vulnerable. We then investigate different levels of self-protections, and show that the problems are generally NP-complete. We develop efficient approximation algorithms for centrally-controlled sensors. For large sensor networks deployed in open areas, we present fully randomized and distributed implementations. Finally, we developed a two-tier architecture, which seamlessly integrates self-protection with the main objectives of the sensor network. Extensive simulations are conducted to illustrate the necessity of self-protection and the performance of our algorithms.

The reminder of this paper is organized as follows: We discuss the related work in Section 2. The self-protection problem is formally presented in Section 3. In Section 4, we consider a centralized scenario and discuss its complexity; we also show effective approximations. Section 5 extends the study to an distributed environment. In section 6, we describe a two-tier architecture to integrate the self-protection and the main protection objectives. Section 7 offer simulation results that verify the effectiveness and efficiency of our algorithms. We conclude our paper in Section 8.

2 Related Work

Wireless sensor networks have received a lot of attention recently due to its unique capabilities and the wide spectrum of applications. A general overview can be found in [1].

In many sensor network applications, providing desired field coverage or object protection is a key design objective. A typical coverage criterion is that every point of the field should be k-covered, which is studied in [13]. The k-coverage problem is further examined in [7], which proposes a sleeping/active schedule to minimize energy consumption. In [8], barrier coverage is considered, where the sensors can be used as barriers of, say, international borders. The problem is formulated as a k-multi-path problem and solved optimally if the sensors are centralized controlled. Distributed algorithms is also discussed in their work. Coverage of individual objects is studied in [2], which shows that the problem is NP-complete and heuristics are then developed. Other related works include variable-quality of coverage [4]. Besides, practical surveillance systems are also under active development; see for example [5].

A closely related and yet opposite research direction is to find break paths in the sensor protected area. A representative example is the *maximum break path* [10], as described in the introduction. The maximum break path is an indication of the quality that the area is protected. It is followed by *exposure paths* [14] that focuses on the paths with the least and most expected coverage.

Our work is motivated by these studies on quality coverage. However, to the best of our knowledge, the above studies do not address the possible weakness of the sensors themselves. Our self-protection does not conflict with these protection objectives; it can be viewed as a complementary new metric for the quality of coverage/protection. This metric is important because without protected sensors, quality coverage/protection for others can hardly be achieved.

3 Self-Protection: The Problem

We formulate the sensor network as a graph $G(V, E)$. V represents the set of sensor nodes, and E is the set of directed links, (u, v), where nodes $u, v \in V$ and v is in the sensing range of u. We use $|V|$ and $|E|$ to denote the number of nodes and the number of links, respectively. A sensor is called *active*, if it can carry out protections currently; otherwise it is called a *sleeping* sensor.

Definition 1 *A sensor network is k-self-protected if all sensors (active or sleeping) are covered by at least $k-1$ active sensors.*

In this paper, we focus on the 2-self-protection only; yet the techniques described can be extended to k-self-protection. In the rest, self-protection simply refers to the

2-self-protection, and we will point out the techniques for generalization whenever necessary.

4 Centralized Scenarios

We first consider the scenario where the sensors can be centralized controlled. This is often achievable in small-scale sensor networks. As energy consumption is a major concern in sensor networks, we use the following measures:

Definition 2 *A Minimum Self-Protection is a self-protection for the senor network, where the number of active nodes is minimized.*

We prove that both the minimum self-protection problem is NP-complete. We then present an approximation algorithm for the minimum protection. We have developed other measurements; for details, we refer to [15].

Theorem 1 *Finding minimum self-protection is NP-complete.*

Proof It is easy to see that the decision problem of validating a given self-protection is solvable in polynomial time. Therefore, the minimum self-protection is in NP class. To show this problem is NP-hard, we reduce the Minimum Set Cover to it; the former is known to be NP-complete [3].

Given a set cover instance (U, C); $U = u_1, u_2, \ldots, u_n$ is the universe of the elements and $C = c_1, c_2, \ldots, c_m$ is the family of the subsets of U, construct network $G = (V, E)$, where each node $v \in V$ corresponds to an element of U or an element of C. Thus we have $|V| = |U| + |C|$. E consists of two parts: 1) Make full connection of nodes representing the elements from C; 2) For each node $v \in V$, representing $u_i \in U$, $1 \leq i \leq n$, connect v with the node $w \in V$, representing $c_j \in C$, $1 \leq j \leq m$ where $u_i \in c_j$.

We next show that by finding a minimum self-protection, P, in G, we can find a minimum set cover for (U, C) in polynomial time. For each node v in P representing an element u_i, delete v and change it to w, which represents the subset c_j containing this single element u_i. The resulting protection is still a minimum self-protection with no isolating node, and this operation is polynomial. It is easy to see that the resulting nodes representing c_j are indeed a minimum set cover, because if there is another set cover with fewer sets, when mapping back to G, we can find a self-protection with fewer nodes, which contradicts to our assumption that P is a minimum self-protection. ■

The minimum self-protection problem can be formulated as a constrained dominating set problem, where the degree of each dominating node must be greater than one. The subgraph formed by the dominating nodes does not need to be connected, however; only isolating nodes are prohibited[1]. We then show that an approximation algorithm exists for minimum self-protection through minimum dominating set problem; the cost of the self-protection is the number of sensors selected to be active.

Lemma 2 *The cost of the minimum self-protection is at most twice of the cost of the minimum dominating set [3]. And this is also a lower bound.*

Proof A dominating set is a set of node where all remaining nodes in the network will connected to at least one node in the dominating set. It is easy to see that a minimum self-protection is a dominating set. We now prove, by contradiction, that the cost of this minimum dominating set is at least half of the cost of minimum self-protection.

If the minimum dominating set contains fewer nodes than half of the minimum self-protection, then we add the same number of nodes adjacent to the nodes in this minimum dominating set. The resulting set of nodes is clearly a minimum self-protection. This contradict to that the cost of the protection is minimum. This bound is also a lower bound since the minimum dominating set can be an independent set, e.g., the network is a straight line. ■

Theorem 3 *A $2(1 + log|V|)$ approximation algorithm exists for minimum self protection.*

Proof A $(1 + log|V|)$ approximation algorithm for minimum dominating set is given in [6]. Since the cost of minimum self protection will not be less than minimum dominating set problem, then by doubling this, we will have an easy $2(1 + log|V|)$ approximation algorithm. ■

The centralized algorithms are suitable for small-scale sensor networks, where all the sensors can be easily controlled through a central unit. For example, the video sensor monitoring systems in museums, where the number of art collections to be protected is very limited.

5 Distributed Scenarios

In a large sensor network, each sensor needs to make decisions based on limited information. In this section, we present two distributed approaches for self-protection, *Pre-scheduled Independent Activation* (PIA) and *Neighbor Co-operative Self-Protection* (NC). In PIA, an activation schedule is pre-defined and each sensor follows this schedule without knowing the behavior of other sensors. In NC, sensors negotiate activation schedules with each other in a distributed manner. In both PIA and NC, while maintaining qualified protection, sensors need to minimize and balance the energy consumption. We study the relationship between

[1] In this paper, we assume there is a mechanism that once an abnormal event is found, the sensor network will notify the responsible parties.

the quality of the self-protection with some key parameters of the system. Let R be the sensing range and l be the life time of a single sensor in full activation. We assume the sensors are uniformly distributed with density d.

5.1 Two Randomized Algorithms for PIA

In the centralized scenario, the sensor network can find a set of sensors so that all the sensors are protected. In the distributed scenario, this deterministic allocation can be difficult to achieve with no global information. We thus adopt the following probabilistic definition for self-protection:

Definition 3 *Given user defined confidence parameter $\delta \in (0, 1)$, a protection is said to be δ-self-protected if in any given area, the probability that the sensors in this area are not protected is less than δ.*

This definition is an extension for our self-protection in the probabilistic point of view. It can be extended to δ-k-self-protection where the probability that a sensor is not k-self-protected is less than δ.

In PIA, a timer and an activation probability p are built in each sensor. When the timer expires, the sensors activate itself with probability p and reset the timer. The key parameter a sensor needs to set for PIA is p, given the user required δ and the network setting. We now discuss two randomized algorithms: in the first, a sensor reactively links to another active sensor after activation; in the second, a sensor pro-actively decides its partner before activation.

Total Random Activation: Each sensor independently makes decisions to activate itself with probability p. After activation, the sensor will search within its sensing range of other active sensors, and connect them as partners. If there is no other active sensor in its neighborhood, the sensor goes back to sleep.

To determine the activation probability p, we assume the sensors are on a unit size mesh where the distance between each neighboring sensor is $\frac{1}{\sqrt{d}}$ and $R > \frac{1}{\sqrt{d}}$. In our simulation, we relax this assumption and show similar results hold for random uniform distribution.

Theorem 4 *The sensor network is δ-self-protected if $p > \frac{2\left(2+\ln\frac{1}{\delta}+\sqrt{\ln\frac{1}{\delta}(4+\ln\frac{1}{\delta})}\right)}{R^2 d}$.*

Proof Let Y_i be a random variable where $Y_i = 1$ if sensor i is activated and $Y_i = 0$ otherwise. Let S denote the set of sensors in a circle with radius $r = \frac{1}{2}R$. If any sensor that falls into this circle is active, all sensors in this circle is covered; see Fig. 2. Define Y where $Y = \sum_{i \in S} Y_i$. The total number of sensors in this circle is at least $n = \frac{1}{2}R^2 d - c$, where c is a constant, as approximated by the inner square of this circle; see Fig. 2. We omit c as it can be compensated by a small adjustment in the probability. Clearly, we have $E[Y] = \frac{1}{2}R^2 dp$. To construct a self-protection, at least 2 sensors need to be activated in this circle, i.e., we need to find $Pr[Y < 2]$. Since each sensor makes activation independently, using Chernoff's inequality [11], we have $Pr[Y < 2] = Pr[Y < \frac{2}{E[Y]}E[Y]] < e^{-(1-\frac{2}{E[Y]})^2 \frac{E[Y]}{2}} < \delta$. By solving the last inequality, we have $p > \frac{2\left(2+\ln\frac{1}{\delta}+\sqrt{\ln\frac{1}{\delta}(4+\ln\frac{1}{\delta})}\right)}{R^2 d}$. The theorem follows as the circle is arbitrarily chosen. ■

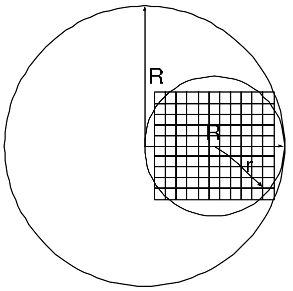

Figure 2. Outside circle has radius R; inside circle has radius $r = \frac{R}{2}$. Any sensor activated in the inside circle will protect all sensors within the this circle. The number of sensors is at least equal to the sensors in the inner box.

Since this bound is a lower bound, while provides guarantee for δ-protection, in practice it may activate more sensors than necessary. It, however, gives us important information of the relations between different parameters. Obviously, the activation probability p is inversely proportional to sensing range R^2 and density d, implying that the sensing range has a significant impact on p. The user confident level is easier to boost, as $p \propto \mathrm{O}(\ln\frac{1}{\delta})$. In fact, from probability theory, repeating the sampling $\mathrm{O}(\log k)$ times will improve δ to $\frac{\delta}{k}$. It is worth noting that, given a certain density d and sensing range R, it is possible that we can not achieve a certain level of self-protection at all; if the sensor network is too sparse. Therefore, to achieve a quality protection, we may have to sacrifice the cost of deploying more sensors. This is formally stated in Corollary 5.

Corollary 5 *To achieve the protection ratio δ, the minimum density of the network is $d > \frac{2\left(2+\ln\frac{1}{\delta}+\sqrt{\ln\frac{1}{\delta}(4+\ln\frac{1}{\delta})}\right)}{R^2}$.*

Proof Directly from $\frac{2\left(2+\ln\frac{1}{\delta}+\sqrt{\ln\frac{1}{\delta}(4+\ln\frac{1}{\delta})}\right)}{R^2 d} < p < 1$. ■

We are also interested in the life time of the system which can be estimated as $L = \frac{l}{p}$ if the active sensor sets are periodically alternated in a random fashion. Corollary 6 implies that the life time of the network is proportional to the sensor density. This is consistent with the experimental observations in [16].

Corollary 6 *To achieve the protection ratio δ and the expected life time L, the minimum density of the sensor network $d > \frac{2L\left(2+\ln\frac{1}{\delta}+\sqrt{\ln\frac{1}{\delta}(4+\ln\frac{1}{\delta})}\right)}{lR^2}$.*

Paired Random Activation: The sensors will first arbitrarily choose one of their neighbors to form pairs. Sensor pairs will activate themselves with certain probability, which, by re-using notations, is also denoted as p. We consider this scheme as opposed to Total Random Activation, where the sensors may find no other active sensors in its surroundings after activation. We have the following observation.

Theorem 7 *The sensor network is δ-self-protected if $p > \frac{2(1+\ln\frac{1}{\delta}+\sqrt{\ln\frac{1}{\delta}(1+\ln\frac{1}{\delta})})}{R^2 d}$.*

The proof technique is similar to Theorem 4. For space consideration, we omit the proof and refer to [15].

We compare Paired Random Activation and Total Random Activation by using the activation bounds in Theorem 4 and 7. Although these bounds are not tight, they give important indication of the performance of the two algorithms. We observe that, when δ is small, the number of sensors required to achieve δ-self-protection in Paired Random is much larger than Total Random. For such small δ, we need a more refined protection and high activation probability for both algorithms. Consequently, even if sensors make activation decision individually in Total Random, the probability that they can not find other active sensors in their neighborhood is relatively small if a large number of sensors are activated; Paired Random, however, might activate more sensor than needed in this case. On the other hand, when δ is large, the two algorithms perform closely. The drawback of Total Random is that some sensors might not find protection after activation in this case. While such analysis is based on a mesh sensor network, our simulations results in Section 7 validate the conclusions for uniformly distributed sensors.

5.2 Neighbor Cooperative Self-Protection

In PIA, to accurately estimate the activation probability p, the density of the sensor network should be known. This, however, can not be easily obtained if the sensors are deployed arbitrarily, e.g., from an aircraft. We now present another distributed self-protection, where sensors work cooperatively to provide necessary protections without knowing the density information.

There have been many studies on neighborhood cooperation for sensor networks, e.g., Geographical Adaptive Fidelity (GAF) [16] and Probing Environment and Adaptive Sleeping (PEAS) [17]. In these studies, sensors operate among different states. Our Neighborhood Cooperative (NC) self-protection is motivated by the above studies. The key difference is that, unlike these schemes where only one active sensor is needed, in our setting, sensors not only need to decide whether they should keep active, but also have to wait for other sensors if appropriate.

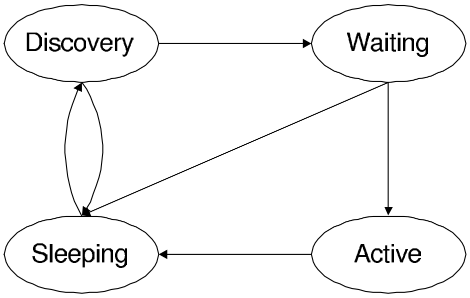

Figure 3. State transition diagram.

The NC algorithm has four states, namely, *active*, *discovery*, *waiting*, and *sleeping*; See Fig. 3 for the state transition diagram. In the sleeping state, the sensor is in the power saving mode for a period of *sleep_time_dur*. After the timer expires, the sensor changes to the discovery state and sends out probe messages to its neighbors. The active neighbor sensors will reply with rejections, which includes their remaining active time. If the sensor receives more than $rNum$ rejections, it will return to the sleeping state and set *sleep_time_dur* to the smallest remaining active time it receives. Otherwise, it will change to the waiting state, and periodically sends out probe messages. If it receives another probe message, the two sensors will form a pair and activate themselves. The pair sensors will stay active for duration of *work_time_dur*. The *work_time_dur* and *sleep_time_dur* are chosen uniformly from [0, MAX_WORK_TIME] and [0, MAX_SLEEP_TIME].

Fig. 4 gives an illustrative example of the state transition for a four-sensor network. In Fig. 4 (a), sensor 2 and 3 are in the active states, protecting each other and surrounding sleeping sensors. In Fig. 4 (b), two sleeping sensors 1 and 4 wake up and send discovery messages to their neighbors. In this example $rNum = 2$; so sensor 1 switches to the waiting state and sensor 4 returns to sleep after receiving two rejections. Sensor 4 sets the sleeping time to the remaining working time for the active sensors. In Fig. 4 (c), Sensor 0 changes to waiting state, and send probe messages to the neighbors. In Fig. 4 (d), the two waiting sensors 0 and 1 become active. The two former active sensors 2 and 3 go to sleep as their working timers expire, and the sleeping sensor 4 wake up to the discovery state.

The basic design philology of NC is to use $rNum$ to control the quality of the self-protection and the sleep/work schedules to balance the energy consumption. Its performance will be evaluated through simulations in Section 7.

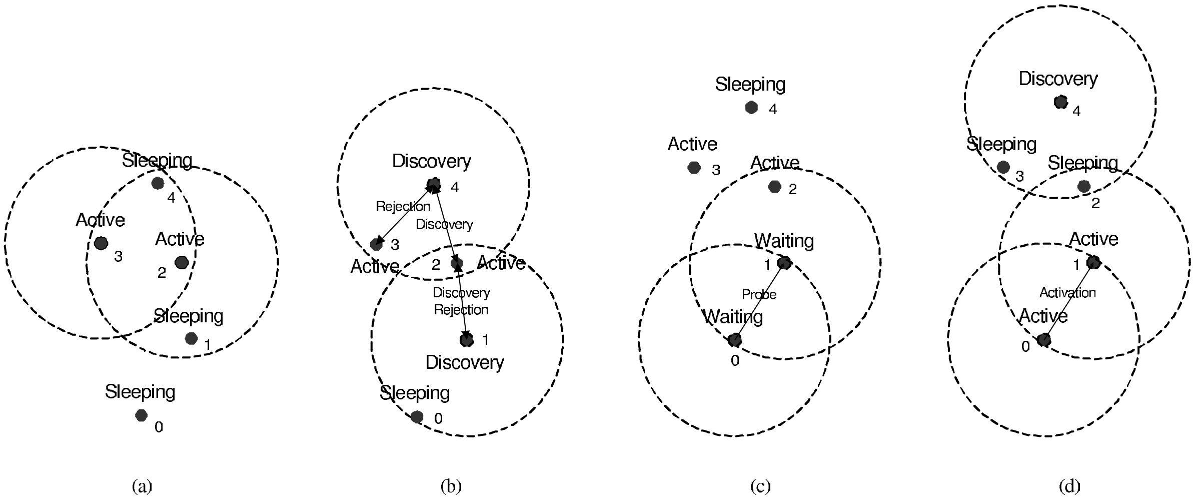

Figure 4. (a) A snapshot of the sensor network; (b) Sensor 1 and 4 are in discovery state; (c) Sensor 4 returns sleep after receiving two rejections ($rNum = 2$); Sensor 1 switch to waiting state; (d) Sensor 0 and 1 change to active state and sensor 4 wake up again.

6 Binding with the Main Objectives

As we mentioned earlier, self-protection improves the robustness of the network, but itself is never the single objective in the system design. It serves as a complement to such *main objective(s)* as monitoring the field or valuable objects. Thus, an effective integration of self-protection with these main protection objective is a critical issue.

We suggest a two-tier architecture, in which the sensor network will first calculate the set of sensors that can provide self-protection. All the sensors will then participate in the operations for the main objectives. The set of sensors to stay in active is the union of that for self-protection and for the main objectives. An interface is provided between the two tiers for communications about their respective energy consumption. Each tier then independently optimize their coverage and their energy consumption.

We adopt this architecture for its simplicity and adaptability to different coverage/protection scenarios (i.e., main objectives). To minimize energy consumption, we adaptively adjust the energy consumptions based on the information exchanged between the two tiers. We omit the details for simplicity and refer to [15] for further discussion.

7 Performance Evaluation

Since we have obtained bounded approximation algorithms for the centralized scenario and the algorithm is usually applied to small set of sensors; we thus focus on the evaluation of the distributed scenario.

In our simulation, unless otherwise specified, we uniformly deploy 500 sensors into a square field of 40m × 40m. The sensing range is set to 3m for each sensor. Each point in our figure represents an average of 50 random and independent experiments.

The Necessity of Self-Protection: As an additional level of protection, self protection has additional demands from the network, such as denser sensor deployment and more energy consumption. It is therefore important to justify its necessity. In the first set of experiment, we use the *maximum break path* [10], to show that the protection quality can be poor without self-protection.

The deployment of sensors should minimize the weight of the maximum break path. To attack this protection, we selectively dismantle a few sensors that are not protected by others. In our implementation, we try to maximize the profit. We randomly selects a subset of sensors and calculate the resultant maximum break path. This repeats $k = 8$ times and we remove the best subset of sensors, and compare the maximum break path before and after the removal.

We set the activation probability $p = 0.1$. In expectation, a total of 50 sensors will be active. We attack isolated sensors only and the attack ratio is set from 0% to 30%. From Fig. 5 we see that by attacking a few isolated sensors, the weight of the break path is increased substantially. For example, in our experiment, if 25% of the isolated sensors, (5.86 on average, i.e., 1.17% of the total nodes or 10% of the active nodes) are attacked, the maximum break path will increase by 39.4%. This degradation is remarkable; also note that a simple interference would achieve the same result of physical removal [12]. We thus conclude that self-protection is of great importance.

Performance of PIA: We study the effectiveness of the two pre-scheduled activation algorithms (Total Random and Paired Random) in Fig. 6. The activation probability for Total Random is from 5% to 40%; The activation probability for Paired Random is halved, so that the expected number of active sensors is the same as that in Total Random. We can see that Total Random performs worse than Paired Random

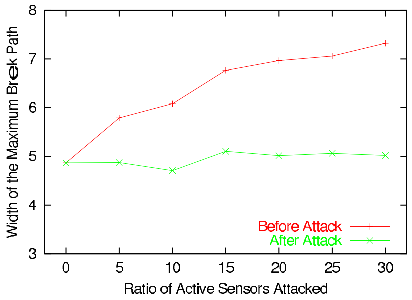

Figure 5. Width of the Maximum Break Path as a function of ratio of active sensor attacked.

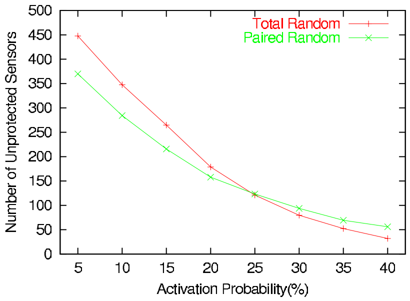

Figure 6. Number of unprotected sensors as a function of activation probability.

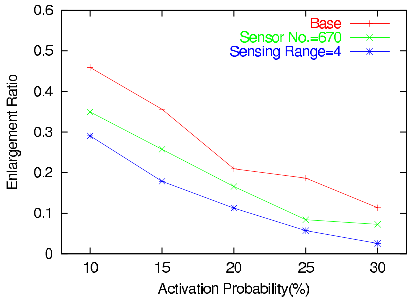

Figure 7. The enlargement ratio of the maximum break path as a function of activation probability.

if p is small, because, after activation, sensors might not find other active sensors in their neighborhood. When p is high, the number of sensors unprotected by Total Random is less than Paired Random. One reason is that in Paired Random, sensors need to find a pair sensor before activation; not all sensors, however, will succeed if all the neighbor sensors are paired by other sensors in advance. In addition, Total Random potentially enables better distribution of the active sensors, which will contribute to its protection capability.

We next evaluate the parameters that affect self-protection quality. In Fig. 7, we consider the change of the maximum break paths before and after an attack for Total Random. In particular, we are interested in the enlargement ratio of the weight of the maximum break path. The base line corresponds to the default network setting. Obviously, the higher the activation probability, the smaller the enlargement of the weight of the maximum break path. To understand the impact of the sensor density and sensing range, we also show the results corresponding to 500 sensors with sensing range of 4m, and 670 sensors with sensing range 3m, i.e., a respective increase of $\frac{1}{3}$ for the sensing range and density. We see that the sensor network with the default setting are most vulnerable to the attack and the enlargement is the highest. It follows our intuition that a denser deployment, or equivalently, larger sensing range, provides better protection. Note that the coverage of each sensor is a square function of the sensing range; hence, an increase in sensing range has a higher impact (less enlargement ratio). These observations are consistent with our analysis in Section 5.

Performance of Neighbor Cooperation: In neighborhood cooperation, we consider the energy consumption of sending, receiving, idling and sensing. We neglect the cost of sleeping, which is generally small in practice. We adopt the parameters in [9] as 20nAh (transmission) and 8nAh (receiving). The energy consumption for sensing is remarkably smaller than transmission, 20:1.447. We assume packet transmission has a rate of 6 per idling time. Therefore, according to [9] we set our costs, transmission : receiving : idling : sensing to 15:6:6:1. Similar energy consumption ratio is also observed in [16]. We assume that the sensors sensing the environment once per second. The total energy for a sensor is set to 80mAh and the duration of our experiment is 1500 seconds.

We compare the difference between PIA and NC with different parameters. For PIA, we set the interval time to 20 seconds, i.e., each sensor will randomly activate itself with probability p every 20 seconds. For NC, to have a similar effect, the ratio *work_time_dur : (sleep_time_dur + work_time_dur)* is set to p. In our setting, $p = 10\%$, *work_time_dur* = 10 seconds, and *sleep_time_dur* = 90 seconds. $rNum$ is set to 2 and 5, respectively. The initial protection ratio is set to be the same.

The comparison is shown in Fig. 8. In PIA, the protection quality is relative stable over time. In NC, the protection is improved over time. This is because in PIA, each time the network rebuilds the active sensor set, the protection capability remains unchanged. In NC, however, the sensors will know the next sleep time from other active sensors and awake at that point of time; This gradually increases the number of sensors needed for protection. Since the process stops according to the number of rejection messages, $rNum$ acts an indicator for the system to discourage (or encourage) future active sensors. Clearly, we can boost the protection capability of PIA by increasing p. As argued in our analysis before, this depends on the deployment of the sensor networks, which may not be easily controlled.

The corresponding energy consumption can be seen in Fig. 9. PIA is more energy effective, as it does not require extra message exchanges, a major energy consumption for sensor networks. In addition, the number of active nodes in use is less than NC as time progresses. We thus argue that PIA is beneficial if the sensor network is well planed so that we can pre-estimate the activation probability more accurately, and NC is better for dynamic environments.

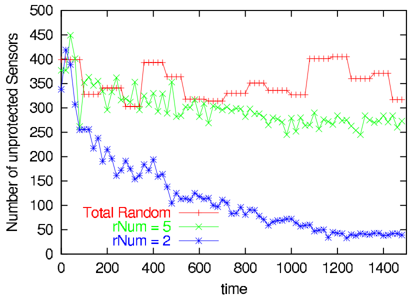

Figure 8. Number of unprotected nodes during time for Total Random, Neighborhood Cooperation with $rNum = 2$ and $rNum = 5$ respectively.

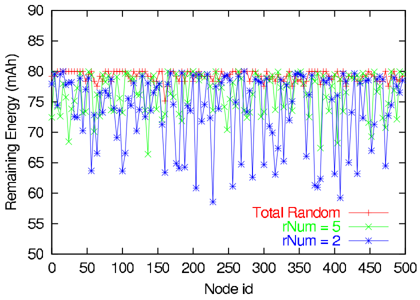

Figure 9. Energy consumption after 1500 seconds for Total Random, Neighborhood Cooperation with $rNum = 2$ and $rNum = 5$ respectively.

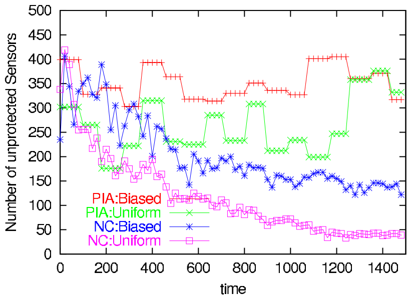

Figure 10. PIA, Total random self-protection for biased and uniform cases; Neighborhood Cooperative self-protection where $rNum = 5$ for biased and uniform cases.

We next consider the impact of sensor distributions. In Fig. 10, we introduce a biased distribution, where $\frac{1}{4}$ of the sensors are uniformly distributed in half of the area. In other words, one part of the network is denser and the other part is sparser. We see that NC performs worse in this biased distribution than in uniform distribution for there are fewer sensors in the sparse area. PIA, however, shows an opposite trend, suggesting that PIA benefit more from the biased network, in particular, the dense part.

8 Conclusion and Future Work

In this paper, we pointed out that the sensors themselves can be the weakness in a wireless sensor network for protection applications. We for the first time addressed and presented a formally study of the *self-protection* problems. We showed the complexity of the problem, and developed efficient algorithms under different networks situations.

In our study, we consider the main protection objectives as a black box. We however conjecture that some main protection objects might be overlapping with and thus assist self-protection while others might introduce conflicts. Thus, an interesting future work is to joint optimize the self-protection and other protection objectives.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks", *IEEE Communications Magazine*, vol. 40, no. 8, pp.102-114, Aug. 2002.

[2] M. Cardei, M. Thai, Y. Li, and W. Wu, "Energy-Efficient Target Coverage in Wireless Sensor Networks", in *Proc. IEEE INFOCOM'05*, Miami, FL, Mar. 2005.

[3] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, San Francisco, 1979.

[4] C. Gui and P. Mohapatra. "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks", in *Proc. ACM MOBICOM'04*, Philadelphia, PA, Sept. 2004.

[5] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-Efficient Surveillance System Using Wireless Sensor Networks", in *Proc. ACM MOBISYS'04*, Boston, MA, June. 2004.

[6] D. Johnson, "Approximation Algorithms for Combinatorial Problems", *Journal on Computing System Science*, no. 9, pp 256-278, 1974.

[7] S. Kumar, T. Lai, and J. Balogh, "On k-Coverage in a mostly Sleeping Sensor Network", in *Proc. ACM MOBICOM'04*, Philadelphia, PA, Sept. 2004.

[8] S. Kumar, T. Lai, and A. Arora, "Barrier Coverage with Wireless Sensors", in *Proc. ACM MOBICOM'05*, Cologne, Germany, Aug. 2005.

[9] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", in *Proc. ACM WSNA'02*, Atlanta, GA. Sept. 2002.

[10] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks", in *Proc. IEEE INFOCOM'01*, Anchorage, AK, Apr. 2001.

[11] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, New York, NY, 1995.

[12] E. C.-H. Ngai, J. Liu, and M. R. Lyu, "On the Intruder Detection for Sinkhole Attack in Wireless Sensor Networks", to appear in *IEEE ICC'06*, Istanbul, Turkey, June. 2006.

[13] S. Slijepcevic and M. Potkonjak, "Power Efficient Organization of Wireless Sensor Networks", in *Proc. IEEE ICC'01*, Helsinki, Finland, Jun. 2001.

[14] G. Veltri, Q. Huang, G. Qu, and M. Potkonjak, "Minimal and Maximal Exposure Path Algorithms for Wireless Embedded Sensor Networks", in *Proc. ACM SENSYS'03*, Los Angeles, CA, Nov. 2003.

[15] D. Wang, Q. Zhang, and J. Liu "Self-Protections for Wireless Sensor Networks", Technical Report, Simon Fraser University, March, 2006.

[16] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad-Hoc Routing", in *Proc. ACM MOBICOM'01*, Rome, Italy, July, 2001.

[17] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks", in *Proc. IEEE ICDCS'03*, Providence, RI, May, 2003.