CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2017; **29**:e4070
Published online 23 December 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.4070

SPECIAL ISSUE PAPER

Maximizing the lifetime of reinforced barriers in wireless sensor networks[‡]

Hyunbum Kim[1,*,†], Jorge A. Cobb[2] and Jalel Ben-Othman[3]

[1]*Department of Computer Science, University of North Carolina at Wilmington, Wilmington, NC 28403, USA*
[2]*Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080, USA*
[3]*Department of Computer Science, University of Paris 13, Villetaneuse, France*

SUMMARY

Recently, barrier-coverage in wireless sensor networks is a critical issue because it can be used for various applications (e.g., intrusion detection and border protection). Many existing works for barrier-coverage assume that an intruder penetrates through two opposite sides such as from top to bottom or from bottom to top and focus on constructing barriers to detect those penetrations. However, in many practical scenarios, it should be desirable to detect an intruder that enters the area of interest through any of its sides and passes through any other of its sides. In this paper, we introduce a new barrier-coverage problem whose goal is maximizing the network lifetime such that any penetration variation of the attacker is guaranteed to be detected. In order to solve the problem, we create a new type of sensor barriers, which is referred as *reinforced barriers*, that can sense any movement variation of the intruder. Also, we propose four different approaches to construct reinforced barriers from a given layout of sensors and we compare their relative performances for maximum number of reinforced barriers through extensive simulations by various scenarios. Copyright © 2016 John Wiley & Sons, Ltd.

Received 5 January 2016; Revised 28 April 2016; Accepted 12 November 2016

KEY WORDS: barrier-coverage; wireless sensor networks; maximum lifetime; scheduling; reinforced barriers

1. INTRODUCTION

During the last few decades, there has been an increased interest in Wireless Sensor Networks (WSN) because it can be used for a wide range of critical applications in industry, science, mobile infrastructure such as environmental monitoring, border protection, etc. Basically, a WSN is composed of sensor nodes, which is equipped with a sensing device, a wireless transceiver, battery. Sensor nodes have limited capabilities such as limited energy resource, limited computational ability. Sensor nodes can sense or monitor specific phenomenon of interest using the embedded sensing device and they communicate with each other to make joint decisions and to transmit their sensing information towards a base station [2–4].

In WSN, the coverage is a fundamental concern. The coverage provided by WSN is largely classified into two sub-areas, full coverage, and partial-coverage. The full coverage over a region of the interest guarantees that any event happening in the region at any moment is detected by the WSN [5–7]. On the other hand, although a partial coverage in WSN may miss some events in the

*Correspondence to: Hyunbum Kim, Department of Computer Science, University of North Carolina at Wilmington, Wilmington, NC 28403.
†E-mail: kimh@uncw.edu
‡A part of this paper has appeared in Proceedings of *IEEE International Conference on Networks (ICON 2013)*[1].

Copyright © 2016 John Wiley & Sons, Ltd.

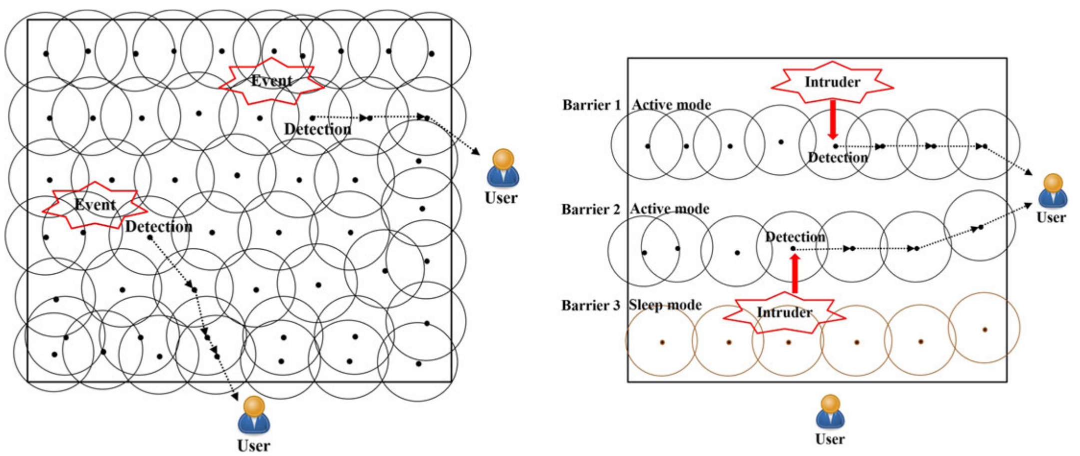
area of interest [8–10], it has an advantage that fewer sensors are required when compared with the full coverage.

In general, each sensor in WSN has two operation modes, sleep-mode and wake up (active) mode. To save own battery, a sensor can be under sleep mode and the sensor node can be wake up mode when necessary in the networks. The sleep-wake up schedule is helpful to maximize the network lifetime. Moreover, sensor nodes are randomly but densely deployed over the region of interest to increase network connectivity. Accordingly, it is highly likely that the same object or event is covered by more than one sensor node simultaneously. Such a high density property can be used to maximize the lifetime of WSN. For example, if multiple sensor nodes cover the same target or area, a sleep-wake up schedule can be applied. So, one can find a sleep-wake up schedule of the nodes and operate the sensors in disjoint sets to maximize the time to cover the target area. Through this sleep-wake up strategy, the total time to cover the target can be improved much longer than the case where all of the sensor nodes are active concurrently. But, the problem of optimal sleep-wake up schedule is NP-hard for full coverage even if every sensor has same energy resource.

Recently, a special form of partial-coverage, known as *barrier-coverage*, has attracted lots of interests by researchers because it can be used for various applications such as intrusion detection, battlefield surveillance, and border protection [11, 12]. A subset of sensors can provide a barrier-coverage over the area of interest if the sensors divide the area into two regions such that any moving objects from one region to another is detected by at least one sensor. Hence, when compared against full coverage, barrier-coverage provides significant savings in the number of sensor nodes which are required to support objectives of the barrier-coverage such as intrusion detection.

Figure 1(a) illustrates a case of full coverage in WSN. Each point represents a sensor and each circle represents the communication range of the sensor. As it can be seen, full coverage guarantees that any event will be detected and such a detection will be reported to the users. On the other hand, Figure 1(b) describes a barrier-coverage using sleep-wake up schedule. In the figure, Barriers 1 and 2 are in active-mode, and Barrier 3 is in sleep-mode to save its battery power. So, we can guarantee that any penetration by intruders is detected by at least two sensors: one sensor is on Barrier 1 and another sensor is on Barrier 2. This is known as 2-barrier coverage, because at least two barriers will detect the intruder, and the general problem is known as k-barrier coverage. Recently, Kumar et al. have shown that the sleep-wake up problem for k-barrier-coverage is polynomial time solvable[12] by developing an optimal sleep-wake up algorithm.

Let us consider practical scenarios which barrier-coverage can be applied. When an intruder tried to penetrate a region of interest, the intruder may not simply pass through two opposite sides, such as from top to bottom or from bottom to top. For example, we have two intersecting and perpendicular



(a) An example of full-coverage

(b) An example of barrier-coverage with sleep-wakeup schedule

Figure 1. Comparison of full coverage and barrier-coverage using sleep-wake up schedule.

Copyright © 2016 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

corridors in a building, and the square-shared region that is common to both of them. Then, the penetration of the attacker must be detected if it crosses the intersection of the two corridors.

Hence, we can consider various penetration patterns of the intruders: (i) passing through from top to bottom, (ii) going through from bottom to top, (iii) trespassing from left to right, (iv), passing through from right to left, (v) turn to the left after entering the region, and (vi) turn to the right after entering the area. Apparently, a system of barrier-coverage should detect those various penetration patterns of the intruder even though two more trespasses happens simultaneously. To guarantee detections of those various attacks, we simply may think a use of full coverage. However, this would result in a significantly larger number of required sensor nodes and a lower network lifetime. To the best of our knowledge, previous studies in barrier-coverage of WSN do not consider these various penetration types of the intruders.

Based on the earlier observations, we introduce a new barrier-coverage concept such that any penetration types of intruders is guaranteed to be detected by the system, and we also introduce the novel concept of a *reinforced barrier*, that implements this form of detection. We will also take advantage of redundancy to obtain multiple reinforced barriers because those multiple barriers can allow us to improve a network lifetime by using a sleep-wake up schedule. Thus, we formally define a problem whose objective is to maximize the network lifetime with construction of reinforced barriers. To solve the problem, we have developed four different approaches and have analyzed their performances with various simulation environments.

This paper is organized as follows. In the next section, we review related works for barrier-coverage of WSN. Then, in Section 3, we introduce our reinforced barrier-coverage as well as discuss the difference between a traditional barrier-coverage and our reinforced barrier-coverage. And we present a formal description of our reinforced barrier-coverage. Followed in Section 4 by four different approaches, we propose for the construction of reinforced barriers. Then, in Section 5, we analyze the performances of the proposed approaches through simulations. Finally, concluding remarks are given in Section 6.

2. RELATED WORK

As a fundamental problem for various applications in WSN, barrier-coverage has been studied widely. First, the notion of barrier-coverage was introduced by Gage [13] in the context of robotic sensors. In [11], Kumar et al. introduced the notion of k-barrier-coverage, which is a practical first work 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 in the region of interest. Also, they introduced weak and strong barrier-coverage in a belt region. In [14], Liu et al. considered the critical conditions for strong barrier-coverage in a strip region when sensor nodes are randomly deployed using a Poisson point distribution process and then they proposed an efficient distributed algorithm to construct barriers on long strip areas. Moreover, Li et al. [15] studied the weak-k-barrier coverage problem and derived a lower bound on the probability of weak k-barrier-coverage. Also, in [16], Saipulla et al. studied barrier-coverage in harsh environment which is difficult for human to reach. They considered barrier-coverage with airdropped wireless sensors using line-based deployments and derived a lower bound for the coverage.

In [12], Kumar et al. defined a sleep-wake up scheduling problem for k-barrier-cover of wireless sensors, whose objective is to extend the lifetime to protect a region of interest using a series of alternating barrier-covers. They developed optimal sleep-wake up scheduling algorithms for k-barrier-coverage, Stint and Prahari, assuming that each sensor has same energy resource in Stint and has different energy levels in Prahari, respectively. Later, Ban et al. developed a distributed algorithm for this problem providing low communication overhead and computation cost, and thus is appropriate for larger scale WSN [17].

Different from previous global barrier coverage, Chen et al. introduced an another barrier-coverage type, local barrier-coverage, which guarantees the detection of intruder whose trajectory is limited to a slice of the belt area [18]. To maximize the lifetime of the local barrier-coverage, they also proposed a sleep-wake up algorithm for this problem.

Copyright © 2016 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

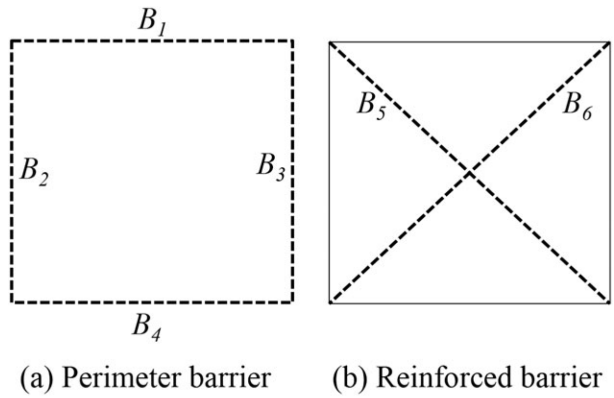


Figure 2. Difference between an existing perimeter barrier and our reinforced barrier.

Also, He et al. [19] proved a sub-optimality of line-based deployment if the length of the shortest line segment is larger than that of the shortest path. They also introduced the concept of a distance-continuous curve and developed an algorithm to provide an optimal deployment of sensor nodes if the deployment curve satisfies the distance-continuous condition. In [20, 21], the authors considered that intruders may have different moving speeds. Based on this, they evaluated the detection probability of temporary paths across the barrier of sensor nodes and derive the maximum possible speed of intruders for different intruders in different scenarios. Then, they defined the minimum weight ε-barrier problem about an efficient scheduling of sensors and proved that the problem is NP-hard.

On the other hand, many researchers have studied perimeter barriers, which can be considered as an another category of barrier. Basically, construction of perimeter barriers requires that we have to find sensor chains enclosing the region with the sensing areas of any two neighbor sensors overlapping with each other's to detect intruders from either entering its interior or exiting from it [22]. In [23], Bhattacharya et al. considered perimeter barrier-coverage on a simple polygon. They developed several algorithms to decide both the optimal locations and the movement scheme. In [24], Hung et al. focused on the perimeter coverage problem such that the perimeter of a big object needs to be monitored, but each sensor can only cover a single continuous portion of the perimeter. Authors proved that the perimeter coverage scheduling problem is NP-hard and then developed a maximum network lifetime scheduling algorithm.

From Figure 2, note that our reinforced coverage is different from perimeter coverage if we assume that the dotted line represents a barrier and the length of the side of the square is l. Perimeter barrier includes B_1, B_2, B_3, B_4 in Figure 2(a), and the perimeter barriers may detect various intrusion types of the attackers. With same square, our reinforced barrier with B_5 and B_6 in Figure 2(b) can also detect a variation of penetrations of intruders. However, while perimeter coverage usually uses movable sensor nodes to create the perimeter barrier, sensors nodes in our reinforced coverage are fixed after random deployment. Even if we intuitively compare reinforced barriers with perimeter barriers with respect to the total length of barriers in a given squared-shaped area, our reinforced barriers have the advantage that the total length of the barriers is $2(\sqrt{2} \cdot l)$ which is shorter than the total length of $4 \cdot l$ for perimeter coverage.

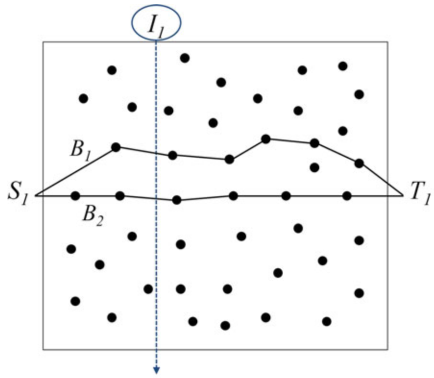
3. A NEW TYPE OF SENSOR BARRIER: REINFORCED BARRIER

In this section, we introduce a new barrier type, *reinforced barrier*. Also, we define our problem formally. Then, we describe how to create reinforced barriers to solve the problem.

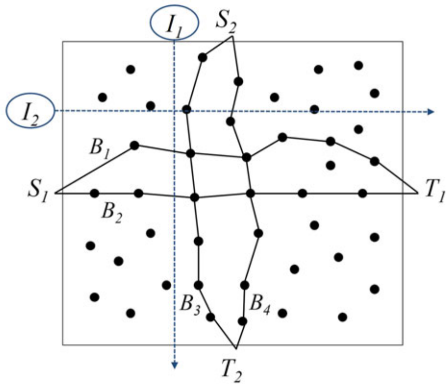
3.1. Reinforced barrier-coverage

Let us consider Figure 3, where a square-shaped sensor field is drawn, along with sensors being represented by dots. We assume that there is an edge (i.e., a solid line) between two sensors in the sensor field if the communication range of two sensors overlap. Only a subset of the possible

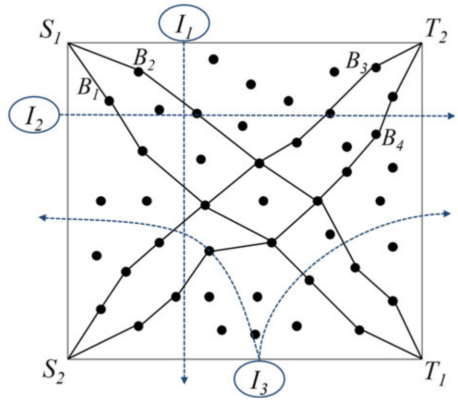
Copyright © 2016 John Wiley & Sons, Ltd.
DOI: 10.1002/cpe



(a) basic k-barriers



(b) Partially improved barriers



(c) Reinforced barriers

Figure 3. Possible concepts of barrier-coverage.

edges are represented in the Figure 3. Let us assume that we consider a virtual node, S_1, to the left border of the field, and another virtual node, T_1, to the right border of the field. So, if a path can be constructed from S_1 to T_1, then it is guaranteed that the intruders' penetrations, indicated by I_1, I_2, I_3, are detected by at least one sensor in the field. We call the path as a *sensor barrier*, and is the typical barrier which had been studied in the literature.

As you can see in Figure 3(a), two sensor barriers B_1 and B_2 are represented between S_1 and T_1. There exists a penetration direction of the intruder I_1, from top to bottom. Clearly, it takes two different advantages that we construct multiple barriers in the field. Firstly, both barriers could operate at the same time. This guarantees a certain degree of fault-tolerance, which is k-barriers. (i.e., At least k sensors can detect the penetration of the intruder if k barriers are activated simultaneously). Secondly, one barrier could be inactive (or asleep) while the other is active (or awake). When the first barrier B_1 is no longer available due to energy depletion of the sensors in the barrier; the second barrier B_2 is activated so that at least one sensor in B_2 can detect the penetration of the intruder I_1. Such a strategy doubles the lifetime of the network. Depending on the requirements of the application, both of these approaches can be used independently, or those approaches can be implemented together; for example, if we assume that there exist 10 barriers in the region of interest, we could make two barriers to be active at any time, which results in network lifetime of five sensor lifetimes finally. This property is known as k-barrier-coverage, where k is the number of concurrently active barriers. In this paper, we focus on the second approach. That is, one barrier is active at all times. And if we can find multiple barriers in the field, each barrier can be active or inactive alternately by sleep-wake up scheduling to maximize the network lifetime.

Copyright © 2016 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

Next, let us consider Figure 3(b). Also, let us assume that in addition to preventing intruders I_1 from crossing from top to bottom, we also would like to prevent intruders from crossing from left to right, which is represented by the direction of intruder I_2 in Figure 3(b). To detect those penetrations of intruders I_1 and I_2, we may generate additional barriers from top to bottom. For example, locate two additional virtual nodes, S_2 and T_2, and construct sensor barriers B_3 and B_4 between them. It follows that those intrusions by I_1 and I_2 are sensed by sensor nodes in B_1, B_2, B_3, B_4.

Those barriers in Figure 3(b) improves the security of the region when compared with barriers in 3(a). However, it is still possible to enter the region from one side and exit out another, which is shown by intruder I_3's penetration in Figure 3. To detect this type of movement variation, along with the ones discussed earlier, we introduce the notion of a *reinforced barrier*. To construct reinforced barrier, in particular, we consider placing four virtual nodes in opposite corners of given square-shaped field: S_1 and T_1 construct one pair of virtual nodes and another pair includes S_2 and T_2. For those virtual nodes, we construct sensor barriers between them in a crossed pattern, respectively. It follows that this formation of barriers, reinforced barrier, effectively removes every undetected penetrations across the region of interest. That is, our reinforced barriers guarantee that any penetration types by intruders I_1, I_2, I_3 can be detected by sensors on reinforced barriers.

3.2. Dependency property of reinforced barrier-coverage

Our reinforced barrier has an important property, dependency. Basically, reinforced barrier is constructed for two pairs of virtual sources and destinations. That is, some barriers can be generated between S_1 and T_1. Also, some barriers can be found between S_2 and T_2. So, we may have the interaction between barriers that cross each other. It means that some sensors in barriers for S_1 and T_1 also can be used in barriers for S_2 and T_2. It follows that there may exist common sensors in reinforced barriers.

Let us consider Figure 4(a) as an example. In the figure, three barriers, B_1, B_2, B_3, are constructed from S_1 to T_1. Also, from S_2 to T_2, we have three additional barriers, B_4, B_5, B_6. Then, one would expect that the network lifetime is three, by pairing up barriers B_1 with B_4, B_2 with B_5, and B_3 with B_6. But, let us assume that some barriers have nodes in common. In Figure 4(a), those common sensors are denoted by small circles, a, b, c, d, respectively. Therefore, B_1 has a node in common with each of B_4 (with node a), B_5 (with node b), and B_6 (with node c). And B_3 also has a node in common with B_4 (with node d). This affects the lifetime as follows.

Suppose that B_1 is paired with B_4, and it is activated as Figure 4(b). As the lifetime of B_1 expires, so will all of nodes a, b, and c. This will also eliminate barriers B_5 and B_6, because they are now missing nodes b and c. Hence, the network lifetime is just one sensor lifetime. On the other hand, let us assume that we pair up B_3 with B_4, and B_2 with B_5, respectively, as Figure 4(c). This pairing increases the network lifetime to two sensor lifetimes. As we could check this sample scenario in Figure 4, because reinforced barrier-coverage has the dependency property, finding a scheduling with maximum network lifetime of reinforced barrier-coverage is not simple, which causes more complexity than basic barriers.

Because of this complexity, we cannot directly apply maximum flow algorithms to search for independent paths (barriers) of sensor nodes, as performed in [12]. Instead, we create reinforced barriers via four algorithms which we present in Section 4.

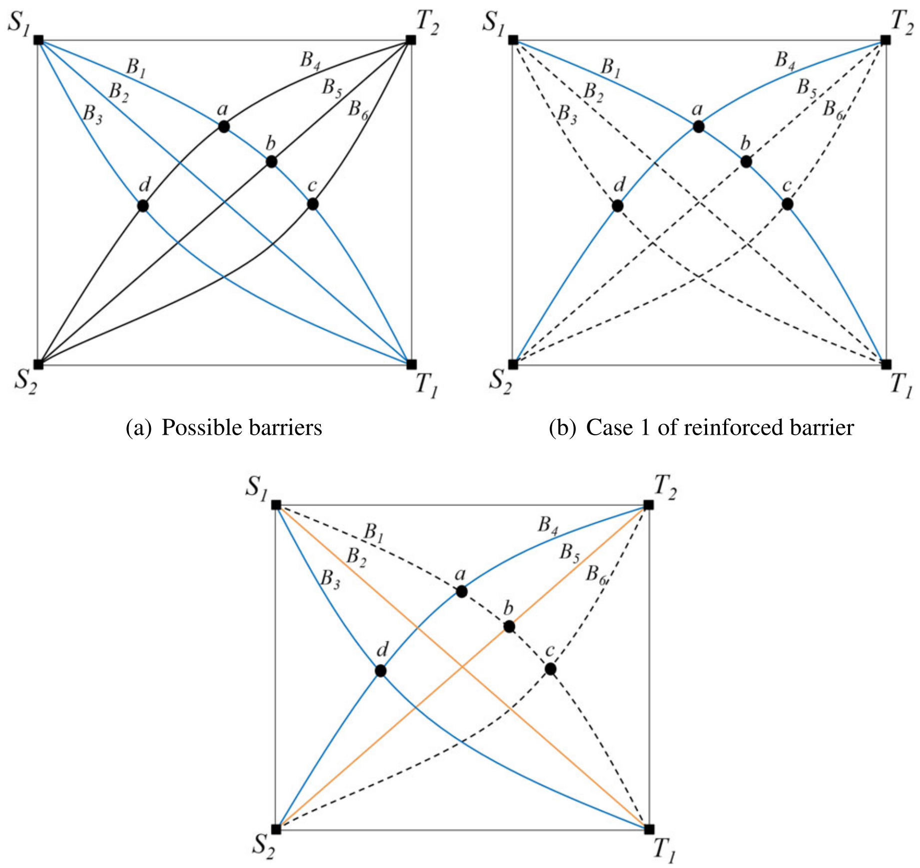
3.3. Problem definition

In this paper, we consider a square-shaped area, A. And a set of sensors, H, are randomly deployed in the area A. After the random deployment, each sensor is static. And each sensor can be either in sleep mode, in which case it uses a negligible amount of battery, or in active mode, in which it senses events within own sensing range g. Once a sensor is set into active mode, it remains in this mode until it is depletes own energy. We assume that all sensor nodes have an equal amount of battery, and thus, an equal lifetime. For simplicity, we also assume each sensor has the same sensing range and communication range denoted by g.

For our reinforced barrier-coverage, we locate four virtual sensors (S_1, T_1), (S_2, T_2), one at each corner of the area A, as shown in Figure 4. Assume that these four sensors have unlimited lifetime and we simplify the definitions that follow.

Copyright © 2016 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe



(a) Possible barriers (b) Case 1 of reinforced barrier

(c) Case 2 of reinforced barrier

Figure 4. Dependency property in reinforced barrier-coverage.

Two sensors, h_1 and h_2, can be considered as *neighbors* if the *euclidian distance* between h_1 and h_2, $Euc(h_1, h_2)$, is at most $2 \cdot g$. A *path* of sensor nodes is a sequence of sensors, $h_1, h_2, \ldots, h_n$, where:

- h_i and h_{i+1} are neighbors, $1 \leq i \leq n-1$.
- $h_1 = S_1$ and $h_n = T_1$, or, $h_1 = S_2$ and $h_n = T_2$.

We are now ready to describe the formal definitions of a reinforced barrier, its lifetime and the proposed problem.

Definition 1 (Reinforced Barriers)
A *reinforced barrier*, or r–barrier, consists of two paths of sensor nodes: one path from S_1 to T_1 and another from S_2 to T_2. It is not necessary condition that these two paths are disjoint.

Definition 2 (Lifetime of Reinforced Barriers)
A *collection* C of r–barriers is composed of a set of r–barriers, where for each $p \in C$ and $q \in C$, p and q are disjoint, that is, they have no sensors in common. The *lifetime* of a collection C is simply considered as its cardinality, $|C|$.

Definition 3 (MaxLRB)
Given a set of wireless sensors H deployed over a square area A, *the maximum lifetime reinforced barrier-coverage (MaxLRB) problem* is to find an r–barrier collection C_{max} with maximum lifetime.

Copyright © 2016 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

4. *MAXLRB* HEURISTICS

In this section, we describe our four different heuristics to seek a collection C of r–barriers. To solve the $MaxLRB$ problem, we search for node-disjoint paths from S_1 to T_1, and also node-disjoint paths from S_2 to T_2, and then combine them together to generate r–barriers.

We use an approach which is similar to Stint algorithm by Kumar et al. [12] to find maximum number of disjoint paths. Firstly, we explain our adaptation of Stint to search for independent paths (node-disjoint paths) from S_1 to T_1. Similarly, we also find node-disjoint paths from S_2 to T_2. Then, we present our proposed heuristics for combining these paths to construct r–barriers.

To find the largest number of independent paths from S_1 to T_1, we implement the following two steps.

Step 1.

Create a *flow graph* $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$, as follows.

- For each sensor $h \in H$, consider two vertices in $V(\mathcal{G})$, h_{in} and h_{out}.
- For each sensor $h \in H$, add a directed edge (h_{in}, h_{out}) to $E(\mathcal{G})$.
- For every pair of sensor nodes u and v in H that are neighbors, add the following two directed edges to $E(\mathcal{G})$: (u_{out}, v_{in}), (v_{out}, u_{in}).
- Add vertices S_1 and T_1 to $V(\mathcal{G})$.
- For each neighbor h of S_1, add the edge (S_1, h_{in}) to $V(\mathcal{G})$.
- For each neighbor h of T_1, add the edge (h_{out}, T_1) to $V(\mathcal{G})$.

Step 2.

Assign a capacity of 1 to each edge in the flow graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$, and implement a maximum flow algorithm, such as Edmonds–Karp algorithm[25], between S_1 and T_1. Edges with a flow of 1 will form independent paths from S_1 to T_1. All other edges will have a flow of 0.

Figure 5 shows maximum number of independent paths after implementing Step 1 and Step 2 based on Stint algorithm in [12]. (i.e., we found five independent paths in the figure). In Figure 5, S_1 is a source and T_1 is a destination. We represented a sensor node as a small circle. Then, each path, B_1, B_2, B_3, B_4, B_5, can be an independent path (or a node-disjoint path) between S_1 and T_1. It follows that one path also can be considered as one barrier. For example, if we consider k-barrier coverage with $k = 1$ in Figure 5, the network lifetime with barrier-coverage is five because each barrier can become active alternately after one barrier expires due to the depletion of battery.

Basically, the intuition behind the earlier two steps is as follows [12]. It is well known that if the edges in the graph have an integer capacity, then the maximum flow will also have an integer flow assignment to each edge. Thus, a flow of 0 or 1 will be assigned to each edge. Additionally, because of the single edge between h_{in} and h_{out}, each node, other than S_1 or S_2, can only take part in a single path.

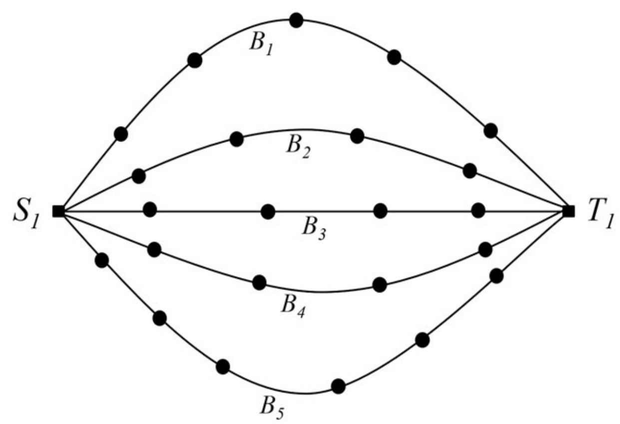


Figure 5. Maximum number of independent paths (or node-disjoint paths) between source to destination.

Based on the earlier strategy to search for the maximum number of independent paths, we develop our heuristics to solve the *MaxLRB* problem.

4.1. Approach 1: independent-paths

As we have discussed in Section 3, sensor nodes cannot belong to more than one r–barrier in a collection C of r–barriers. One possible strategy to ensure this is the case is that all paths found from S_1 to T_1 do not have any sensors in common with paths from S_2 to T_2. Using this way, enabling an r–barrier has no effect on the future performance of other r–barriers.

To perform this, we can consider that one is tempted to first find all the independent paths from S_1 to T_1 and we remove these paths. Then, we search for all the independent paths from S_2 to T_2. This strategy will likely lead to allow no paths from S_2 to T_2 unfortunately because many sensors will be used in the first step.

So, we select an iterative approach, in which one r–barrier constructed completely is found at a time. The steps are performed as follows.

- Find the maximum number of node-disjoint paths from S_1 to T_1 using current available sensor nodes. Furthermore, search for the maximum number of node-disjoint paths from S_2 to T_2.
- An r–barrier is formed by selecting one path from S_1 to T_1 and another path from S_2 to T_2 such that these two paths are node-disjoint.
- From the field, remove the sensors which are included at the above r–barrier.

The earlier steps are iterated until we cannot obtain any additional pair of node-disjoint paths. The pseudocode is presented in Algorithm 1 in more detail, which we have named the *independent-path*.

Algorithm 1 *independent-path*
Inputs: H, A, g, Output: C

set r–barrier collection: $C \leftarrow \emptyset$;
set unselected sensor nodes: $H' \leftarrow H$;
while $H' \neq \emptyset$ **do**
    create a flow graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ using A, H', and g;
    let P_1 be the set of node-disjoint paths from S_1 to T_1;
    let P_2 be the set of node-disjoint paths from S_2 to T_2;
    search for a pair of node-disjoint paths (p_1, p_2), where $p_1 \in P_1$ and $p_2 \in P_2$;
    if (p_1, p_2) exists **then**
        $C \leftarrow C \bigcup (p_1, p_2)$;
        $H' \leftarrow H' - (nodes(p_1) \bigcup nodes(p_2))$;
    else
        break;
    end if
end while
return C

4.2. Approach 2: shared-paths

Now, we move to our second approach, which is referred as *shared-path*. Differently from *independent-path*, it takes an advantage that some sensors can be reused within the same r–barrier, provided that they did not interfere with another r–barrier. Hence, the path from S_1 to T_1 is able to share sensors with the path from S_2 to T_2, provided that those two paths belong to the same r–barrier, and do not take part in any other r–barrier. Basically, the intuition behind this approach is that shared sensors are serving a double purpose, by joining in each of the two paths, which makes a more effective use of the available nodes.

Based on this idea, we iterate adding one r–barrier at at time until we cannot obtain any additional r–barrier. So, the below steps are performed in each iteration.

Copyright © 2016 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

- Search for the maximum number of node-disjoint paths from S_1 to T_1. Also, find the maximum number of node-disjoint paths from S_2 to T_2.
- A r–barrier is created by choosing a path from S_1 to T_1 and another path from S_2 to T_2 on condition that those two paths share at least one sensor.
- Remove from the field the sensors comprised by the above r–barrier.

The pseudocode of approach 2 is very similar to that of Algorithm 1. So, it is possible that we simply replace line 7 in Algorithm 1 by the following line:

search for a pair of paths (p_1, p_2), where $p_1 \in P_1$, $p_2 \in P_2$, where p_1 and p_2 have at least one sensor in common.

4.3. *Approach 3: combined-paths*

As the third approach to solve *MaxLRB* problem, we consider a combination of approach 1 and 2. We refer the approach 3 as *combined-paths*.

Basically, *combined-paths* adds one r–barrier at at time as approach 1 and 2. But, it first adds as many barriers as possible whose two paths have no common sensors. If we cannot continue this process, then r–barriers are added whose two paths have sensors in common. We describe the pseudocode of *combined-paths* in Algorithm 2 in more detail.

Algorithm 2 ***combined-paths***

Inputs: U, A, g, Output: C

```
set r–barrier collection: C ← ∅;
set unselected sensor nodes: U′ ← U;
while U′ ≠ ∅ do
    create a flow graph 𝒢 = (V(𝒢), E(𝒢)) using A, U′, and g;
    let P1 be the set of node-disjoint paths from s1 to d1;
    let P2 be the set of node-disjoint paths from s2 to d2;
    find a pair of node-disjoint paths (p1, p2), where p1 ∈ P1 and p2 ∈ P2;
    if (p1, p2) exists then
        C ← C ⋃ (p1, p2);
        U′ ← U′ − (nodes(p1) ⋃ nodes(p2));
    else
        find a pair of paths (p1, p2), where p1 ∈ P1 and p2 ∈ P2 that share at least one sensor;
        if (p1, p2) exists then
            C ← C ⋃ (p1, p2);
            U′ ← U′ − (nodes(p1) ⋃ nodes(p2));
        else
            break;
        end if
    end if
end while
return C
```

4.4. *Approach 4: minimum-intervention-paths*

Our last method is referred to as *minimum-intervention-paths*. Similar to our previous approaches, it initially looks for many possible paths (i.e., barriers): one path set is between S_1 to T_1 and another path set is between S_2 and T_2.

The basic idea of *minimum-intervention-paths* is to provide many possible barriers for the *next* iteration of the algorithm. Recall the dependency property of r–barriers that we showed in Figure 4(a). As seen from Figure 4, if we select barrier B_1 between S_1 and T_1, it has the maximum effect or *maximum intervention* with other barriers B_4, B_5, B_6. If we select B_1 as a part of one r–barrier, we may not find additional r–barriers after the selection because sensor nodes a, b, c are

Copyright © 2016 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

removed once B_1 is chosen. So, at the next search for an r–barrier, we may find very few, if any, possible barriers between S_2 and T_2 in Figure 4(a).

By considering the property of dependency in reinforced barrier-coverage, we construct an r–barrier by selecting a path from P_1 that has the least number of paths in P_2 with whom it shares sensors. Similarly, we choose a path in P_2 that interferes the least with paths in P_1.

In summary, the steps below are iterated by adding one r–barrier at a time until we cannot obtain any additional r–barrier.

- Find the maximum number of node-disjoint paths from S_1 to T_1. The paths are referred to as set P_1. Moreover, obtain the maximum number of node-disjoint paths from S_2 to T_2. These paths are referred to as set P_2.
- From P_1, choose one path p_1 which has the minimum effect or minimum intervention on paths in P_2. That is, the chosen path has the smallest number of sensors which are used as parts of P_2.
- From P_2, select one path p_2 which has the minimum effect or minimum intervention on paths in P_1.
- p_1 and p_2 form the desired r–barrier.
- Remove from the field the sensors taking part in the above r–barrier.

In Section 5, we will show that this *minimum-intervention-paths* outperforms the other approaches. The pseudocode is described in Algorithm 3 in more detail.

Algorithm 3 ***minimum-intervention-paths***
Inputs: U, A, g, Output: C

set r–barrier collection: $C \leftarrow \emptyset$;
set unselected sensor nodes: $U' \leftarrow U$;
while $U' \neq \emptyset$ **do**
    create a flow graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ using A, U', and g;
    let P_1 be the set of node-disjoint paths from s_1 to d_1;
    let P_2 be the set of node-disjoint paths from s_2 to d_2;
    find a p_1 which has the minimum common number of sensors with P_2, where $p_1 \in P_1$;
    find a p_2 which has the minimum common number of sensors with P_1, where $p_2 \in P_2$;
    if p_1 and p_2 exist **then**
        $C \leftarrow C \bigcup (p_1, p_2)$;
        $U' \leftarrow U' - (nodes(p_1) \bigcup nodes(p_2))$;
    else
        break;
    end if
end while
return C

5. EXPERIMENTAL EVALUATION

In this section, we analyze and discuss the performance of the proposed four approaches, *independent-path*, *shared-path*, *combined-paths*, and *minimum-intervention-paths*, which we have described in Section 4.

We simulated our various experiments in square areas, $400 \times 400m^2$, $500 \times 500m^2$, $600 \times 600m^2$ where H sensor nodes are randomly deployed in the region initially. Also, we implemented the proposed approaches with different shaped area of size $500 \times 500m^2$, $400 \times 500m^2$, $300 \times 500m^2$, respectively.

For the used simulation setting and environment, the code of the proposed approach using language C^{++} has been implemented at the server with two components. Each component consists of Dual Core AMD Opteron(Tm) processor 285, CPU MHz - 2592.690, cache size - 1024 KB,

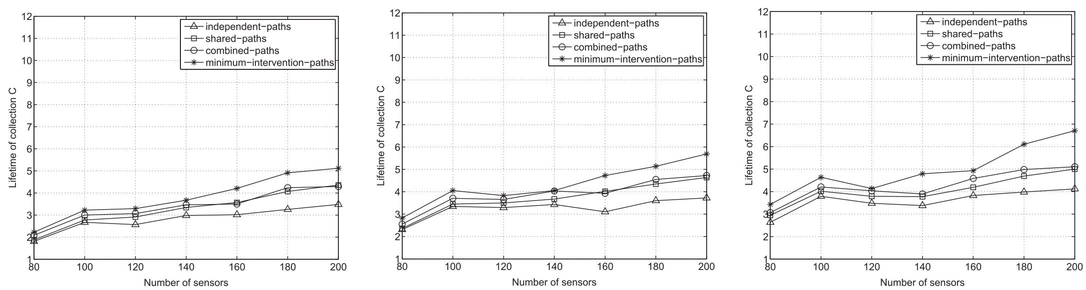
Copyright © 2016 John Wiley & Sons, Ltd.

Operating System - CentOS Linux. Through the system based on the code, we got numerical results of the lifetime value C.

Each numerical result value of the simulation represents the average result of 100 different graphs. In our simulations, we considered that the number of sensors ranges from 80 to 200 and we also used that the transmission radius of each sensor is ranging from 70 to 100 in our experiments. Interestingly, when we have checked the performances as a whole, *minimum-intervention-paths* outperforms other approaches: *independent-path*, *shared-path*, and *combined-paths*. Now, we analyze the simulation results by different scenarios and different field sizes.

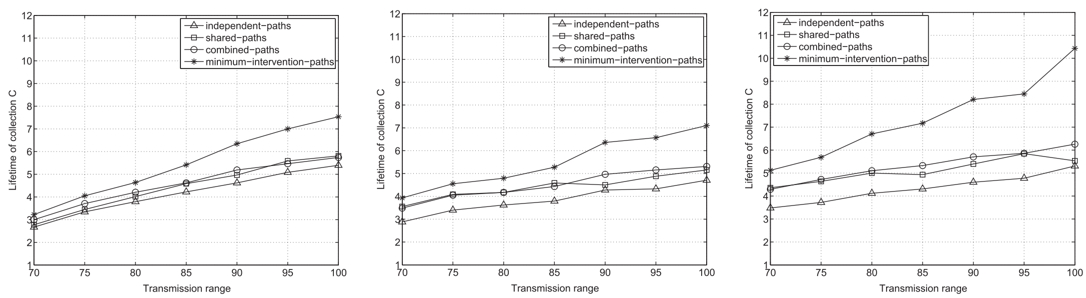
As the first performance evaluation in our contribution, we compare *independent-path*, *shared-path*, *combined-paths*, and *minimum-intervention-paths* with area size $500 \times 500m^2$ for lifetime of a collection C, respectively. As we defined a lifetime of the collection C or its cardinality $|C|$ in Section 3, $|C|$ can be considered as a total network lifetime maintaining our reinforced barrier-coverage. Hence, maximizing value of $|C|$ is equivalent to maximizing lifetime of r-barriers in the field. Let us consider Figure 6. It shows results for our four different approaches by different radius $g = 70, 75$, and 80, respectively. Through this simulation scenario, we can verify that *minimum-intervention-paths* shows better performance than *independent-path*, *shared-path*, and *combined-paths*. Furthermore, as seen from Figure 6, lifetime of a collection C increases for all approaches as the number of sensor nodes increases. On the other hand, we consider our approaches by different number of sensor nodes. Figure 7 represents results by the proposed approaches by different total number of sensor nodes $n = 100, 150$, and 200, respectively. As you can see in the Figure 7, as a transmission radius of sensor increases, a lifetime of a collection C increases for every approach. Also, we have checked that as the number of sensor increases, *minimum-intervention-paths* shows better performance than other approaches as Figure 7.

In our second group of experiments, we have implemented four approaches with another field size $400 \times 500m^2$. At this simulation group, we considered that our approaches are implemented both by different radius of sensor $g = 70, 75$, and 80 and by different number of sensors $n = 100, 150$, and 200. By Figure 8, we can check that *minimum-intervention-paths* outperforms other approaches



(a) $g = 70$ (b) $g = 75$ (c) $g = 80$

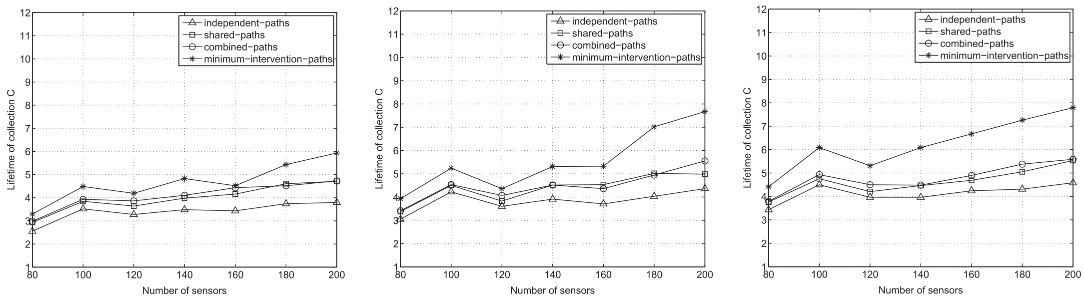
Figure 6. Comparison for lifetime of collection C for different radii by four approaches in 500×500 region.



(a) $n = 100$ (b) $n = 150$ (c) $n = 200$

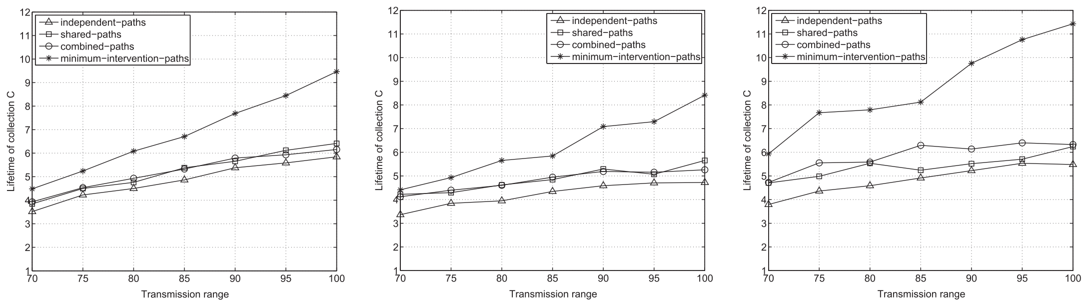
Figure 7. Comparison for lifetime of collection C for different number of sensors by four approaches in 500×500 region.

Copyright © 2016 John Wiley & Sons, Ltd.



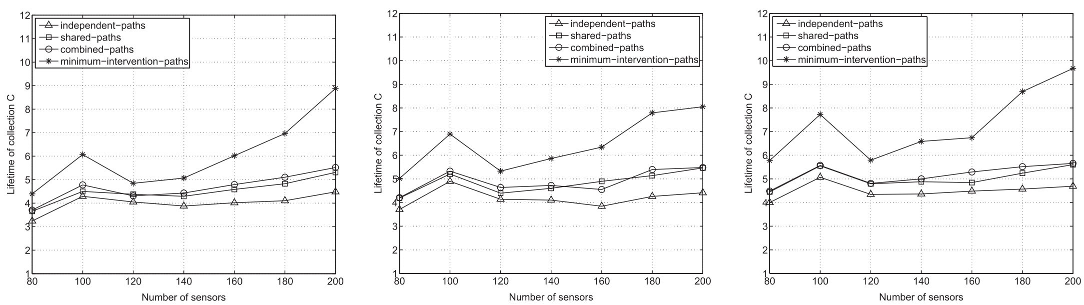
(a) $g = 70$ (b) $g = 75$ (c) $g = 80$

Figure 8. Comparison for lifetime of collection C for different radii by four approaches in 400×500 region.



(a) $n = 100$ (b) $n = 150$ (c) $n = 200$

Figure 9. Comparison for lifetime of collection C for different number of sensors by four approaches in 400×500 region.



(a) $g = 70$ (b) $g = 75$ (c) $g = 80$

Figure 10. Comparison for lifetime of collection C for different radii by four approaches in 300×500 region.

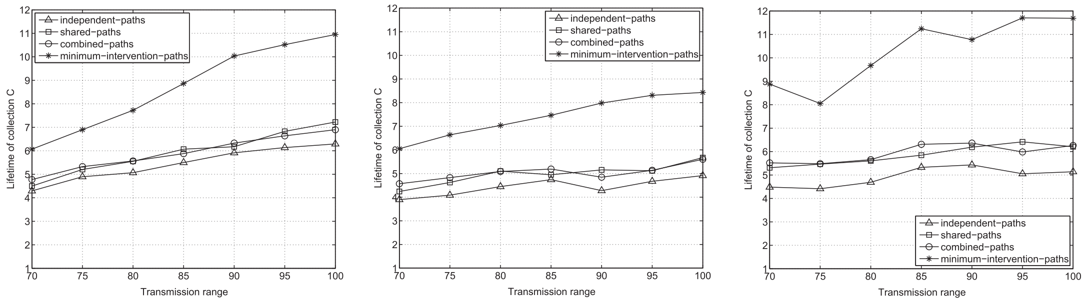
by different radius. And, by Figure 9, it is verified that *minimum-intervention-paths* shows better results of a collection C than other approaches, too.

At our third simulations, four approaches have been simulated with another area size $300 \times 500 m^2$. The area $300 \times 500 m^2$ will provide bigger density of sensors than $500 \times 500 m^2$. Similar with the first and second group of simulations, we considered our simulations by different radius of sensor $g = 70, 75$, and 80 and by different number of sensors $n = 100, 150$, and 200. As it can be seen from Figure 10 and 11, *minimum-intervention-paths* outperforms other approaches for both scenarios. Furthermore, through the third group of experiments, we can conclude that as the density of sensor nodes increases, *minimum-intervention-paths* shows significantly better performance than other approaches, *independent-path*, *shared-path*, and *combined-paths*.

At our fourth experiments, we have simulated the proposed approaches in square units with the number of sensors $n = 100$ to analyze performances of the four approaches for lifetime of C. The used square-shaped areas are $400 \times 400\ m^2$, $500 \times 500\ m^2$, and $600 \times 600\ m^2$, respectively.

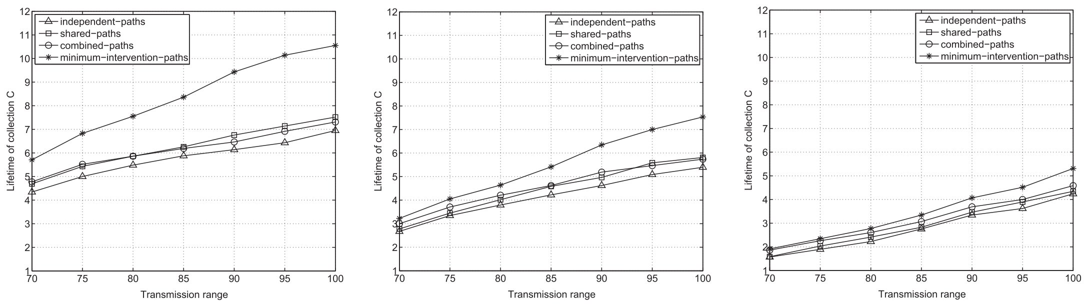
Copyright © 2016 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe



(a) $n = 100$ (b) $n = 150$ (c) $n = 200$

Figure 11. Comparison for lifetime of collection C for different number of sensors by four approaches in 300×500 region.



(a) 400×400 area (b) 500×500 area (c) 600×600 area

Figure 12. Comparison for lifetime of collection C for different square-shaped areas by four approaches with n = 100.

The simulations show that *minimum-intervention-paths* outperforms other approaches, *independent-path*, *shared-path*, and *combined-paths* in those areas. Also, through these simulations as Figure 12, we found that as the density of sensors increases, the value of C also increases. It follows that with n = 100, the smaller area $500 \times 500m^2$ returns a bigger value of lifetime C than the area $600 \times 600m^2$. And the smaller area $400 \times 400m^2$ shows a bigger value of C than the area $500 \times 500m^2$ with the same number of sensors n = 100.

Based on the entire simulations with various sensor numbers, transmission ranges, square-shaped areas, we found that *minimum-intervention-paths* outperforms other approaches as a whole. So, *minimum-intervention-paths* shows the best result of lifetime C. *Combined-paths* and *shared-path* is ranked as the second one and third one, respectively. *Independent-path* shows the worst performance of lifetime C. Also, as the number of sensors increases, the proposed approaches returns bigger value of lifetime C. Bigger transmission ranges also returns bigger value of lifetime C, too. It follows that as density of sensor node increases, the proposed approaches returns the increased lifetime C because the bigger density basically allows the network to have more independent paths between virtual nodes S_1 and T_1 as well as S_2 and T_2.

6. CONCLUDING REMARKS

This paper introduced a new barrier-coverage type, reinforced barrier-coverage, which is able to detect any penetration variation of the intruder in the square-shaped area. Then, we formally defined a new barrier-coverage problem whose objective is to maximize the network lifetime with reinforced barriers, r-barriers. By finding the maximum number of r–barriers in the network, we consider the maximum network lifetime with r-barriers by using a sleep-wake up schedule. To solve the problem, we have proposed four different approaches in order to maximize the lifetime for r-barriers, and we then analyzed their performances through extensive simulations. As a future work, we plan to study

Copyright © 2016 John Wiley & Sons, Ltd.

fault-tolerant r–barriers, which can maintain r–barriers when some sensors in current r–barriers are depleted by excessive consumption of battery.

REFERENCES

1. Kim H, Cobb JA. Maximum lifetime of reinforced barrier-coverage in Wireless Sensor Networks. *Proceedings of the 19th IEEE International Conference on Networks (ICON)*, Singapore, 2013; 1–6.
2. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. *Computer Networks* 2002; **38**(4):393–422.
3. Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey. *Computer Networks* 2008; **52**(12):2292–2330.
4. Stankovic JA. Wireless sensor networks. *IEEE Computer* 2008; **41**(10):92–95.
5. Huang C, Tseng Y. The coverage problem in a wireless sensor network. *Proceedings of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, San Diego, 2003; 115–121.
6. Zhang H, Hou J. On deriving the upper bound of α-lifetime for large sensor networks. *Proceedings of the ACM International Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc)*, Tokyo, 2004; 121–132.
7. Cardei M, Thai M, Li Y, Wu W. Energy-efficient target coverage in wireless sensor networks. *IEEE Annual International Conference on Computer Communications (INFOCOM)*, Miami, 2005; 1976–1984.
8. Gao S, Wang X, Li Y. p-Percent coverage schedule in wireless sensor networks. *The International Conference on Computer Communications and Networks (ICCCN)*, St. Thomas, 2008; 1–6.
9. Vu CT, Chen G, Zhao Y, Li Y. A universal framework for partial coverage in wireless sensor networks. *IEEE International Performance Computing and Communications Conference (IPCCC)*, Phoenix, 2009; 1–8.
10. Li Y, Vu C, Ai C, Chen G, Zhao Y. Transforming complete coverage algorithms to partial coverage algorithms for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 2011; **22**(4):695–703.
11. Kumar S, Lai TH, Arora A. Barrier coverage with wireless sensors. *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Cologne, 2005; 284–298.
12. Kumar S, Lai TH, Posner ME, Sinha P. Maximizing the lifetime of a barrier of wireless sensors. *IEEE Transactions on Mobile Computing (TMC)*, Vol. 9, 2010.
13. Gage D. Command control for many-robot systems. *Unmanned Systems Magazine* 1992; 10 (4): 28–34.
14. Liu B, Dousse O, Wang J, Saipulla A. Strong barrier coverage of wireless sensor networks. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Hong Kong, 2008; 411-420.
15. Li L, Zhang B, Shen X, Zheng J, Yao Z. A study on the weak barrier coverage problem in wireless sensor networks. *Computer Networks* 2011; **55**(3):711–721.
16. Saipulla A, Liu B, Wang J. Barrier Coverage with air-dropped wireless sensors. *Proceedings of the 28th IEEE Military Communications Conference (MILCOM)*, Baltimore, 2008; 1–7.
17. Ban D, Feng Q, Han G, Yang W, Jiang J, Dou W. Distributed scheduling algorithm for barrier coverage in wireless sensor networks. *Proceedings of the 2011 Third International Conference on Communications and Mobile Computing (CMC)*, IEEE Computer Society, Washington, DC, USA, 2011; 481–484.
18. Chen A, Kumar S, Lai TH. Local barrier coverage in wireless sensor networks. *IEEE Transactions on Mobile Computing* 2009; 9(4): 491–504.
19. He S, Gong X, Zhang J, Chen J, Sun Y. Curve-based deployment for barrier coverage in wireless sensor networks. *IEEE Transactions on Wireless Communications* 2014; **13**(2):724–735.
20. Li J, Chen J, Lai TH. Energy-efficient intrusion detection with a barrier of probabilistic sensors. *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, Orlando, Fla, USA, 2012; 118–126.
21. Chen J, Li J, Lai TH. Energy-efficient intrusion detection with a barrier of probabilistic sensors: global and local. *IEEE Transactions on Wireless Communications* 2013; **12**(9):4742–4755.
22. Wang C, Wang B, Liu W. Movement strategies for improving barrier coverage in wireless sensor networks: A survey. in *Proceedings of the 13th International Conference on Communication Technology (ICCT)*, Jinan, 2011; 938–943.
23. Bhattacharya B, Burmester M, Hu Y, Kranakis E, Shi Q, Wiese A. Optimal movement of mobile sensors for barrier coverage of a planar region. *Theoretical Computer Science (Elsevier)* 2009; **410**:5515–5528.
24. Hung K, Lui K. Perimeter coverage scheduling in wireless sensor networks using sensors with a single continuous cover range. *EURASIP Journal on Wireless Communications and Networking* 2010; volume 2010, Article ID 926075; 17 pages.
25. Edmonds J, Karp RM. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM* 1972; **19**(2):248–264.

Copyright © 2016 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe