A Distributed Constant Factor Self-Protection Algorithm for Wireless Sensor Networks

Kamrul Islam
School of Computing
Queen's University
Kingston, Canada
Email: islam@cs.queensu.ca

Selim G. Akl
School of Computing
Queen's University
Kingston, Canada
Email: akl@cs.queensu.ca

Abstract

Wireless sensor networks have drawn a lot of attention owing to their potential use in different surveillance applications including object monitoring, path protection, area coverage, and so on. Since sensors provide monitoring to other objects, it is often necessary to give them a level of protection such that sensors can take certain actions when attacks are targeted on them. One natural idea is to monitor sensors by their neighbors such that the neighbors can inform the base station when monitoring sensors are in danger. Keeping this in mind, the p-self-protection subset problem has been formally introduced in [4] to find a subset of sensors such that for any sensor in the network there are at least p active sensors from the subset that monitor it. The authors in [4] first show that finding minimum 1-self-protection subset is NP-complete and then give an $O(\log n)$-approximation algorithm. Later, with a view to obtaining a constant factor approximation ratio, centralized and distributed algorithms to solve the general p-self-protection subset problem are presented in [3]. In this paper, we present a counterexample showing that the algorithm in [3] fails to find a p-self-protection subset in the network. Later, we present a distributed algorithm for the p-self-protection subset problem and prove that the algorithm achieves a constant factor approximation ratio. To the best of our knowledge, this is the first constant factor approximation ratio to the p-self-protection subset problem. Our distributed algorithm runs in $O(\log^* n)$ time where n is the total number of sensors in the network.

Keywords: Wireless Sensor Network, Self-Protection, Distributed Algorithm, Maximal Independent Set.

1. Introduction

Once deployed with a limited amount of energy, nodes in sensor networks continue to spend energy by sensing, gathering, and distributing data from the environment. Among myriad of applications, sensor networks are used for protection and surveillance of many different applications such as museum monitoring, military surveillance, object tracking, and so on. More specifically, nodes carry out protection and monitoring activities by collecting and sending timely signals to a base station to take appropriate measures in case some unusual events occur to the objects they monitor.

Although sensors are mainly used for protection of objects as mentioned, there are situations when it is vital to monitor the sensors as well. The issue of monitoring sensors arises because battery-powered sensors are vulnerable to damage, failure and attacks. If sensors themselves are incapable of performing monitoring, then how could we protect our objects? It is therefore important to ensure that sensors are active and functioning as desired. One of the ways this can be done is to require neighboring sensors to monitor the conditions of each other.

Imagine a scenario where among the set of sensors there is a single sensor which is monitoring a certain object. If, for some reason, this sensor malfunctions or runs out of energy then the object will be left unprotected. However, if there were some other neighboring sensors which could monitor this particular sensor then any status of that sensor (whether running out of energy or in danger) could be relayed by them to the base station so that proper action could be taken, such as, for example, replacing the sensor by other sensors. This scenario raises the issue that sensors themselves require a certain level of protection. The problem of protecting the sensors by other sensors is called *self-protection*. In its simplest form, a sensor network is self-protected if all sensors are protected by at least one active sensor (other than itself), where active sensors are sensors which monitor their neighbors by being active. In general, a wireless sensor network is p-self-protected if at any moment for any sensor there are at least p active sensors (other than itself) that can monitor it. This problem has been formally introduced and investigated by Liu et al [4].

The rest of the paper is organized as follows. In Section 2 we describe related work and in Section 3 we provide definitions and assumptions that are used throughout the paper. A distributed algorithm (whereby the nodes in the network execute the algorithm individually) is presented in

Section 4. We provide a theoretical analysis of our algorithm in Section 5 followed by conclusions in Section 6.

2. Related Work

Since sensors are power-limited and monitoring requires sensors to be active all the time thereby consuming valuable energy, finding a small number of sensors to do the monitoring has been one of the main objectives of research in this regard. This small subset of active sensors are used to provide protection or coverage for every location of the area, in which they are deployed, and at the same time other sensors can be put in energy-efficient sleeping mode to conserve energy. A number of coverage problems, including protection of certain static objects with a required degree of fault-tolerance, and methods to solve them have been surveyed by Wu et. al [5], where the main focus is on the area coverage problem. A generalization of the area coverage problem called k-coverage, i.e., whether every point is covered by at least k sensors, is studied by Kumar et el [10]. The k-coverage problem is directly related to the fault-tolerance issue in sensor networks. This is because k-coverage basically provides fault-tolerance in the network since failure of at most $k' < k$ monitoring nodes to monitor a particular area will not leave that area uncovered. Gui et al [7] propose algorithms for area coverage problems considering fault-tolerance, energy-consumption and quality of coverage.

A shift from the coverage problem to the problem of protecting the sensors themselves was first made by Liu et al [4]. In [4] they prove that finding minimum 1-self-protection subset (that is, finding a subset of nodes of minimum cardinality such that each node in the network is protected by at least one other node) is NP-complete through a reduction from the set-cover problem [11]. As a consequence they give a centralized algorithm for this problem and achieve a $2(1+\log n)$ approximation ratio, where n is the total number of sensors. That is, their algorithm finds a 1-self-protection subset which is at most O($\log n$) times the cardinality of the minimum 1-self-protection subset. However, they do not give any algorithm for the general p-self-protection subset problem. The authors also show that focusing only on the quality of area or object covering does not necessarily make the sensors to be self-protected. Subsequently, a probabilistic definition of the self-protection subset (a protection is called δ-p-self-protection if the probability that a sensor is not p-self-protected is less than δ), as well as two randomized algorithms to solve the problem in a distributed fashion, are given in [4].

Later, the problem of finding minimum p-self-protection subset was thoroughly studied along with some variants such as requiring the protecting sensors be also connected, allowing sensors with varying sensing ranges, and so on, by Wang et. al [3]. In their paper, the authors claim that they have two constant factor approximation algorithms (in both centralized and distributed settings) for this problem. Unfortunately, their claim is in error. We provide a counterexample to show an instance of a graph where the p-self-protection algorithm does not form a subset that p-protects the nodes in G. They discuss other variants of the problem, namely, how p-self-protection can be achieved when the sensing radii of the sensors are not the same, or generating a certain number of p-self-protection subsets and activating them one after the other.

In this paper, we investigate the p-self-protection problem. By presenting a counterexample of the work of Wang et. al [3], we provide a distributed algorithm to solve the p-self-protection problem. To the best of our knowledge this is the first distributed constant factor approximation algorithm for the p-self-protection problem. This result is significant in two ways: first it achieves a constant factor which is better than the previous (logarithmic factor) results and second it is distributed.

3. Model and Definitions

We assume that sensors are deployed in the plane and model the network by an undirected unit disk graph $G = (V, E)$ where the vertex set V denotes the set of sensors and E represents the links $(u, v) \in E$ between two sensor nodes $u, v \in V$ if they are within their transmission range. We assume that every sensor node $u \in V$ has the same transmission range which is normalized to one. Define the neighbor sets $N(u)$ and $N[u]$ of sensor node u as $N(u) = \{v|(u, v) \in E, u \neq v\}$ and $N[u] = N(u) \cup \{u\}$. By $N_k(u)$ we mean the set of nodes from u which are at most k hops away from u. For simplicity we use $N(u) = N_1(u)$. The degree of a node is the number of neighbors it has, that is, $|N(u)|$. Each node is identified by a unique index.

A set $I \subseteq V$ is called an independent set if no two nodes $u, v \in I$ are neighbors. A set $I \subseteq V$ is called an maximal independent set (MIS) if no independent set $T \supset I$ exists.

3.1. Minimum p-self-protection Subset

A sensor network is p-self-protected if, for any sensor, there are at least p neighboring active sensors (a sensor is not a neighbor to itself) that monitor it. The minimum p-self-protection subset [3] is a subset P of V to be set as active sensors such that the network is p-self-protected and the cardinality of P is minimized. See Figure 1 where the set of black nodes form an 1-self-protection (Figure 1 (a)) and a 2-self-protection (Figure 1 (b)) subsets. As mentioned before, it is proved in [4] that computing a minimum 1-self-protection subset is NP-complete, which implies that the general minimum p-self-protection subset is also NP-complete. From now on instead of using 'p-self-protection' we use 'p-protection'. We provide a distributed constant factor approximation algorithm for the p-protection subset problem when $p \in \{1, 2, 3\}$, which, we think, are likely the more useful values of the protection parameter. This is because in real life if we allow more sensors $p > 3$ for the protection then we waste a lot of energy from the monitoring sensors which are doing nothing but providing protection to other sensors. Let OPT_1, OPT_2 and OPT_3 denote the minimum 1-, 2- and 3-protection subsets, respectively.

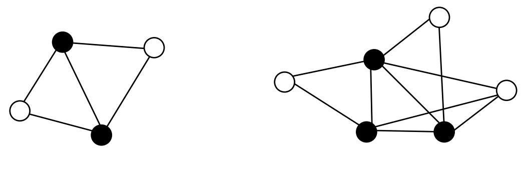

Figure 1. Minimum 1-self-protection and 2-self-protection subsets.

We assume a synchronized message passing system as in [13], [14] in which time is divided into equal rounds. In each communication round a node of the network graph is capable of receiving messages sent in the previous round, performing local computations, and broadcasting a message to its neighbors. The time complexity of an algorithm is the number of rounds it needs to produce the desired results. It is well known that a synchronous message passing algorithm can be turned into an asynchronous algorithm with the same time complexity, but with a larger message complexity. Sensors have their own ids and connectivity information and obtain their neighbors' indices and degrees, through exchanging *"Hello"* messages.

4. Minimum p-protection Subset Problem

In this section, we present a counterexample to show an instance where the p-protection algorithm fails to find a p-protection subset of nodes [3].

4.1. Wang et. al [3] Algorithm: p-protection Subset Problem

The algorithm in [3] for the general minimum p-protection subset problem is now described. In the first iteration, the algorithm [3] first finds an MIS M_1 based on the ranks (each node u is assigned a unique integer from $[1, n]$ called its rank, $r(u)$) of the nodes where a node in selected to be in M_1 if it has the largest rank among all its neighbors. Each node not selected in M_1 is assigned a unique rank $r(v) = r(v) + k * n$, where initially k is assigned 1 and incremented by one at each iteration, and the ranks of the nodes belonging to the MIS are not changed. Then in the next iteration, the algorithm generates a new MIS M_2 from the nodes not selected in M_1 based on their ranks. Thereafter, $k = k + 1$ and ranks are updated by the equation $r(v) = r(v) + k * n$ for those nodes which did not belong to M_1 and M_2. This procedure of generating new MISs is repeated p times, and it is claimed that this algorithm finds p MISs. Finally, as this algorithm continues, if a node $u \in M_i$, $1 \leq i \leq p$ has fewer than p neighboring nodes in $\bigcup_{i=1}^{p} M_i$, then u selects a neighbor v to protect u to such that u is p-protected. According to [3], the union of all M_i's forms the p-protection subset.

In order to prove this claim (Theorem 2 in [3]), it is stated that for any node $u \in \bigcup_{i=1}^{p} M_i$, u has at least $p-1$ protectors from $\bigcup_{i=1}^{p} M_i$ since it has been protected by MIS nodes in every iteration except the iteration in which it is selected. Then if u has only $p - 1$ neighboring nodes in $\bigcup_{i=1}^{p} M_i$, it just adds an arbitrary neighbor v to protect itself.

However, this statement is not correct. Figure 2 shows a counterexample for the p-protection subset problem, where $p = 4$. Assume that the algorithm finds the first MIS M_1 consisting of nodes shown with label 1. According to the algorithm, another MIS M_2 is found consisting of the nodes that do not belong to M_1. However, in the example, there is no way to form a second MIS M_2 from the remaining nodes because any attempt to construct the second MIS without using nodes from the previous MIS will fail. More specifically, if we want to form the second, the third or the fourth MISs M_2, M_3 or M_4, each will consist of exactly two nodes (shown with label 2, label 3, and label 4, respectively) and none of which is an MIS of the whole graph. This means that in any case, either the middle label 1 node (when we form M_2 and M_4) or the label 1 nodes on the left and on the right (for M_3) are not protected by any sensors. Again, according to [3], for all nodes $u \in \bigcup_{i=1}^{p} M_i$, u has at least $p - 1$ protectors from $\bigcup_{i=1}^{p} M_i$, given that u has already been protected by MIS nodes in every iteration except the iteration in which it is selected as a node in MIS. However, in the example of Figure 2, the middle label 1 node is not protected by $p - 1 = 3$ nodes.

Finally, this erroneous observation leads them to falsely claim that this can achieve a 10-approximation algorithm for the general p-protection subset problem. The authours of the paper [3] agreed that the counterexample we provide is correct [6].

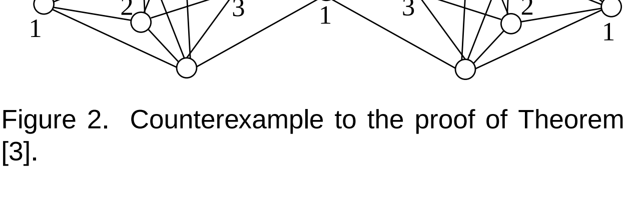

Figure 2. Counterexample to the proof of Theorem 2 [3].

4.2. Solution: 1-protection Subset Problem

For the minimum 1-protection subset problem we present a distributed version of the centralized algorithm in [3] that gives a constant approximation ratio. First, in order to prove a correct constant factor ratio we use a well-known property of MIS in unit disk graphs. Specifically we use the following fact from [9].

Theorem 4.1: [9] Let I be any MIS of the unit disk graph G and u be an arbitrary node in I. The number of nodes in I that are exactly two hops away from u are at most 23.

Proof: The proof follows from the standard area argument. The disks of radius 0.5 centered at the nodes in I that are exactly two hops away from u all lie within the annulus centered at u of radii 0.5 and 2.5 ared disjoint. Therefore, the number of nodes in I that are exactly two hops away from u is less than $\frac{\pi.2.5^2-\pi.0.5^2}{\pi.0.5^2} = 24$.

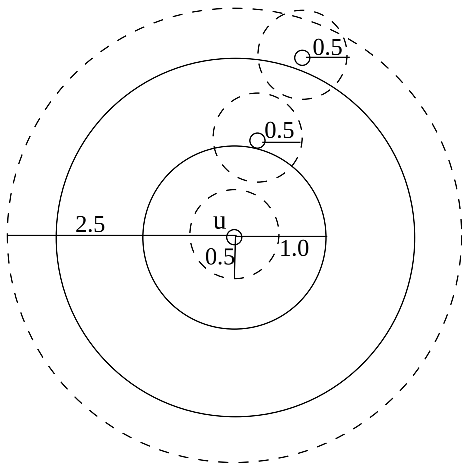

Figure 3. The disks of radius 0.5 centered at the nodes in I that are within three hops away from u all lie within the annulus centered at u of radii 0.5 and 2.5 and are disjoint [9].

□

4.2.1. Distributed Algorithm. First, nodes distributedly construct an MIS (following the algorithm in [8]) and then each member of the MIS arbitrarily selects one neighbor to protect it. All nodes in the MIS as well as the selected neighbors are set as active. This forms a 1-protected set since each node not belonging to the MIS is protected by at least a member of the MIS and every node in the MIS is protected by its selected neighbor. The set of all active nodes forms the 1-protection subset A_1. The constant approximation ratio is obtained as follows.

Theorem 4.2: For the minimum 1-protection subset problem, the distributed algorithm finds a valid 1-protection subset A_1 and has size at most 33 times the minimum 1-protection subset OPT_1, that is, $|A_1| \leq 33 * |OPT_1|$.

Proof: Consider any node $u \in V$. There are two cases, either $u \in MIS$ or $u \notin MIS$:

Case 1: If $u \in MIS$ then there is no MIS node in its neighborhood but it can have at most 23 MIS nodes in (Theorem 4.1) its two-hop neighborhood. If each of these 23 nodes selects a unique node from $N(u)$ and u selects its active neighbor then there will be at most 24 active nodes in $N(u)$.

Case 2: If $u \notin MIS$ then it can have at most five MIS nodes in $N(u)$. Each of the five MIS nodes can arbitrarily select a unique node $N(u)$ and at the same time each of 23 MIS nodes in the two-hop neighborhood can select a node in $N(u)$. The MIS nodes and their selected neighbors are set active. Thus we have 33 active nodes in $N(u)$. In the optimal 1-protection subset a node u must have at least one active node in its neighborhood. Thus the approximation factor is 33. Thus $|A_1| \leq 33 * |OPT_1|$. □

4.3. Distributed Algorithm $\mathcal{D}$: 2 and 3-protection Subset Problems

In this section, we present a distributed algorithm for the p-protection subset problem where $p \in \{2, 3\}$. The algorithm constructs protection subsets A_2 and A_3 for the 2 and 3-protection subset problems.

4.3.1. 2-protection:. We first run the distributed MIS [8] algorithm on G and let D_1 be the set of nodes returned by the algorithm. Nodes in D_1 are set as active. Each node $u \in D_1$ maintains a subset $M_1(u) \subseteq N(u)$ which is initialized to null. Node u adds at most five nodes from its neighbors to $M_1(u)$ to give the required protection to the nodes in $N(u) \setminus M_1(u)$. In the following we elaborate on how neighbors are inserted in $M_1(u)$.

Beginning with $M_1(u) = \phi$, at first, u selects an arbitrary neighbor node $v \in N(u)$ and adds it to $M_1(u)$. With $M_1(u) = \{v\}$, node u checks whether all nodes in $N[u] \setminus M_1(u)$ are covered by $M_1(u)$. Otherwise, u selects the closest (in terms of Euclidean distance) node $v_{i'} \in N(u)$ such that $vv_{i'} \notin E$ (ties are broken arbitrarily). $v_{i'}$ is added to $M_1(u)$ (i.e., $M_1(u) = M_1(u) \cup \{v_{i'}\}$).

Now u checks whether $M_1(u)$ covers the nodes in $N[u] \setminus M_1(u)$, if not, then u continues adding nodes $v_{i''} \in N(u)$ to $M_1(u)$ such that $v_m v_{i''} \notin E$, $v_m \in M_1(u)$ and $v_{i''}$ is the closest (in terms of Euclidean distance) to some node $v_m \in M_1(u)$. $v_{i''}$ is added to $M_1(u)$ and u checks whether $M_1(u)$ covers $N[u] \setminus M_1(u)$. If not, then the process is repeated. Due to the property of unit disk graph we know that $M_1(u)$ can contain at most 5 such nodes (independent nodes), in which case all nodes in $N[u] \setminus M_1(u)$ are covered. Notice that as u forms $M_1(u)$, u is already protected by $M_1(u)$. If

$|M_1(u)| = 1$ then u arbitrarily selects a node $v \notin M_1(u)$ to ensure its required 2-protection.

All nodes in $\bigcup_{u \in D_1} M_1(u)$ are set as active. Notice that as $M_1(u)$'s are formed following the above procedure, all nodes $V \setminus (D_1 \cup (\bigcup_{u \in D_1} M_1(u)))$ are at least 2-protected. This is because a node $v \in V \setminus (D_1 \cup (\bigcup_{u \in D_1} M_1(u)))$ belongs to $N(v') \setminus M_1(v')$ for some $v' \in D_1$ and since v' computes $M_1(v')$, v must be covered by v' and $M_1(v')$. Hence v is 2-protected. Each node in $\bigcup_{u \in D_1} M_1(u)$ is guaranteed to be at least 1-protected since these nodes are protected by D_1. In order to achieve 2-protection for these nodes, each node $u' \in \bigcup_{u \in D_1} M_1(u)$ which is not 2-protected arbitrarily selects a neighbor to protect itself. These selected neighbors are set as active. Let A_2 denote the set of all active nodes in G. Then the set A_2 protects the nodes in G.

4.3.2. 3**-protection:.** Given G, we first run the distributed MIS [8] algorithm and obtain D_1 as before, and the nodes in D_1 are set active. Following the procedure of constructing $M_1(u)$, each node $u \in D_1$ now constructs a subset $M_2(u)$ of at most 10 nodes to provide 3-protection to all nodes in $N[u] \setminus M_2(u)$. If $|M_2(u)| = 2$ for any $u \in D_1$ then u arbitrarily selects a neighbor $v \notin M_2(u)$ to provide its own 3-protection. However, any node $u_i \in M_2(u)$ is at least 1-protected since u_i is protected by u. In order to obtain 3-protection, each node $u' \in \bigcup_{u \in D_1} M_2(u)$ which is not 3-protected arbitrarily selects at most two nodes to protect itself. Now all nodes in $\bigcup_{u \in D_1} M_2(u)$ and their selected neighbors are set as active. Let A_3 denote the set of all active nodes in G. Then the set A_3 provides 3-protection to the nodes in G.

In the following, we give the distributed algorithm $\mathcal{D}$ for the p-protection algorithm that is executed at each node u:

Algorithm $\mathcal{D}$ for p-protection subset

Input: An undirected graph G and an integer parameter $p \in \{1, 2, 3\}$

Output: Protection set A_i, $i \in \{1, 2, 3\}$

Run Distributed MIS Algorithm [8]
Let D_1 be the MIS found
If $u \in D_1$
  If $p = 1$
    Select $v \in N(u)$ to protect u
    Set u and v active
  Endif
  If $p \geq 2$
    Construct $M_{p-1}(u)$ of at most $5*(p-1)$ neighbors
    //$M_{p-1}(u)$ covers nodes in $N[u] \setminus M_{p-1}(u)$
      If $|M_{p-1}(u)| = p - 1$
        Add $v \in N(u)$, $v \notin M_{p-1}$ to M_{p-1}
      EndIf
    Set u and the nodes in M_{p-1} active
  EndIf
EndIf
If $u \notin D_1$ and active
  If u is not p-protected
    Select $p - 1$ neighbors and set them active
  EndIf
EndIf
//All active nodes form the p-protection subset A_p

5. Theoretical Properties of Algorithm $\mathcal{D}$

In this section we provide the correctness of the algorithm, that is, we prove that it produces a p-protection subset for all the nodes in the underlying graph. We begin with the following observation without proof:

Observation 5.1: Any node $u \in V$ must have degree at least p to achieve p-protection subset.

Lemma 5.2: Active nodes in G constitute the p-protection subset to the nodes in G.

Proof: In Theorem 4.2, we prove the case when $p = 1$. For $p \in \{2, 3\}$, we prove the Lemma by contradiction. Assume that there is a node $m \in V$ that is not p-protected. Let $p_1 = 2$ and and $p_2 = 3$.

We consider the following three cases:

Case 1: $m \in D_1$. Since $m \in D_1$ and not p_i-protected ($i = \{1, 2\}$) that means it could not form $M_i(m)$ (for p_i-protection) of cardinality of at least p_i (also recall that its cardinality can be at most $5i$) to obtain the desired p-protection. So $|M_i(m)| < p_i$. On the one hand, if $|M_i(m)| = 0$ then this implies $|N(m)| = 0$, a contradiction because the degree of a node in connected graph G cannot be 0. On the other hand, $|M_i(m)| = p_i - 1$ implies

$|N(m)| = p_i - 1$ which is also a contradiction since the degree of m will be at least p_i.

Case 2: $m \in M_i(m')$ for some $m' \in D_1$. Node m is guaranteed to have 1-protection since it is connected to $m' \in D_1$. Then m selects at most $p_i - 1$ nodes from its neighbors if m is not p_i-protected, in order to have the desired p_i-protection. If it cannot find $p_i - 1$ neighbors to protect it, then m has degree less than p_i, a contradiction.

Case 3: m is not set as active. Since D_1 is an MIS in G, m must be connected to some node $m' \in D_1$. If m' could not form $M_i(m')$ of size at least p_i to cover m then either m' has degree less than p_i or m will be included in $M_i(m')$. In both cases, it yields a contradiction since the degree of m' must be greater than p_i in the first situation and in the second situation if $m \in M_i(m')$ then m will be active. □

In the following we prove the main results of the paper.

Theorem 5.3: $|A_2| \leq 78.5 * |OPT_2|$ where OPT_2 is a minimum 2-protection subset and A_2 is the 2-protection subset returned by our algorithm.

Proof: We prove our constant factor approximation by bounding the number of neighbors of a node u that can join in A_2 to provide 2-protection.

There are two different cases to be considered, $u \in D_1$ and $u \notin D_1$.

Case 1: $u \in D_1$. Thus there is no node $v \in D_1$ in u's neighborhood. u selects a set of at most five nodes in $M_1(u)$ to cover nodes in $N(u) \setminus M_1(u)$.

From Theorem 4.1 u can have at most 23 nodes in its two-hop neighborhood which belong to D_1. Consider node $v \in D_1$ which is exactly two hops away from u. Since v selects a set $M_1(v)$ of at most five independent nodes there can be at most $|M_1(v) \cap N(u)| \leq 2$ (see Figure 4 (a)). We show this by the standard area equation [1], [2]. The area equation for the intersection (also known as "vesica piscis" or lens) of the two disks centered on u and v (where v is 2-hop neighbor of u) each with radius R, is at most $A = 1/6(4\pi - 3\sqrt{(3)}) * R^2$. Plugging $R = 1$ into the equation we get $A \leq 1/6(4\pi - 3\sqrt{(3)}) = 1.2283$. Thus we can have at most $5/\pi * 1.2283 = 1.95$ nodes. That is, $|M_1(v) \cap N(u)| \leq 2$. Node u selects at most 5 nodes from $M_1(u)$ and each of the 23 nodes $v \in D_1$ can have at most 2 nodes selected from $N(u)$ to form $M_1(v)$. Thus most $23 * 2 + 5 = 51$ nodes from $N(u)$ are used to provide 2-protection. However, each of these 51 nodes selects at most one node from its neighbor if it is not 2-protected. Thus we can have at most 102 nodes selected from $N(u)$. Since the minimum 2-protection subset OPT_2 requires exactly 2 nodes selected from the neighborhood of any node in V, u selects at most 51 times more nodes from $N(u)$.

Case 2: $u \notin D_1$. The case is similar to the above except u can have $|D_1 \cap N(u)| \leq 5$ nodes in $N(u)$, (see Figure 4(b)). Each of these 5 nodes $v \in D_1 \cap N(u)$ can select at most 5 nodes from $N(u)$ to form $M_1(v)$. Then each node $v' \in M_1(v)$ will be at least 1-protected. If it is not 2-protected, then it can select at most one node from $N(u)$, totalling 5+5*5+5*5=55 nodes selected from $N(u)$. Adding this to 102 nodes we get at most 157 neighbors selected from $N(u)$. Since the minimum 2-protection subset OPT_2 requires exactly 2 nodes selected from the neighborhood of any node in V, u selects at most 78.5 times more nodes from $N(u)$.

Thus our algorithm achieves a constant approximation factor of 78.5 for the 2-protection subset problem. □

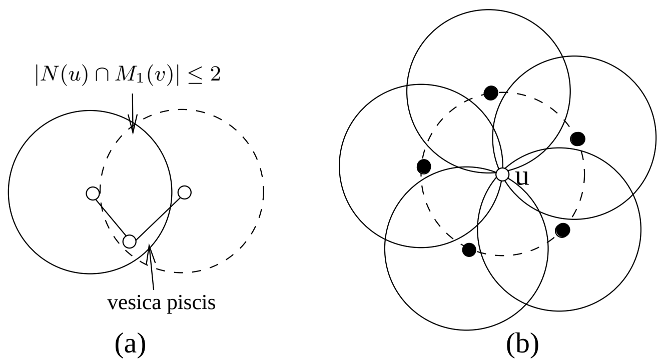

Figure 4. (a) If $v \in D_1$, $v \in N_2[u] \setminus N(u)$ then $|N(u) \cap M_1(v)| \leq 2$ nodes. (b) Node u can have $|D_1 \cup N(u)| \leq 5$ nodes.

Following the same proof technique given above, we provide the bound for the 3-protection case.

Theorem 5.4: $|A_3| \leq 153.66 * |OPT_3|$.

Proof: We have the two following cases:

Case 1: $u \in D_1$. Node u selects a set of at most ten nodes in $M_2(u)$ to cover nodes in $N[u] \setminus M_2(u)$. Thus $|M_2(v) \cap N(u)| \leq 4$ for some node $v \in D_1$ two hops away from u. Therefore, following the same line of argument, we have at most $23 * 4 + 10 + 2 * (23 * 4 + 10) = 306$ nodes selected from $N(u)$. Since the minimum 3-protection subset OPT_3 requires exactly 3 nodes selected from the neighborhood of any node in V, u selects at most 102 times more nodes from $N(u)$.

Case 2: $u \notin D_1$. Each of these 5 nodes $v \in D_1 \cap N(u)$ can select at most 10 nodes from $N(u)$ to form $M_1(v)$. Then each node $v' \in M_1(v)$ will be at least 1-protected. If it is not 3-protected, then it can select at most two nodes from $N(u)$, totalling $5 + 5 * 10 + 2 * 5 * 10 = 155$ nodes selected from $N(u)$. Adding this to 306 nodes we get at most 461 neighbors selected from $N(u)$.

Since the minimum 3-protection subset OPT_3 requires exactly 3 nodes selected from the neighborhood of any node in V, u selects at most 153.66 times more nodes from $N(u)$.

Thus our algorithm achieves a constant approximation factor of 153.66 for the 3-protection subset problem. □

5.1. Time and Message Complexities

Exchanging indices and degrees with direct neighbors can be done in a constant time [15]. Line 1 in our algorithm $\mathcal{D}$ constructs a distributed MIS [8] in a unit disk graph which requires $O(\log^* n)$ time [8] where n is the number of nodes in G. The algorithm in [8] can be applied in any growth bounded graph, where growth boundedness means that there is a constant number of independent nodes in some fixed hop neighborhood. Line 2-20 of algorithm $\mathcal{D}$ can be executed in constant time since nodes only perform local computation which is free [15]. Thus algorithm $\mathcal{D}$ needs $O(\log^* n)$ time.

Each node broadcasts a single message containing its index and degree to its direct neighbors. As mentioned in [8], the algorithm for generating an MIS requires at most $O(|E| \log^* n)$ messages. Also nodes in D_1 inform their neighbors whether they will be active which totals at most $O(n)$ messages. Thus the message complexity of algorithm $\mathcal{D}$ is $O(|E| \log^* n)$.

This suggests the following theorem:

Theorem 5.5: For unit disk graphs the p-protection subset problem ($p \in \{1, 2, 3\}$) can be solved within a constant factor in $O(\log^* n)$ time and with $O(|E| * \log^* n)$ messages.

6. Conclusion and Open Problems

Providing protection to sensors by their neighbors in a sensor network is an interesting problem. The need to protect sensors arises because they are vulnerable to attacks, failure or damage while they monitor objects or provide coverage to a certain area. By having some protector sensors around others allows us to know the status of these sensors and take appropriate actions (replacing the dying and faulty sensors, if possible) in case those sensing sensors fail or run out energy. In this paper, we have studied the problem of protecting the sensors by one another namely, the p-protection subset problem, and provided a distributed solution with a constant approximation factor. To the best of our knowledge, this is the first constant factor approximation ratio to p-self-protection subset problem which is significantly better than the previous $O(\log n)$ approximation ratio.

We have identified some open problems that result from this work. First, it will be interesting to obtain a stronger and smaller constant factor for the p-protection subset problem. We do not know whether it is easy to have a 'small' constant factor bound for the p-protection subset problem when $p > 3$. It is easy to see that following our procedure we can obtain solutions when $p > 3$, however, the constant factor will be much larger as the values of p increase. Providing a 'small' constant for arbitrary values of p is a challenge. Our constant factor approximation is large because we focus only on the worst case behavior, however we believe that the approximation ratio will be much smaller in general cases. Besides, the p-protection subset we find is not necessarily connected, thus, finding a connected p-protection set, that is, the nodes in the protection set induce a connected subgraph, would be interesting. Another challenging open problem is to find a disjoint family of p-protection sets such that each of the sets can be used for a round and then can be replaced by another set for the next the round, and so on, in order to properly balance energy consumption.

References

[1] Pedoe, D.: Circles: A Mathematical View, rev. ed. Washington, DC: Math. Assoc. Amer., 1995.

[2] Rawles, B.: Sacred Geometry Design Sourcebook: Universal Dimensional Patterns. Nevada City, CA: Elysian Pub., pp. 11. 1997.

[3] Wang, Y., Li, X., Zhang, Q.: Efficient Self Protection Algorithms for Static Wireless Sensor Networks. In: Proc. Globecom 2007.

[4] Wang, D., Zhang, Q., Liu, J.: Self-protection for Wirless Sensor Networks. In: Proc. IEEE ICDCS, 2006.

[5] Cardei, M., Wu, J.: Energy-efficient Coverage Problems in Wireless Adhoc Sensor Networks. Computer Communications Journal, Vol, 29, No 4. pp 413-420. 2006.

[6] Personal Communication with Xiang Yang Li and Yu Wang.

[7] Gui, C., Mohapatra, P.: Power Conservation and Quality of Surveilance in Target Trackinng Sensor Networks. In: Proc. ACM MobiCom, 2004.

[8] Schneider, J., Wattenhofer, R.: A Log-Star Distributed Maximal Independent Set Algorithm for Growth-Bounded Graphs. In: Proc. PODC, 2008.

[9] Alzoubi, K. M., Wan, P.J., Frieder, O.: Message-optimal Connected-dominating-set Construction for Routing in Mobile Adhoc Networks. In: Proc. MobiHoc 2002.

[10] Kumar, S., Lai, T.H., Arora, A.: Barrier Coverage with Wireless Sensors. In: Proc. ACM Mobicom, 2004.

[11] Garey M., and Johnson, D.: Computers and Intractibility, a Guide to the Theory of NP-completeness. W. H. Freeman and Company, 1979.

[12] Clark, B., Colborn, C., Johnson, D.: Unit Disk Graphs. Discrete Mathematics, 86:165-177. 1990.

[13] Moscibroda, T., Wattenhofer, R.: Maximizing the Lifetime of Dominating Sets. In: Proc. WMAN, 2005.

[14] Pemmaraju, S., Pirwani, I.: Energy Conservation in Wireless Sensor Networks via Domatic Partitions. In: Proc. MobiHoc 2006.

[15] Peleg, D.: Distributed Computing: A Locality-Sensitive Approach. SIAM, Philadelphia, PA, 2000.