Ad Hoc Networks 36 (2016) 296–307

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Maximum lifetime dependable barrier-coverage in wireless sensor networks☆

Donghyun Kim[a,b,*], Hyunbum Kim[c], Deying Li[d], Sung-Sik Kwon[b], Alade O. Tokuta[b], Jorge A. Cobb[e]

[a] *Division of Algorithms and Technologies for Networks Analysis, Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam*
[b] *Department of Mathematics and Physics, North Carolina Central University, Durham, NC 27707, United States*
[c] *Department of Computer Science, University of North Carolina at Wilmington, Wilmington, NC 28403, United States*
[d] *School of Information, Renmin University of China, Beijing 100872, PR China*
[e] *Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080, United States*

ARTICLE INFO

Article history:
Received 24 November 2014
Revised 3 July 2015
Accepted 6 August 2015
Available online 12 August 2015

Keywords:
Barrier coverage
Wireless sensor network
Maximum lifetime scheduling

ABSTRACT

In a wireless sensor network, a subset of sensor nodes provides a barrier-coverage over an area of interest if the sensor nodes are dividing the area into two regions such that any object moving from one region to another is guaranteed to be detected by a sensor node. Recently, Kumar et al. introduced scheduling algorithms for the maximum lifetime barrier-coverage problem. The algorithms achieved the optimal lifetime by identifying a collection of disjoint subsets of nodes such that each subset in the collection can provide barrier-coverage over the area, and by activating each subset in turn. This introduces a new security problem of these scheduling algorithms called barrier-breach. We show there could be a way to penetrate the area protected by barrier-covers when one barrier-cover is replaced by another. To deal with this issue, we propose three different remedies for the algorithms. In addition, we compare the performance of the three approaches against an upper bound via extensive simulation and make a discussion on the results.

1. Introduction

Wireless sensor network (WSN) is regarded as a decent network technology for a wide range of important applications such as battlefield surveillance, intrusion detection, environmental monitoring, etc. A WSN is composed of a large number of sensor nodes. Each sensor node is equipped with a sensing device, a computing unit, a wireless transceiver and a limited energy source such as a battery. A sensor node can monitor specific phenomenon using the embedded sensing device and forward the data toward a base station [10,11]. In the literature, the coverage provided by a WSN is largely classified into two categories: full-coverage and partial-coverage. A WSN is supporting full-coverage over a target area only if any event happening in the area at any moment is guaranteed to be detected by the WSN [1,3,12–15]. In contrast, a WSN providing partial-coverage may miss some event in an area of interest [2,16–18].

In the literature, a subset of sensor nodes provides barrier-coverage over an area of interest if the sensor nodes are dividing the area into two regions such that any object moving from one region to anther is guaranteed to be detected by a sensor node. As a result, barrier-coverage can be considered as a special case of partial-coverage. There are many

☆ A part of this paper has been appeared in the Proceedings of the *IEEE Global Communications Conference (GLOBECOM 2012)* [4].

* Corresponding author. Tel.: +1 (919) 530– 6567.
E-mail addresses: donghyun.kim@tdt.edu.vn, donghyun.kim@nccu.edu (D. Kim), kimh@uncw.edu (H. Kim), deyingli@ruc.edu.cn (D. Li), skwon@nccu.edu (S.-S. Kwon), atokuta@nccu.edu (A.O. Tokuta), cobb@utdallas.edu (J.A. Cobb).

http://dx.doi.org/10.1016/j.adhoc.2015.08.004
1570-8705/

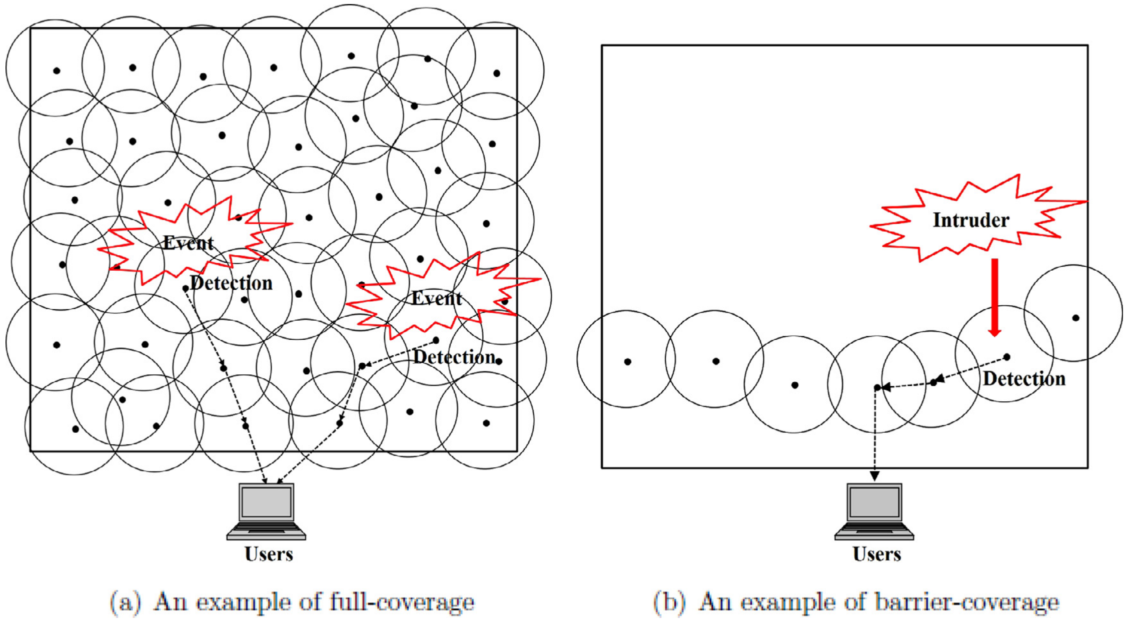


Fig. 1. Illustration of full-coverage and barrier-coverage.

important applications of barrier-coverage such as intrusion detection, and thus it has attracted lots of attentions recently [5–9,19,22–24]. In the rest of this paper, we call a set of sensor nodes providing barrier-coverage over an area simply as a *barrier-cover of wireless sensors*. Fig. 1 illustrates an example of both full-coverage and barrier-coverage.

When it is compared with full-coverage model, barrier-coverage model requires much fewer sensors and thus costs less. Hence, this coverage model has been known to be an attractive approach for various applications such as intrusion detection in which the full-coverage model is somehow excessive. Kumar et al. also introduced the k-barrier-coverage model as a security enhanced model of barrier-coverage. A sensor network provides k-barrier-coverage over an area, where $k \geq 1$ is a given security parameter if any attempt to cross an area covered by the sensor network is guaranteed to be detected by at least k distinct sensors.

In many application scenarios, WSNs are randomly but densely deployed over an area of interest to ensure connectivity. Consequently, it is highly likely that the same target is covered by more than one sensor node simultaneously. Frequently, such a redundancy is appropriately exploited to maximize the lifetime of the sensor networks. For example, if several sensor nodes cover the same target, one can find a sleep-wakeup schedule of the nodes and operate the nodes one by one to maximize the time to cover the target. Clearly, in this way, the total time to cover the target can be extended much longer than the case where all of the sensors are used concurrently. The problem of finding the optimal sleep-wakeup schedule is NP-hard for full-coverage model even if all sensors have equal lifetime. Recently, Kumar et al. [23] have shown that the sleep-wakeup problem for k-barrier-coverage sensor networks is solvable by developing two polynomial time optimal sleep-wakeup algorithms, *Stint* and *Prahari*. The *Stint* considers the case when the remaining battery level of each sensor is same. On the other hand, *Prahari* deliberates on the harder case in which each sensor may have different remaining battery levels.

In this paper, we introduce a new security problem which exists in the sleep-wakeup scheduling algorithms for the maximum lifetime k-barrier-cover of wireless sensors by Kumar et al. To simplify our discussion, we set a security parameter k to 1 and show when a barrier-cover of wireless sensor is replaced with another, the barrier-covers can be useless by one or more locations, namely *barrier-breaches*, which can be exploited by a trespasser to intrude without being detected. Then, we propose three algorithms which can be used to eliminate barrier-breaches from a sleep-wakeup schedule produced by *Stint* and *Prahari*. The first one is applied on the output of the algorithms and the second and third one are applied to the input of the algorithms. At last, we compare the performance of the three approaches against the theoretical upper bound via extensive simulation and analyze the results.

Furthermore, as an extension of [4], we can summarize additional contributions as follows. Firstly, we newly proposed the third algorithm referred as Algorithm 3. Different from previous two algorithms which we considered in [4], the Algorithm 3 focuses on the quality of residual graph by checking the maximum flow value of the residual graph. So, it finally finds the maximum number of node-disjoint paths, which is the maximum number of non-penetrable barriers. Secondly, we implemented Stint and three different algorithms through extensive simulations and various scenarios. Then, we have compared their performances and have shown that the newly proposed Algorithm 3 outperforms other algorithms which we considered in [4]. To show the results, we created all related figure graphs and discussed the results in the new section. Thirdly, we formally defined the introduced problem and enhanced a structure of the paper as well as related studies by considering additional parts and references.

The rest of this paper is organized as follows. Section 2 reviews the related work. In Section 3, we study important preliminaries and introduce the concept of barrier-breach, a new security issue of barrier-covers of wireless sensors. We show how Stint, an optimal scheduling algorithm for maximizing lifetime barrier-coverage in wireless sensor network by Kumar et al. [23], suffers from barrier-breaches. Then, Section 5 describes our three approaches to solve this issue. In Section 6, we thoroughly analyze the performance of the proposed approaches via extensive simulations. We also discuss how our algorithms can be used to solve this problem in Prahari. Finally, we conclude this paper in Section 7.

2. Related works

In the literature, the problem of computing optimal sleep-wakeup schedule of wireless sensors is also known as the maximum lifetime coverage problem. Based on the survey by Cardei and Wu [29], traditional maximum lifetime coverage problems can be classified into three types; area coverage problems [30,31], point (or target) coverage problems [32,33], and minimum breach coverage problems [34]. Initially, the maximum lifetime coverage problem is modeled as a problem of computing the maximum number of disjoint subsets of sensors such that each subset covers the targets or areas successfully. Since the abstracted problem is NP-hard, many approximation algorithms have been proposed. Later, Cardei et al. [14] have shown that there can be a better scheduling of nodes if we do not enforce the subsets to be disjoint.

A barrier-cover of wireless sensors is a set of sensors located between two sides (e.g. top and bottom) such that an object moving from one side to the other side (from top to bottom or from bottom to top) has to be detected by at least one sensor. The barrier-coverage is uniquely different from the traditional sensor coverage since it does not cover whole area or all targets. The notion of barrier-coverage was firstly introduced by Gage [28] in the context of robotic sensors. In [19], Kumar et al. introduced the notion of k-barrier-coverage, which is a generalization of barrier-coverage in a sense that an intruder is guaranteed to be detected by at least k different sensors while moving from one side to the other side. They also defined weak and strong barrier-coverage in a belt region and represented efficient algorithms and optimal deployment pattern to require k-barrier-coverage when sensors are deployed deterministically. In addition, they derived critical conditions for weak barrier-coverage with high probability. In [20], Liu et al. provided the critical conditions for strong barrier-coverage in a strip region and proposed an efficient distributed algorithm to form barriers on long strip region without any constraint on the movement pattern of an intruder to cross the region.

In [23], Kumar et al. studied a sleep-wakeup scheduling problem for k-barrier-cover of wireless sensors, whose goal is to prolong the time to protect an area of interest using a series of alternating barrier-covers. They proposed polynomial time centralized algorithms for the problems, which implies that an optimal sleep-wakeup schedule for k-barrier-coverage of sensors can be obtained within polynomial time unlike the traditional coverage problems. Later, Ban et al. presented a distributed algorithm for this problem which is with low communication overhead and computation cost, and thus is appropriate for larger scale sensor networks [35]. Different from previous global barrier coverage, Chen et al. proposed the concept of local barrier which guarantees the detection of intruder whose trajectory is limited to a slice of the belt area [21]. To maximize the lifetime of local barrier-coverage, they also developed a sleep-wakeup algorithm for this problem.

3. Preliminaries

In this section, we review *Stint* and *Prahari*, which are proposed by [23] and sleep-wakeup scheduling algorithms for k-barrier-coverage. To simplify our discussion, we set k to 1 and introduce a new security problem in barrier-coverage, namely *barrier-breaches*. Then, we describe how barrier-breaches occur when Stint and Prahari are implemented.

In [23], Kumar et al. proved that Stint is an optimal sleep-wakeup algorithm for k-barrier coverage when an energy level of each node is equal. Because of its gravity in our discussion, we firstly describe it in detail in this section. Largely, Stint is composed of the following three steps.

- Step 1: Generate a coverage graph $\mathcal{G}$ with a sensor network with n sensor nodes and sensor node's sensing range r in the target region as well as s and t which are corresponding to the left and right boundary of the region, respectively. There is an edge between two sensor nodes in $\mathcal{G}$ if their sensing ranges are overlapping. A node is connected to s (and t) in $\mathcal{G}$ if its sensing range is touching left border (or right border).
- Step 2: $\mathcal{G}$ is transformed into $\mathcal{G}'$ such that each vertex $u \in V(\mathcal{G})$ is split into two vertices u_{in} and u_{out}, and there exist directional edges: incoming edge and outgoing edge.
- Step 3: From $\mathcal{G}'$, compute a max-flow value m from s and t is computed by using a max-flow algorithm. Note that s and t can be included in each path. Also, the max-flow value m is used to identify m node-disjoint paths. It follows that the max-flow value m is equivalent to the maximum number of node-disjoint paths from s to t. Now, the subset of nodes in each node-disjoint path can provide barrier-coverage over the region.

Now, we describe more details of the three steps of Stint, whose goal is to find maximum number of barrier-covers. Let us assume that the sensing range r of each sensor is equal and also assume the sensing range is set to 1.

Step 1. From a given set N of nodes and a sensor's sensing range r (Fig. 2(a)), Stint creates a coverage graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ (Fig. 2(b)) as follows:

(i) set $V(\mathcal{G}) \leftarrow N$,
(ii) for each $u, v \in V(\mathcal{G})$, there exists an edge, $(u, v) \in E(\mathcal{G})$, only if the distance between u and v is at most 2 (i.e. their sensing range is overlapping). If the distance between u and v is greater than 2, their sensing range would no overlap and there does exist no edge from u and v. Also, the case, the distance between u and v is 2 at the maximum, would not allow for any uncovered space along the path of the closest distance between u and v.

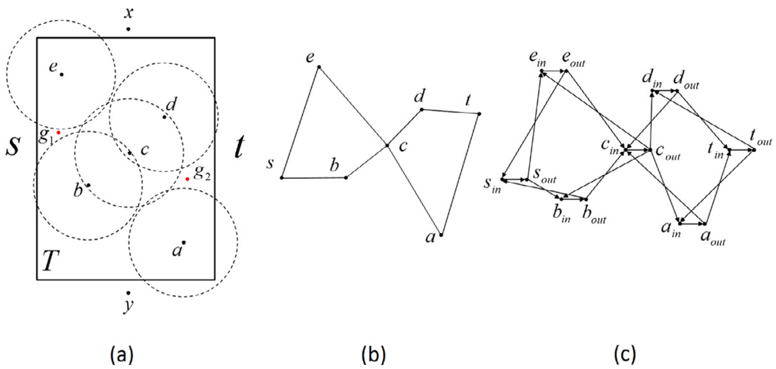


Fig. 2. This figure illustrates how Kumar et al.'s sleep-wakeup algorithm (Stint) for barrier-covers of sensor nodes with the same remaining energy-level works [23]. (a) illustrates a set N of nodes. (b) illustrates a coverage graph $\mathcal{G}$ induced from N. (c) illustrates an induced graph $\mathcal{G}'$ from $\mathcal{G}$. A max-flow value over $\mathcal{G}'$ is equivalent to the number of node-disjoint paths in $\mathcal{G}'$.

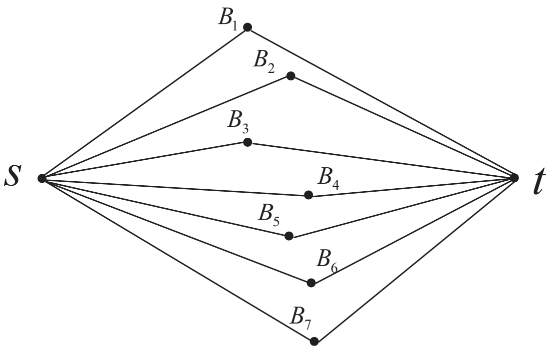


Fig. 3. In the final step of Stint, the output of a max-flow algorithm on a coverage graph $\mathcal{G}$ is used to produce a m different barrier-covers. The covers are activated from the one nearest to the intruder (i.e. $B_1 \to \cdots \to B_7$).

(iii) add virtual source node s and destination node t that are located at two opposite sides of the target region T, and

(iv) for each $u \in V(\mathcal{G})$, $(u, s) \in E(\mathcal{G})$, only if the distance between u to the left border of T is no greater than 1. Similarly, an additional edge (u, t) exists as $(u, t) \in E(\mathcal{G})$ only if the distance between u to the right border of T is no greater than 1.

Step 2. $\mathcal{G}$ is transformed into $\mathcal{G}'$ (Fig. 2(c)) such that each vertex $u \in V(\mathcal{G})$ is split into two vertices u_{in} and u_{out}, and there exists a directional edge, $u_{\text{in}} \to u_{\text{out}}$. For each incoming edge from w to u in $\mathcal{G}$, there is a directional edge from w_{out} to u_{in} in $\mathcal{G}'$. Likewise, there is a directional edge from u_{out} to w_{in} in $\mathcal{G}'$ for each outgoing edge from u to w in $\mathcal{G}$.

Step 3. Then, the max-flow algorithm such as Edmonds–Karp algorithm [25] is applied to $\mathcal{G}'$ to search for the max-flow value m from s to t. The max-flow value m achieved in the previous step is used to identify m node-disjoint paths from s to t in $\mathcal{G}$. Here, we say two paths are said to be node-disjoint only if they do not have any common node in the two paths between s and t (still can have s and t). It follows that the value of the max-flow is equal to the maximum number of node-disjoint paths from s to t in $\mathcal{G}$. The set of nodes in each path among m node-disjoint paths forms one legitimate barrier-cover. The algorithm then alternatively employs the set of nodes in each barrier-cover as shown in Fig. 3. Note that in this figure, we operate the barrier-covers B_i in the increasing order of i, no intruder can penetrate the region until all of the barriers are expired.

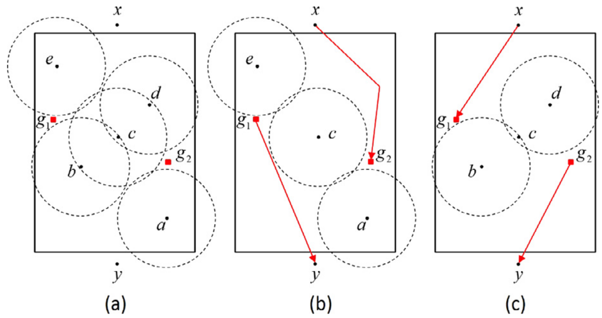


Fig. 4. These figures illustrate two different cases ($x \to g_2 \to y$ and $x \to g_1 \to y$) in each of which an intruder can move along two alternating barrier-covers without being detected even though each of the barrier-cover is flawless.

4. Barrier-breach among alternating barrier-covers

In this section, we introduce a new security problem in Kumar et al.'s sleep-wakeup scheduling algorithms for maximum lifetime barrier-coverage problem. We first introduce definitions of potential-breach-points and barrier-breach and later show how their two algorithms, Stint and Prahari, suffer from this issue.

4.1. Barrier-breach among alternating barrier-covers of wireless sensors

For easier understanding of the readers, we use Fig. 4 for our discussion. Suppose an intruder penetrates from x to y and all sensor nodes a, b, c, d, and e have equivalent lifetime of one time unit. If we form a single barrier-cover with all of the available sensor nodes $\{a, b, c, d, e\}$, we can obtain one legitimate barrier-cover. Alternatively, we can derive two barrier-covers, i.e. two disjoint subsets, $B_1 = \{a, c, e\}$ and $B_2 = \{b, d\}$. Note that each of the subset can form a legitimate barrier-cover.

If we turn on all sensors $\{a, b, c, d, e\}$ simultaneously, then we can protect y from an intruder at x only for one time unit. On the other hand, if we use B_1 and B_2 alternatively, there will be a barrier-cover of wireless sensors splitting x and y during

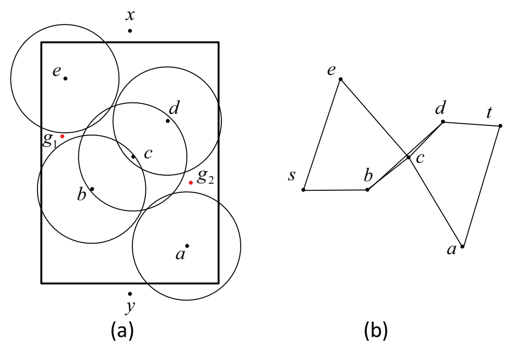


Fig. 5. In this figure, g_1 and g_2 are two potential-breach-points which occurs barrier-breach between two barrier-covers, $B_1 = \{a, c, e\}$ and $B_2 = \{b, d\}$.

two time unit. While this seems nicer, but this strategy can suffer from the following problem, in which an intruder may trespass into y without being detected with two time unit. There are the following two cases.

Case 1. Suppose that we utilize B_1 first. Then, the intruder can move from x to g_2 without being detected (Fig. 4(b)). After the sensors in B_1 are exhausted and those in B_2 are activated, it is possible that the intruder move from g_2 to y (Fig. 4(c)).

Case 2. Suppose that we schedule B_2 first. Then, the intruder can move from x to g_1 without being detected (Fig. 4(c)). After the sensors in B_2 are exhausted and those in B_1 are activated, the intruder can move from g_1 to y (Fig. 4(b)).

From this observation, we can conclude that if we alternatively utilize two barrier-covers which commonly cover some area, there may exist a sequence of movements, e.g. $x \to g_2 \to y$ in Case 1 above, if there exists some locations like g_1 and g_2 that an intruder can exploit to trespass without being detected.

Definition 1 (potential-breach-points)**.** A location g_i (such as g_1 and g_2 in Fig. 5) in a Euclidean space is called a "potential-breach-points" between two alternating barrier-covers B_1 and B_2 which are observing any intruder trying to move from a position x to another position y if

(i) x is outside the sensing range of B_1 and B_2,
(ii) one of the barrier-covers cannot detect an intruder moving from its previous location to g_i, and
(iii) the other barrier-cover cannot detect an intruder moving from g_i to anther location y.

Note that this definition can be easily extended among more than two alternating barrier-covers.

Definition 2 (barrier-breach)**.** A "barrier-breach" is an activity that current barriers is to be broken due to potential-breach-points. So, after "barrier-breach", any moving object or intruder can pass through from a position x to another position y without any detection by sensors.

Definition 3 (non-penetrable barrier-coverage)**.** A set of barrier-covers with corresponding schedule forms a non-penetrable barrier-coverage of sensors only if there are no potential-breach-points between any two alternating barrier-covers in the set. The "non-penetrable barrier-coverage" is equivalent to the "non-crossing barrier-coverage".

Definition 4 (crossing barrier-covers)**.** Two barrier-covers B_1 and B_2 are crossing with each other if there are two nodes v_1, $v_2 \in B_1$ and $u_1, u_2 \in B_2$ such that the line connecting v_1 and v_2 is crossing with the line connecting u_1 and u_2, where v_1 and v_2 are neighbors and also u_1, u_2 are neighbors.

Definition 5 (planar graph)**.** A graph in 2D Euclidean space is a planar graph if the graph has no two edges crossing with each other.

Theorem 1. *There exists a schedule in Stint for a set of barrier-covers which does not allow any barrier-breach between any pair of barrier-covers if no two barrier-covers in the set are crossing each other or a $\mathcal{G}$ induced from a set N of sensors by Step 1 of Stint is a planar graph.*

Proof. Suppose we have an area of interest T and an intruder moves from the top side x to the bottom side y as like Fig 4. Also, suppose we have a set of noncrossing barrier-covers, $B_1, B_2, \ldots, B_l$, which are sorted in the increasing order of their distance toward x (i.e. Fig. 3). Now, suppose we utilize the barrier-covers in this order, but there is a barrier-breach. Then, there should be some B_i and potential-breach-points p located above the sensing range of all nodes in B_i such that when B_i is turned off and B_{i+1} is turned on, an intruder at p is lower than the sensing range of a node in B_{i+1}. However, this is contradicting to our assumption that B_i and B_{i+1} are non-crossing and B_i is nearer to x than B_{i+1}. As a result, there is no such p, and this lemma is true. □

Definition 6 (MaxLNB)**.** Given a set of wireless sensor nodes deployed over an area T, the maximum lifetime non-penetrable barrier-coverage (MaxLNB) problem is to find the sleep-wakeup schedule of nodes such that the time to continuously provide a barrier-cover over T without suffering from the barrier-breach problem is maximized.

4.2. The barrier-breach problem in Stint

By Theorem 1, if it is guaranteed that Stint generates a set of barrier-covers that does not cross with each other, Stint would not have any barrier-breach. However, Stint does not provide any guarantee that no two barrier-covers overlap with each other. In detail, suppose we have a square area T of interest with a set of sensor nodes $N = \{a, b, c, d, e\}$ as Fig. 5. Assume that an intruder moves from top to bottom only over square area T. The initial step of Stint is to transform a new graph $\mathcal{G} = (V, E)$ from T and N as we mentioned it in Section 4.1. We can simply derive $\mathcal{G}$ shown in Fig. 5(b) from Fig. 5(a). The second step of Stint is to apply a max-flow algorithm such as Edmonds–Karp algorithm for $\mathcal{G}'$ derived from $\mathcal{G}$ which has been described in Section 4.1. As a final step in Stint, the max-flow value m by Edmonds–Karp algorithm can be considered as m node-disjoint paths from source s to t. That is, the nodes in one node-disjoint path among m paths build up a barrier-cover. In [23], they have shown that the max-flow value m over $\mathcal{G}'$ is equivalent to the maximum time m which a set of sensors with the same remaining energy level is able to form barrier-covers in given area T. Absolutely, it must be true if the issue is considered without barrier-breaches. But, during m units of time, their solution may not result in a non-penetrable barrier-coverage. For example, if the max-flow value is two where a max-flow algorithm is applied to $\mathcal{G}'$, then there are two node-disjoint paths, namely, P_1 and P_2, respectively. If P_1 and P_2 cross with each other, we can verify that there exist some potential-breach-points such as g_1 and g_2 in Fig. 5(a). Note that we can choose only P_1 and P_2 even if we consider two node-disjoint paths. Accordingly, even though it is impossible to construct a non-penetrable barrier-coverage over T during two units of time, it maintains only for one unit of time. After all, either P_1 or P_2 should be chosen in order to remove the potential-breach-points. In essence, this problem happens because the max-flow algorithm cannot assure that two node-disjoint barrier-covers would not cross with each other whatever the activation schedule of the covers Stint generates.

Still, this is a very significant issue considering the applications of barrier-coverage model such as an intrusion detection in WSN. Let us assume that there is a system that still remains secure even though an intruder knows all information including sleep-wakeup scheduling algorithms and locations of sensor nodes. Then, we may conclude such a system is truly secure. But, it is a very strong assumption (and too optimistic) that the attacker does not have any knowledge about those.

4.3. The barrier-breach problem in Prahari

Now, we discuss how Prahari suffers from the barrier-breach problem. We note the major difference between Stint and Prahari is that Prahari assumes the remaining energy level of each node can be different. Whereas Stint creates a coverage graph $\mathcal{G}$ with uniform link capacity from a given network, Prahari generates a coverage graph $\mathcal{G}_L$ with non-uniform link capacity. Except the difference, both Stint and Prahari execute same procedures as follows: (i) applying a max-flow algorithm, (ii) searching for the maximum number of independent paths, (iii) verifying the maximum number of barrier-covers from the computed max-flow. Based on these properties, we argue that if there are any two crossing paths among the found maximum number of independent paths, Prahari also may suffer from the barrier-breach problem as Stint does.

5. Three approaches to extend the lifetime of non-penetrable barrier-cover of sensors

In this section, we introduce three different approaches to provide barrier-coverage without suffering from the barrier-breach problem. To make our discussion easier, we first discuss how the approaches can be used to deal with the barrier-breach problem in the output of Stint. Later, we also discuss how to handle the problem in the output of Prahari.

5.1. Greedy-Cover-Eraser: first approach for barrier-breach problem

By Theorem 1, there exists a schedule of multiple barrier-covers if no two covers overlap with each other. However, Stint produces an output, a set of barrier-covers which can cross with each other. Therefore, one can eliminate all

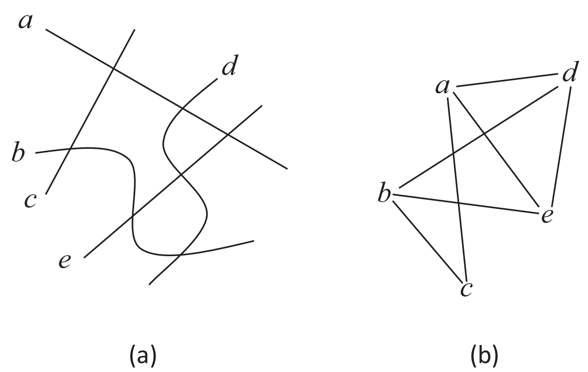


Fig. 6. This figure illustrates that the problem of computing the maximum number of non-crossing paths can be converted into the maximum independent set problem, which is NP-hard. (a) is an example of crossing barrier-covers after Stint. As it can be seen from (a), some barrier-covers are crossing with each other while some others are not crossing. Then, finding the maximum number of non-crossing paths is equivalent to finding the maximum independent set. (b) is an example to compute the maximum independent set. Then, (a) can be transformed into the graph in (b).

potential-breach-points by not using some of the barrier-covered produced by Stint. Suppose we have a set of barrier-covers after executing Stint. Then, some covers are crossing with each other while some others are not (see Fig. 6(a)). Then, a new graph $G = (V, E)$ from the barrier-covers is constructed as follows (Fig. 6(b)).

(i) For each barrier-cover u, add a corresponding vertex to V.
(ii) For each pair of barrier-covers such that u and v cross with each other, add an edge (u, v) to E.

In this way, the problem of computing the maximum number of non-crossing barrier-covers (Fig. 6(a)) is converted into the problem of finding the maximum number of nodes in G such that they are not adjacent with each other (Fig. 6(b)). It is noteworthy that the transformed problem is equivalent to the maximum independent set (MIS) problem which is NP-hard. Therefore, we can find an approximate solution of this problem by exploiting an existing approximation algorithm for the MIS problem such as the $O(|n|/(\log|n|)^2)$-approximation by Boppana and Halldorsson [26], where n is the number of sensors. And the greedy in Halldorsson and Radhakrishnan [27] is borrowed. Algorithm 1 is the pseudocode of this first approach, namely Greedy-Cover-Eraser.

5.2. Greedy-Edge-Eraser: second approach for barrier-breach problem

In this section, we introduce our second approach to deal with the barrier-breach problem.

Theorem 2. *Consider a coverage graph $\mathcal{G}$ in Stint. Then, there exists a subgraph $\mathcal{G}_p$ of $\mathcal{G}$ such that $\mathcal{G}_p$ is a planar graph and the optimal lifetime of non-penetrable barrier-coverage of a set N of sensors in $\mathcal{G}$ is equivalent to the max-flow value m in $\mathcal{G}_p$.*

Proof. Suppose if we find an optimal lifetime non-penetrable barrier-cover of N. Then, by Theorem 1, the

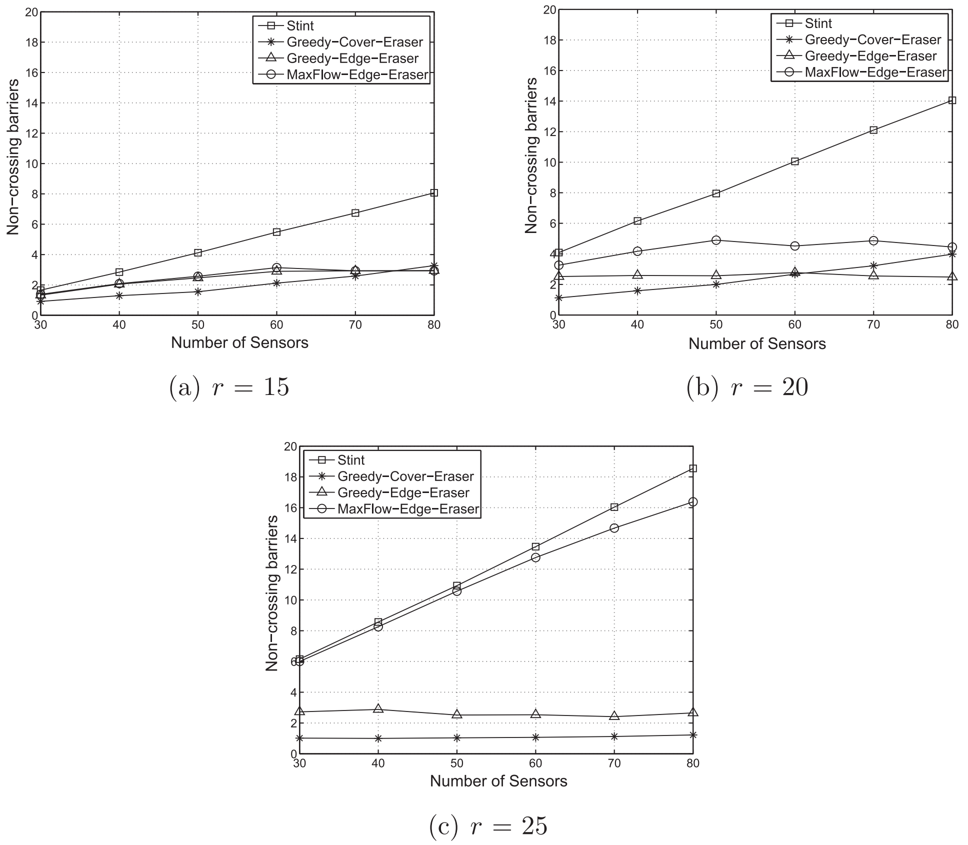


Fig. 7. Comparison of non-crossing barriers for different radius by Stint and three different approaches in 100×100 region.

Algorithm 1 Greedy-Cover-Eraser (N, T).

Set max-flow value $m \leftarrow 0$ and barrier-covers set $B \leftarrow \emptyset$.
Apply Stint with N as the input and obtain the set B of m node-disjoint barrier-covers.
Induce a new graph $G = (V, E)$ such that $V \leftarrow x$ for each barrier-cover $x \in B$ and $E \leftarrow (x, y)$ for each pair of barrier-covers $x, y \in B$ which are crossing with each other.
Set non-crossing barrier-cover set $O \leftarrow \emptyset$.
while $V \neq \emptyset$ **do**
    For each node v in $G = (V, E)$, compute the degree of v in G, $\deg(v)$, which is equivalent to the number of edges ending at v. Also, compute the set of neighbors of v, $Ne(v)$.
    Find v from V such that $v = \arg\min_{u \in V} \deg(u)$. v and any edge neighboring to v is also removed from E. Then, set $G \leftarrow G \setminus \{v \cup Ne(v)\}$.
    Set $O \leftarrow O \cup \{v\}$.
end while
Return O and $m' = |O|$.

---

non-penetrable barrier-cover can be split into a set of m node-disjoint paths. By the definition of the non-penetrable barrier-cover, the paths are non-overlapping with each other, and thus the union of them should be a planar graph. Therefore, this theorem is true. □

Since the integer linear programming is NP-hard, it is unlikely for us to obtain an optimal solution for it within a polynomial time. Thus, we will use the following theorem to tackle such a high complexity.

Theorem 3. *Consider $\mathcal{G}$ induced from N by Stint. Then, there exists a planar graph $\mathcal{G}_p$, which is a subgraph of $\mathcal{G}$ such that the max-flow algorithm can compute an optimal lifetime non-penetrable barrier-cover of N.*

Proof. This is true by Theorems 1 and 2. □

By Theorem 3, the rest of our job is to find such a subgraph $\mathcal{G}_p$ from a given $\mathcal{G}$ by removing subset of crossing edges. For this purpose, we iteratively select an edge which is crossing the maximum number of the other edges and remove the chosen edge from $\mathcal{G}_p$ until no two edges are crossing with each other. Then, the resulting graph will be a planar subgraph $\mathcal{G}_p$ of $\mathcal{G}$. Now, replace $\mathcal{G}_p$ with $\mathcal{G}$ in Step 1 of Stint. By Theorem 1, Stint will produce a feasible solution to our problem. Algorithm 2 is the pseudocode of this greedy approach.

5.3. MaxFlow-Edge-Eraser: third approach for barrier-breach problem

In this section, we introduce another way to deal with the barrier-breach problem. In Algorithm 2, we introduce a way to induce a planar graph $\mathcal{G}_p$ from $\mathcal{G}$ such that an output of

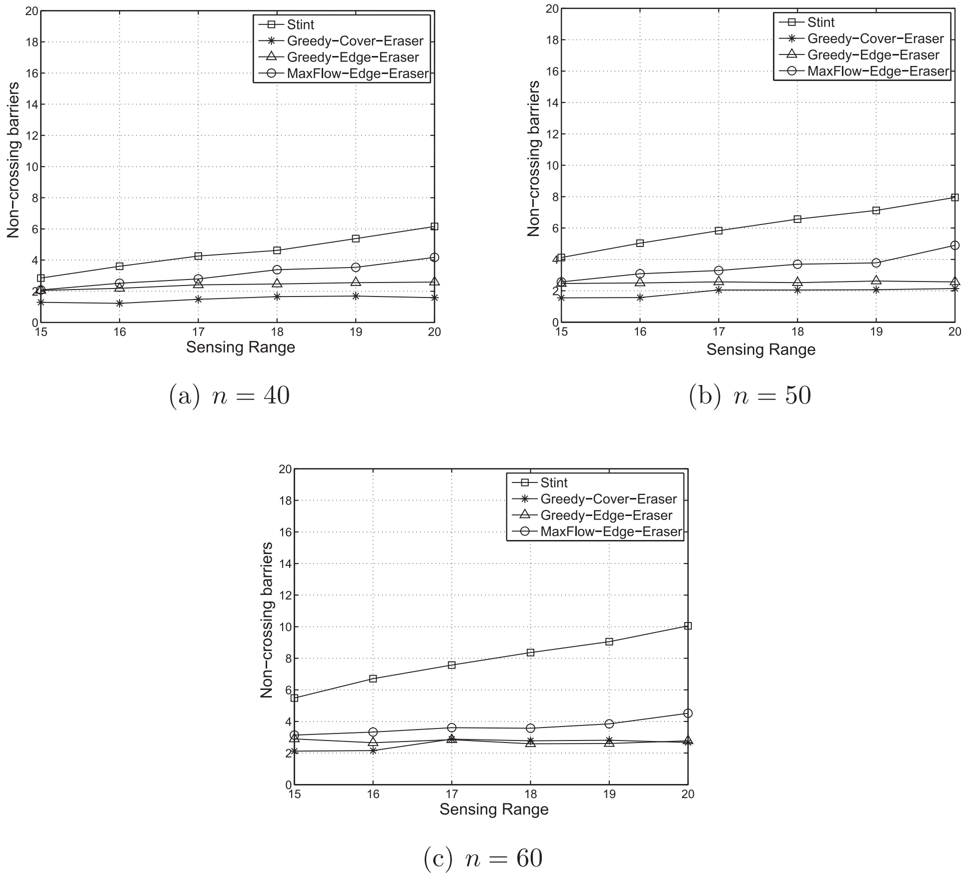


Fig. 8. Comparison of non-crossing barriers for different number of sensors by Stint and three different approaches in 100 × 100 region.

Algorithm 2 Greedy-Edge-Eraser (N, T).

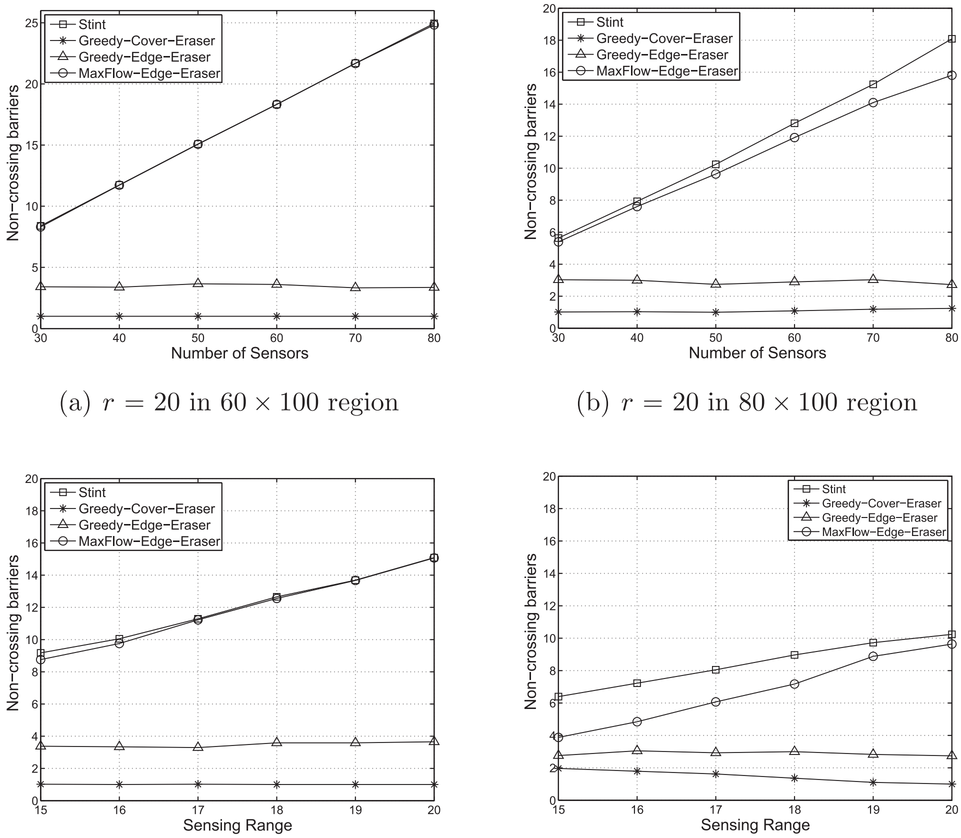
With N and r as inputs, apply the first step of Stint (on Pages 6, 7, 8) and obtain coverage graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$.
Set $\mathcal{G}_p \leftarrow \mathcal{G}$.
while there is no two edges in $\mathcal{G}_p$ crossing with each other **do**
  Find an edge $e_{\max} \in E(\mathcal{G}_p)$ such that $e_{\max}$ is an edge crossing with the maximum number of other edges in $\mathcal{G}_p$.
  Set $E(\mathcal{G}_p) \leftarrow E(\mathcal{G}_p) \setminus \{e_{\max}\}$.
end while
Now, $\mathcal{G}_p$ is a planar graph. Apply the rest of the steps of Stint with $\mathcal{G}_p$ and obtain a set O of node disjoint non-crossing covers.
Return O and $m' = |O|$.

Stint over the planar graph is free from the barrier-breach problem. However, this strategy can may result in a set of barrier-covers with significantly smaller size, i.e. the number of available covers could be much smaller. This is mainly due to the fact that when a greedy strategy is applied to eliminate one of two crossing edges and try to build a planar graph $\mathcal{G}_p$, we do not consider the global effect of such an edge elimination. That is, when we have multiple candidates to eliminate, we better remove one of them such that the maximum flow of residual graph without that edge to be eliminated is maximized. For this purpose, we modified Algorithm 2 and then introduce another approach, MaxFlow-Edge-Eraser.

Similar to Greedy-Edge-Eraser, MaxFlow-Edge-Eraser also computes a planar subgraph $\mathcal{G}_p$ from an induced coverage graph $\mathcal{G}$ via Step 1 of Stint. The major difference between MaxFlow-Edge-Eraser and Greedy-Edge-Eraser is that while Greedy-Edge-Eraser is repeatedly removing an edge which is crossing with the most number of the other edges and focuses on the construction of a planar subgraph itself, MaxFlow-Edge-Eraser is also focusing on the quality of residual graph by checking the maximum flow value of the residual graph. Algorithm 3 is the formal definition of MaxFlow-Edge-Eraser.

5.4. *Adoption of our three approaches to Prahari*

So far, we applied our three approaches to Stint. Similarly, we can apply our three approaches to Prahari. Note that the major difference between Stint and Prahari is that Prahari assumes the remain energy level of each node can be different. That is, although Stint generates a coverage graph $\mathcal{G}$ with uniform link capacity from a given network, Prahari forms a coverage graph $\mathcal{G}_L$ with non-uniform link capacity. Using non-uniform link capacity, we still compute the maximum number of node-disjoint paths by max-flow



(a) $r = 20$ in 60×100 region (b) $r = 20$ in 80×100 region

(c) $n = 50$ in 60×100 region (d) $n = 50$ in 80×100 region

Fig. 9. Comparison of non-crossing barriers for different width size of region by Stint and three different approaches.

Algorithm 3 MaxFlow-Edge-Eraser (N, T).

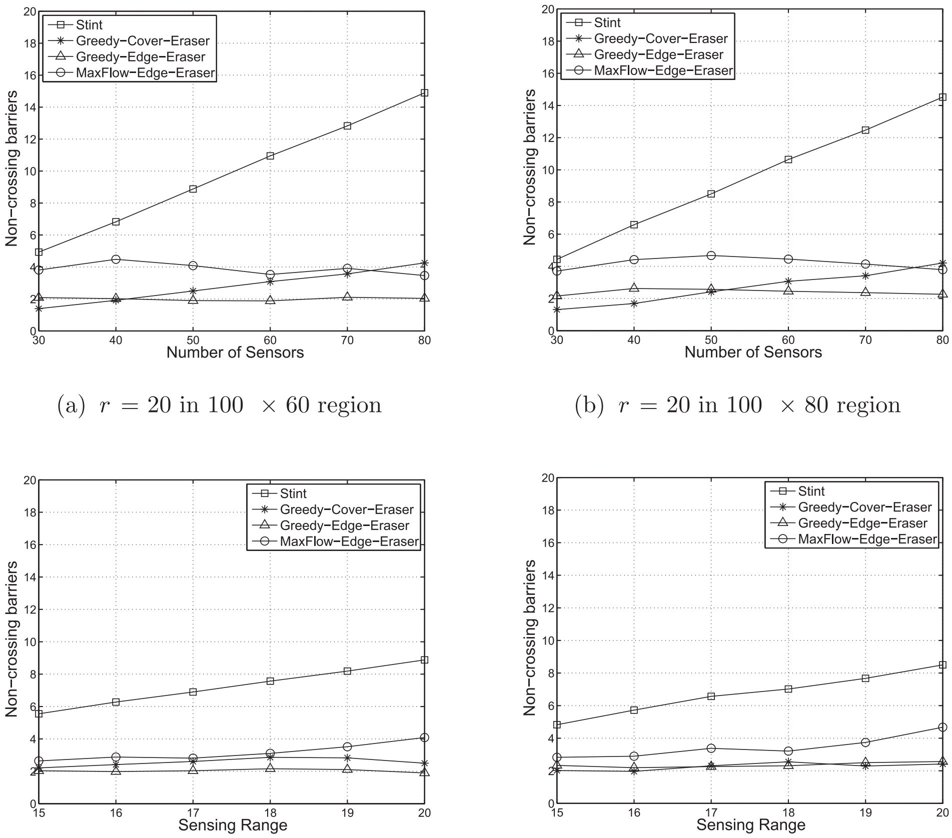
With N and r as inputs, apply the first step of Stint (on Pages 6, 7, 8) and obtain coverage graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$.
Set $\mathcal{G}_p \leftarrow \mathcal{G}$.
while there is no two edges in $\mathcal{G}_p$ crossing with each other **do**
    For each edge $e \in E(\mathcal{G}_p)$, compute the max flow F_e from s to t over $\mathcal{G}_p \setminus \{e\}$, i.e. the residual graph of $\mathcal{G}_p$ without e.
    Find $e_{\max}$ such that $e_{\max} = \arg\max_{e \in E(\mathcal{G}_p)} F_e$.
    Set $E(\mathcal{G}_p) \leftarrow E(\mathcal{G}_p) \setminus \{e_{\max}\}$.
    Run Stint itself by generating a coverage graph $\mathcal{G}_p$ using the updated $E(\mathcal{G}_p)$.
end while
Using $\mathcal{G}_p$, calculate a set O of node disjoint non-crossing covers.
Return O and $m' = |O|$.

algorithm, which are considered as barrier-covers as Stint does. Such barrier-covers by Prahari still suffer from the barrier-breach problem as we mentioned it in Section 4.3. Therefore, for barrier-covers by Prahari, we can use our three approaches in order to remove the crossing edges among barrier-covers.

6. Simulation results and analysis

In this section, we analyze the performance of Stint and three approaches which we have presented in Section 5. Before analyzing the proposed approaches, we would like to represent the difficulty of finding non-crossing barriers m. To do so, we can calculate the total number of crossing edges among sensor nodes after n sensor nodes are deployed in the field and the network topology with sensing range r is constructed using unit disk graph property. For example, if we consider 100 different graphs with in 100×100 square area and each graph has the number of sensors is 50 and the sensing range $r = 20$, then our experiment shows about 1370 total number of crossing edges among sensor nodes as an average value for 100 different graphs. From those many crossing edges, finding maximum number of non-crossing barriers (flows), value m, is surely difficult and is critical issue to solve barrier-breach problem, which we have found from [23]. Our heuristic approaches provide the possible solutions of the barrier-breach problem to find maximum value m.

We simulated the three approaches by our various scenarios, which we used up in several squares or rectangle shaped



(a) $r = 20$ in 100×60 region (b) $r = 20$ in 100×80 region

(c) $n = 50$ in 100×60 region (d) $n = 50$ in 100×80 region

Fig. 10. Comparison of non-crossing barriers for different height size of region by Stint and three different approaches.

areas with size of width 100×100, 60×100, 80×100, 100×60 and 100×80, respectively. Also, n sensor nodes are randomly deployed in the each area. Each experiment represents the average result of 100 different graphs. The number of nodes is ranging from 30 to 80 and the radius of sensors is ranging from 15 to 25 in our simulations. Note that our simulations consider Stint as an upper bound. That is, the result values of all proposed algorithms should be less than values by Stint because all algorithms will remove crossing barrier-covers from Stint to construct non-crossing barriers which solves a barrier-breach problem. And such a removal will cause the result value by each algorithm to be smaller than Stint's one. But, we want to make sure that such a result does not mean that Stint simply outperforms other algorithms because Stint does not provide non-crossing barriers and Stint's result is used as the only theoretical upper bound for performance comparison among algorithms. Therefore, we conclude that the third approach, MaxFlow-Edge-Eraser, outperforms the Greedy-Cover-Eraser and Greedy-Edge-Eraser as a whole when we have checked the performance by various simulations. Now, we analyze the results of different experiments in more detail.

As the first performance analysis in our contribution, we compare Stint, Greedy-Cover-Eraser, Greedy-Edge-Eraser and MaxFlow-Edge-Eraser for value m, respectively. Fig. 7 shows comparison of value m by Stint and our three approaches with different sensing range r in 100×100 square area. Let us consider that Stint is an upper bound for value m. First of all, MaxFlow-Edge-Eraser outperforms other approaches. When the number of sensor nodes increases, the third approach's performance is closer to Stint than other approaches. Interestingly, when the network has the bigger sensing range r in 100×100 square area, MaxFlow-Edge-Eraser shows better performance than the network with small sensing r. That is, the m value of MaxFlow-Edge-Eraser is closer to Stint's value m, which is an upper bound, when considered with more range r. As shown in Fig. 7(a)–(c), Stint and MaxFlow-Edge-Eraser have significant increases for value m as the radius r increases in a graph. On the other hand, Greedy-Cover-Eraser and Greedy-Edge-Eraser do not show any significant increase. On the contrary, Greedy-Cover-Eraser and Greedy-Edge-Eraser have slight decrease when they used bigger r in the network.

In our second group of simulations, the performance of Stint and three approaches are compared for value m. Fig. 8 represents the performance for value m in 100×100 square region by Stint and three algorithms with different number of sensors, when the network has $n = 40, 50, 60$, respectively. When we verify the results of Fig. 8, as the transmission range r increases from 15 to 20, MaxFlow-Edge-Eraser shows the better performance than other approaches, too.

At the third scenario, we evaluate the performance of Stint and three approaches with rectangle region with different width size of simulation area. Fig. 9(a) and (b) shows the results with same radius $r = 20$ in different width 60×100 and 80×100, respectively. As a whole, the value m by Stint and three approaches in 60×100 is bigger than it in 80×100 because the narrow width allows it to have low crossing probability of the edges in flows. Under these environments, MaxFlow-Edge-Eraser still shows better result than other approaches. Compared with 100×100 area, the networks with smaller width allows MaxFlow-Edge-Eraser to much closer to Stint's result than 100×100 area. Fig. 9(c) and (d) represents the results with the same number of sensors $n = 50$ by different width networks area 60×100, 80×100. Similar to Fig. 9(a) and (b), we have checked that the value m in 60×100 of Stint and all approaches except Greedy-Cover-Eraser is bigger than it in 80×100 as a whole. Also, we verified that MaxFlow-Edge-Eraser still outperforms other algorithms in networks with different width sizes.

Lastly, we have simulated Stint and three approaches in order to verify the results of rectangle squared region with different height size of region. Fig. 10(a) and (b) describes the performance with same radius $r = 20$ by different height 100×60, 100×80. Interestingly, Greedy-Cover-Eraser has better performance than Greedy-Edge-Eraser and MaxFlow-Edge-Eraser when the number of nodes n is only 80. But, for other cases, MaxFlow-Edge-Eraser still outperforms Greedy-Cover-Eraser and Greedy-Edge-Eraser. Similarly, Fig. 10(c) and (d) shows the results with the same number of sensor nodes $n = 50$ by different heights.

7. Summary and concluding remarks

In this paper, we defined a new security problem in the existing sleep-wakeup scheduling algorithms in order to maximize lifetime barrier-coverage of wireless sensors. By identifying a set of points, namely potential-breach-points, an intruder can penetrate into the area without any detection of sensors by alternating barrier-covers, which there exists barrier-breach. Such potential-breach-points are found when one barrier-cover of sensors is replaced by another barrier-cover crossing with it. We proved there exists a schedule for a set of barrier-covers that does not allow any barrier-breach if two barrier-covers do not cross each other. To solve the problem, we proposed three heuristics that can be applied to both Stint and Prahari. Through extensive simulations, we evaluate their performances against the theoretical upper bound and analyzed the results. As a future work, we plan to study a distributed algorithm for the problem with a low complexity.

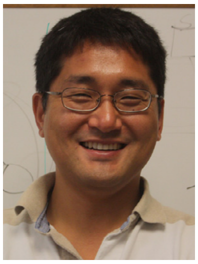
Acknowledgments

This research was supported in part by US National Science Foundation (NSF) CREST No. HRD-1345219. This paper was jointly supported by National Natural Science Foundation of China under grant 91124001, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China 10XNJ032.

References

[1] J. Gao, J. Li, Z. Cai, H. Gao, Composite event coverage in wireless sensor networks with heterogeneous sensors, in: Proceedings of the 34th Annual IEEE International Conference on Computer Communications (INFOCOM 2015), 2015.
[2] Y. Li, C. Ai, Z. Cai, R. Beyah, Sensor scheduling for p-percent coverage in wireless sensor networks, Cluster Comput. 14 (1) (2011) 27–40.
[3] C. Vu, Z. Cai, Y. Li, Distributed energy-efficient algorithms to maximize network lifetime for coverage problem in adjustable sensing ranges wireless sensor networks, Discrete Math. Algorithms Appl. (DMAA) 1 (3) (2009) 299–317.
[4] D. Kim, J. Kim, D. Li, S.-S. Kwon, A.O. Tokuta, On sleep-wakeup scheduling of non-penetrable barrier-coverage of wireless sensors, in: Proceedings of the IEEE Global Communications Conference (GLOBECOM 2012), Anaheim, CA, USA, December 3–7, 2012, pp. 321–327.
[5] J.-L. Lee, D. Kim, L. Fan, H. Chang, Barrier-coverage for city block monitoring in bandwidth sensitive vehicular adhoc networks, in: Proceedings of The IEEE 10th International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2014), Maui, Hawaii, December 19–21, 2014.
[6] H. Luo, H. Du, D. Kim, Q. Ye, R. Zhu, J. Zhang, Imperfection better than perfection: Beyond optimal lifetime barrier coverage in wireless sensor networks, in: Proceedings of The IEEE 10th International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2014), Maui, Hawaii, December 19–21, 2014.
[7] D. Li, B. Xu, Y. Zhu, D. Kim, W. Wu, Minimum (k, ω)-angle barrier coverage in wireless camera sensor networks, Int. J. Sens. Netw. (IJSNET) 19 (2) (2015).
[8] L. Guo, D. Kim, D. Li, W. Chen, A.O. Tokuta, Constructing belt-barrier providing β-quality of monitoring with minimum camera sensors, in: Proceedings of the 23rd International Conference on Computer Communications and Networks (ICCCN 2014), Shanghai, China, August 4–7, 2014.
[9] B. Xu, D. Kim, D. Li, J. Lee, H. Jiang, A.O. Tokuta, Fortifying barrier-coverage of wireless sensor network with mobile sensor nodes, in: Proceedings of the 9th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2014), Harbin, China, June 23–25, 2014.
[10] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Netw. 38 (4) (2002) 393–422.
[11] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Comput. Netw. 52 (12) (2008) 2292–2330.
[12] C. Huang, Y. Tseng, The coverage problem in a wireless sensor network, in: Proceedings of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), 2003.
[13] H. Zhang, J. Hou, On deriving the upper bound of α-lifetime for large sensor networks, in: Proceedings of The 5th ACM International Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc), 2004.
[14] M. Cardei, M. Thai, Y. Li, W. Wu, Energy-efficient target coverage in wireless sensor networks, in: IEEE Annual International Conference on Computer Communications (INFOCOM), Miami, USA, 2005.
[15] M.T. Thai, Y. Li, F. Wang, D.-Z. Du, $O(\log n)$-localized algorithms on the coverage problem in heterogeneous sensor networks, in: Proceedings of the 26th IEEE International Performance Computing and Communications Conference (IPCCC), 2007.
[16] S. Gao, X. Wang, Y. Li, p-percent coverage schedule in wireless sensor networks, in: The International Conference on Computer Communications and Networks (ICCCN), St. Thomas, Virgin Islands, August 3–7, 2008.
[17] C.T. Vu, G. Chen, Y. Zhao, Y. Li, A universal framework for partial coverage in wireless sensor networks, in: IEEE International Performance Computing and Communications Conference (IPCCC), Phoenix, AZ, December 14–16, 2009.
[18] Y. Li, C. Vu, C. Ai, G. Chen, Y. Zhao, Transforming complete coverage algorithms to partial coverage algorithms for wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. (TPDS) 22 (4) (2011) 695–703.
[19] S. Kumar, T.H. Lai, A. Arora, Barrier coverage with wireless sensors, in: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom), 2005.
[20] B. Liu, O. Dousse, J. Wang, A. Saipulla, Strong barrier coverage of wireless sensor networks, in: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2008.
[21] A. Chen, S. Kumar, T.H. Lai, Barrier coverage with wireless sensors, in: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom), 2005.
[22] A. Saipulla, C. Westphal, B. Liu, J. Wang, Barrier coverage of line-based deployed wireless sensor networks, in: Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM), 2009.

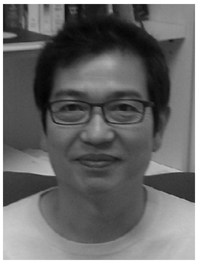
[23] S. Kumar, T.H. Lai, M.E. Posner, P. Sinha, Maximizing the lifetime of a barrier of wireless sensors, IEEE Trans. Mobile Comput. (TMC) 9 (2010) 1161–1172.
[24] H. Yang, D. Li, Q. Zhu, W. Chen, Y. Hong, Minimum energy cost *k*-barrier coverage in wireless sensor networks, in: Proceedings of the 5th International Conference on Wireless Algorithms, Systems, and Applications (WASA), 2010.
[25] J. Edmonds, R.M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, J. ACM 19 (2) (1972) 248–264.
[26] R. Boppana, M.M. Halldorsson, Approximating maximum independent sets by excluding subgraphs, BIT Numer. Math. 32 (2) (1992) 180–196.
[27] M.M. Halldorsson, J. Radhakrishnan, Greed is good: approximating independent sets in sparse and bounded-degree graphs, Algorithmica (1997) 145–163.
[28] D. Gage, Command control for many-robot systems, in: Proceedings of the Nineteenth Annual AUVS Technical Symposium (AUVS-92), 1992.
[29] M. Cardei, J. Wu, Energy-efficient coverage problems in wireless ad hoc sensor networks, Comput. Commun. 29 (4) (2006) 413–420. (special issue on Sensor Network)
[30] S. Slijepcevic, M. Potkonjak, Power efficient organization of wireless sensor networks, in: Proceedings of IEEE International Conference on Communications (ICC), 2001.
[31] D. Tian, N.D. Georganas, A coverage-preserving node scheduling scheme for large wireless sensor networks, in: Proceedings of the 1st ACM Workshop on Wireless Sensor Networks and Applications, 2002.
[32] M. Cardei, D.-Z. Du, Improving wireless sensor network lifetime through power aware organization, Wireless Netw. 11 (3) (2005) 333–340.
[33] K. Kar, S. Banerjee, Node placement for connected coverage in sensor networks, in: Proceedings of WiOpt 2003: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2003.
[34] X.-Y. Li, P.-J. Wan, O. Frieder, Coverage in wireless ad-hoc sensor networks, IEEE Trans. Comput. 52 (2002).
[35] D. Ban, Q. Feng, G. Han, W. Yang, J. Jiang, W. Dou, Distributed scheduling algorithm for barrier coverage in wireless sensor networks, in: Proceedings of the 2011 Third International Conference on Communications and Mobile Computing (CMC), 2011.



Donghyun Kim received the B.S. degree in electronic and computer engineering from the Hanyang University, Ansan, Korea (2003), and the M.S. degree in computer science and engineering from Hanyang University, Korea (2005). He received the Ph.D. degree in computer science from the University of Texas at Dallas, Richardson, USA (2010). Currently, he is an assistant professor in the Department of Mathematics and Physics at North Carolina Central University, Durham, USA. His research interests include wireless networks, mobile computing, and approximation algorithm.

Hyunbum Kim is currently an assistant professor in the Department of Computer Science at University of North Carolina at Wilmington. He received his Ph.D. in computer science from the University of Texas at Dallas. His research interests include algorithm design/analysis/optimizations in various areas including wireless sensor networks, mobile computing, cyber physical systems, distributed computing and cyber security. He is a member of IEEE.

Deying Li received the M.S. in mathematics from Huazhong Normal University in 1988 and Ph.D. degrees in computer science from City University of Hong Kong in 2004. She is currently a professor in the Department of Computer Science from Renmin University of China. Her research includes wireless networks, mobile computing and algorithm design and analysis.



Sung-Sik Kwon received the Ph.D. degree in mathematics from University of North Carolina at Chapel Hill in 1996. He is currently an associate professor in the Department of Mathematics and Computer Science at North Carolina Central University. His research interests include numerical PDEs and nonlinear optimization.

Alade O. Tokuta received the B.S. and M.S. degrees in electrical engineering from Duke University, Durham, the E.E. degree in electrical engineering from Columbia University, NY, and Ph.D. degree in electrical engineering and computer science from University of Florida, Gainesville. Currently, he is a professor in the Department of Mathematics and Physics at the North Carolina Central University, Durham, USA. His research interests include robotics; computer image synthesis/vision; networking, and algorithm design.

Jorge A. Cobb is currently an associate professor in the Department of Computer Science at The University of Texas at Dallas. He received the B.S. in Computer Science (with highest honors) from The University of Texas at El Paso in 1987. He also received an M.A. in computer science and a Ph.D. degrees in computer science from The University of Texas at Austin. While in the Ph.D. program at The University of Texas at Austin, he was awarded Ph.D. scholarship. He has been a member of the technical program committee of many international conferences, including the IEEE International Conference on Network Protocols and the IEEE Local Computer Networks conference. His main research area is computer networking, with an emphasis on scheduling for quality of service guarantees, wireless networks, inter-domain routing, and self-stabilizing systems. He is a member of the IEEE.