Maximum Lifetime of Reinforced Barrier-Coverage in Wireless Sensor Networks

Hyunbum Kim
Department of Computer Science
Bethune-Cookman University
Daytona Beach, Florida 32114
kimh@cookman.edu

Jorge A. Cobb
Department of Computer Science
The University of Texas at Dallas
Richardson, Texas 75080
cobb@utdallas.edu

***Abstract*—A subset of sensors in wireless sensor networks 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 another is guaranteed to be detected by a sensor. In many practical scenarios, it may be desirable to detect an intruder that enters the region through any of its sides and exits through any other of its sides. That is, not only detect top-down movement, but also side-to-side, and even turn from one side to another. In this paper, we introduce a new barrier-coverage problem whose objective is to maximize the network lifetime such that any penetration of the intruder is detected. To solve the problem, we create a new form of sensor barriers, which we refer to as *reinforced barriers*, which can detect any movement variation of the attacker. Also, we propose three approaches to obtain these barriers from a given layout of sensor nodes, and we then compare their relative performances through extensive simulations.**

I. Introduction

A Wireless Sensor Networks (WSN) is an appropriate 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 towards a base station [1], [2].

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 [3]–[5]. In contrast, a WSN providing partial-coverage may miss some event in an area of interest [6]–[8].

Generally, a WSN is 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 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 operates the nodes one by one to maximize the time to cover the target. Obviously, 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.

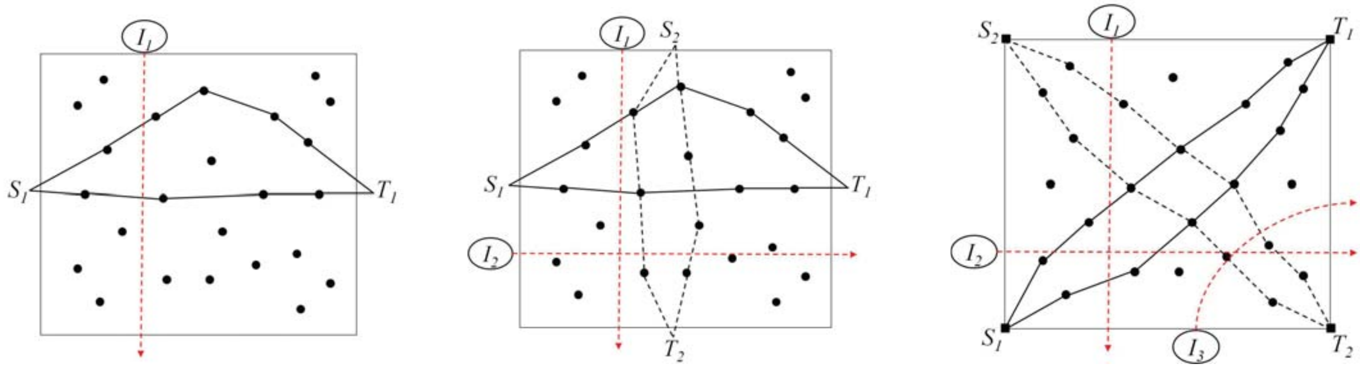
Recently, a form of partial-coverage, known as *barrier-coverage*, has attracted lots of attention because it is appropriate for many important applications, such as intrusion detection [9]–[11]. A subset of sensors provides barrier-coverage over an area of interest if the sensors are dividing the area into two regions such that any object moving from one region to another is guaranteed to be detected by at least one sensor. As a result, when compared against full-coverage, barrier-coverage provides significant savings in the number of sensors that are necessary to provide intrusion detection.

In many practical scenarios, when an intruder tries to penetrate an area 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, consider two intersecting and perpendicular corridors in a building, and the square-shaped area that is common to both of them. The movement of an intruder must be detected if it crosses the intersection of the two corridors. There are various penetration types of the intruder: (1) passing through from top to bottom, (2) passing through from bottom to top, (3) passing through from left to right, (4) passing through from right to left, (5) turn to the left after entering the area, (6) turn to the right after entering the area. Clearly, a system should detect various penetration types of the intruder even though two or more penetrations occur simultaneously. Although a simple use of full-coverage may allow these forms of detection, this would result in a significantly larger number of required sensors. Also, to the best of our knowledge, previous studies in barrier-coverage do not consider these trespass cases of an intruder.

Based on the above motivations, we introduce a new barrier-coverage concept such that any type of penetration is detected by the system, and we formally define the concept of a *reinforced barrier*, that implements this form of detection. We will also take advantage of redundancy in the network to obtain multiple reinforced-barriers and thus improve network lifetime through a sleep-wakeup schedule.

This paper is organized as follows. In the next section, we

This research was supported in part by an NSF Division of Graduate Education grant number 0742477.



(a) Basic barriers (b) Partially improved barriers (c) Reinforced barriers

Fig. 1. Possible formation of barriers.

discuss the difference between a traditional barrier-coverage and our reinforced barrier-coverage. We then review related works for barrier-coverage of WSN in Section III. We present a more formal description of reinforced-barriers in Section IV, followed in Section V by three approaches we propose for the construction of these barriers. Then, in Section VI, we analyze the performances of the three approaches through simulations. Finally, we conclude this paper in Section VII.

II. A New Type of Sensor Barrier

Consider Fig. 1(a), where a square sensor field is depicted, along with sensors being represented by dots. If the sensing range of two sensors overlap, then we draw an edge between the sensors. Only a subset of the possible edges are shown in Fig. 1(a). In particular, assume there is a virtual node, S_1, to the left of the field, and another virtual node, T_1, to the right of the field. If we can construct a path from S_1 to T_1, then an intruder cannot cross the area in the direction indicated by I_1, without being detected by at least one sensor. Such a path is referred to as a *sensor barrier*, and is the typical sensor barrier found in the literature.

Two barriers are illustrated in Fig. 1(a). Multiple barriers can be taken advantage of in two different ways. First, both barriers could operate simultaneously. This guarantees a certain degree of fault-tolerance, since at least two sensors will detect the intruder. Second, one barrier could be inactive (sleep) while the other is active (awake). When the first barrier is running out of power, the second barrier is activated. This doubles the lifetime of the network. Both of these approaches can be used independently, or they can be combined; e.g., if ten barriers exist, we could have two barriers active at any time, which results in a network lifetime of five sensor lifetimes. This is known as k-barrier-coverage, where k is the number of concurrently active barriers.

In this paper, we focus on the second approach. I.e., one barrier is active at all times. We discuss extending this to the k-barrier approach in our concluding remarks.

Consider next Fig. 1(b). Assume that in addition to prevent intruders from crossing from top to bottom, we want to prevent intruders from crossing from left to right, as indicated by the direction I_2. This could be accomplished by adding additional sensor barriers from top to bottom; i.e., consider by adding two additional virtual nodes, S_2 and T_2, and constructing barriers between them.

The above improves the security of the area, but it is still possible to enter the area from one side and exit out another, as shown by direction I_3 in Fig. 1(c). To prevent this type of movement, along with the ones discussed above, we introduced the notion of a *reinforced barrier*. In Fig. 1(c), we consider placing virtual nodes in opposite corners, and constructing sensor barriers between them in a crossed pattern. This effectively eliminates every undetected movement across the sensor field.

Maximizing lifetime with reinforced barriers is more complicated than maximizing lifetime in regular barriers. This is due to the interaction between barriers that cross each other. Consider Fig. 2 as an example. It consists of three barriers from S_1 to T_1, and three additional barriers from S_2 to T_2. 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. However, assume that some barriers have nodes in common. These are denoted by circles in Fig. 2. Thus, B_4 has a node in common with each of B_1, B_2, and B_3, and B_6 has a node in common with B_1. This affects the lifetime as follows.

Assume B_4 is paired with any of B_1, B_2, or B_3, and activated. As the lifetime of B_4 expires, so will all of nodes a, b, and c. This will also eliminate barriers B_2 and B_3, since they are now missing a node. Hence, the network lifetime is just one sensor lifetime. On the other hand, assume we pair up B_1 with B_6, and B_2 with B_5. This increases the network lifetime to two sensor lifetimes.

Due to this complexity, we cannot directly apply maximum flow algorithms to find independent paths of sensors, as done in [11]. Instead, we find reinforced barriers via three iterative algorithms which we present in Section V.

III. Related Work

The notion of barrier-coverage was firstly introduced by Gage [12] in the context of robotic sensors. In [9], Kumar et al. introduced the notion of k-barrier-coverage, which is a

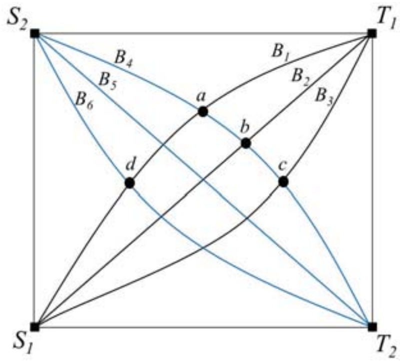


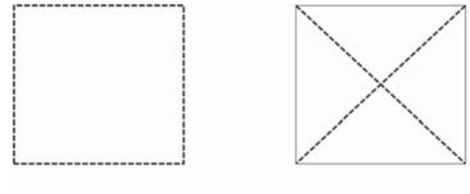
Fig. 2. Dependency property in reinforced barrier-coverage.

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. In [13], 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.

In [11], 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 an optimal sleep-wakeup schedule for k-barrier-coverage. 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 [14]. Moreover, Kim et al. introduced a new security problem in the existing sleep-wakeup scheduling algorithms to maximize lifetime non-penetrable barrier-coverage of wireless sensors [15]. 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 [16].

On the other hand, many researchers have studied perimeter barriers which is another type of barrier. Constructing perimeter barriers requires finding sensor chains enclosing the region with the sensing areas of any two adjacent sensors overlapping with each other's to detect attackers from either entering its interior or exiting from it [17]. For perimeter barrier coverage on a simple polygon, Bhattacharya et al. proposed several algorithms to determine the optimal locations and the movement scheme in [18]. In [19], Hung et al. studied the perimeter coverage problem where the perimeter of a big object needs to be monitored, but each sensor can only cover a single continuous portion of the perimeter. They proved the perimeter coverage scheduling problem is NP-hard and then proposed scheduling algorithm to maximize network lifetime.

Note that our reinforced coverage is different from perimeter coverage if we assume that the dotted line represents each barrier and a length of square is l in Fig. 3. Perimeter barrier may detect various penetration types of the intruder as Fig. 3. (a). However, while perimeter coverage usually uses mobile sensors to construct the barrier, sensors nodes in our reinforced coverage are fixed after random deployment. Even if we intuitively compare reinforced barriers with perimeter barriers for total length of barriers in a given squared-shaped area, our reinforced barriers take an advantage that total length of barriers $2(\sqrt{2} \cdot l)$ of reinforced coverage should be shorter than total length of barriers $4 \cdot l$ of perimeter coverage.



(a) Perimeter barrier (b) Reinforced barrier

Fig. 3. Difference between perimeter barrier and reinforced barrier.

IV. Reinforced Barrier-coverage

In this section, we define our problem formally. Then, we describe how to create reinforced barriers to solve the problem.

A. Problem Definition

We consider a square area, A, where a set of sensors, U, have been randomly deployed. The location of each sensor is fixed after its deployment. Each sensor can be either in sleep mode, in which case it uses a negligible amount of power, or in service mode, in which it senses its environment. Once a sensor is set into service mode, it remains in this mode until it runs out of power. All sensors are assumed to have an equal amount of power, and thus, an equal lifetime. For simplicity, we also assume that all sensors have the same range, which we denote by g.

We enhance the area A with four virtual sensors (s_1, d_1, s_2, d_2), one at each corner, as shown in Fig. 2. These four sensors have unlimited lifetime, and simplify the definitions that follow.

Two sensors, u_1 and u_2, are said to be *neighbors* if the Euclidian distance between them is at most $2 \cdot g$. A *path* of sensor nodes is a sequence of sensors, $u_1, u_2, \ldots, u_n$, where:

- u_i and u_{i+1} are neighbors, $1 \leq i \leq n-1$.
- $u_1 = s_1$ and $u_n = d_1$, or, $u_1 = s_2$ and $u_n = d_2$.

We are now ready to present the definition of a reinforced barrier.

Definition 4.1 (Reinforced Barriers):

- A *reinforced barrier*, or r–barrier, consists of two paths of sensor nodes: one path from s_1 to d_1 and another from s_2 to d_2. These two paths need not be disjoint.
- A *collection* C of r–barriers consists of a set of r–barriers, where for each $p \in C$ and $q \in C$, p and q are disjoint, i.e., they have no sensors in common.
- The *lifetime* of a collection C is simply its cardinality, $|C|$.

Definition 4.2 (MaxLRB): Given a set of wireless sensor nodes S deployed over an square-shaped area A, *the maximum* *lifetime reinforced barrier-coverage (MaxLRB) problem* consists of finding an r–barrier collection C_{max} with maximum lifetime.

V. *MaxLRB* HEURISTICS

We next present our three heuristics for obtaining collection C of r–barriers. Our objective is to search for node-disjoint paths from s_1 to d_1, and also node-disjoint paths from s_2 to d_2, and then combine them together in order to construct r–barriers.

To find disjoint paths, we use an approach similar to the one used in the Stint algorithm for straight-barriers [11]. We first present our adaptation of [11] to finding disjoint paths from s_1 to d_1. Finding disjoint paths from s_2 to d_2 is similar. We then present our three heuristics for combining these paths to form r–barriers.

To find the largest number of disjoint paths from s_1 to d_1, we perform the following two steps.

Step 1.
Create a *flow graph* $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$, as follows.

- For each sensor $u \in U$, include two vertices in $V(\mathcal{G})$, u_{in} and u_{out}.
- For each sensor $u \in U$, add a directed edge (u_{in}, u_{out}) to $E(\mathcal{G})$.
- For every pair of sensor nodes u and v in U 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 d_1 to $V(\mathcal{G})$.
- For each neighbor u of s_1, add the edge (s_1, u_{in}) to $V(\mathcal{G})$.
- For each neighbor u of d_1, add the edge (u_{out}, d_1) to $V(\mathcal{G})$.

Step 2.
Assign a capacity of 1 to each edge in the flow graph, and run a maximum flow algorithm, such as Edmonds-Karp algorithm [20], from s_1 to d_1. Edges with a flow of 1 will form disjoint paths from s_1 to d_1. All other edges will have a flow of 0.

The intuition behind the above two steps is as follows [11]. 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. Hence, each edge will have a flow of 0 or 1. In addition, due to the single edge between u_{in} and u_{out}, each node, other than s_1 or s_2, can only participate in a single path.

Armed with the above method to find the maximum number of disjoint paths, we move to our heuristics for the *MaxLRB* problem.

As a side-note, it is worthy of mentioning that finding the maximum number of disjoint paths as above takes polynomial time. However, r-barriers are akin to multi-commodity flows, since we have two sets of paths, one from s_1 to d_1 and another from s_2 to d_2. The multi-commodity flow problem is NP-complete, even in the case of only two commodities and edges with 0-1 capacity. r-barriers have the additional constraint that each path from s_1 to d_1 has to be paired with a path from s_2 to d_2, and these two paths may share nodes. We thus conjecture that *MaxLRB* is NP-hard, but a proof remains elusive. We leave this for future work.

A. *Approach 1: independent-path*

As discussed in Section IV, in a collection of r–barriers, sensor nodes cannot belong to more than one r–barrier. One way to ensure this is the case is that all paths chosen from s_1 to d_1 do not share any sensors with paths from s_2 to d_2. In this manner, enabling an r–barrier has no effect on the future performance of other r–barriers.

To accomplish this, one is tempted to first find all the disjoint paths from s_1 to d_1, remove these paths, and then find all the disjoint paths from s_2 to d_2. Unfortunately, this will likely lead to having no paths from s_2 to d_2, because many nodes will be consumed in the first step.

Therefore, we choose an iterative approach, in which one complete r–barrier is found at a time. The steps are as follows.

- Obtain the maximum number of node-disjoint paths from s_1 to d_1. Also, obtain the maximum number of node-disjoint paths from s_2 to d_2.
- An r–barrier is obtained by choosing a path from s_1 to d_1 and another path from s_2 to d_2 such that these two paths are node-disjoint.
- Remove from the field the sensors comprised by the above r–barrier.

The above steps are repeated until we can not find any additional pair of node-disjoint paths. The pseudocode is presented in Algorithm 1 in more detail, which we call as *independent-path*.

Note that the pair of paths chosen at each iteration may affect the availability of other pairs of paths in subsequent iterations. Due to the complexity of the problem, we simply choose one pair of paths at random. We will investigate other variations in future work.

B. *Approach 2: shared-path*

Our second approach, which is referred as *shared-path*, takes advantage that some nodes can be reused within the same r–barrier, provided they don't interfere with another r-barrier. That is, the path from s_1 to d_1 is able to share sensors with the path from s_2 to d_2, provided these two paths belong to the same r-barrier, and do not participate in any other r-barrier. The intuition behind this approach is that shared nodes serve a double purpose, by participating in each of the two paths, which makes a more effective use of the available sensors.

We again iterate, adding one r–barrier at a time, until no more can be added. The steps are each iteration are as follows.

- Obtain the maximum number of node-disjoint paths from s_1 to d_1. Also, obtain the maximum number of node-disjoint paths from s_2 to d_2.
- An r–barrier is obtained by choosing a path from s_1 to d_1 and another path from s_2 to d_2 such that these two paths share at least one sensor node.

Algorithm 1 ***independent-path***

Inputs: U, A, g, Output: C

1: set r–barrier collection: $C \leftarrow \emptyset$;
2: set unselected sensor nodes: $U' \leftarrow U$;
3: **while** $U' \neq \emptyset$ **do**
4:     create a flow graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ using A, U', and g;
5:     let P_1 be the set of node-disjoint paths from s_1 to d_1;
6:     let P_2 be the set of node-disjoint paths from s_2 to d_2;
7:     find a pair of node-disjoint paths (p_1, p_2), where $p_1 \in P_1$ and $p_2 \in P_2$;
8:     **if** (p_1, p_2) exists **then**
9:         $C \leftarrow C \bigcup (p_1, p_2)$;
10:         $U' \leftarrow U' - (nodes(p_1) \bigcup nodes(p_2))$;
11:     **else**
12:         break;
13:     **end if**
14: **end while**
15: return C

- 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. We simply replace line 7 by the following line:

> find 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.

C. Approach 3: combined-paths

Approach 3 can be considered as a combination of approaches 1 and 2. We refer the approach 3 as *combined-paths*. It adds one r–barrier at a time. It first adds as many barriers as possible whose two paths have no shared sensors. When this process cannot continue, then r–barriers are added whose two paths share sensors. We will discover in Section VI that this approach outperforms the other two. The pseudocode is described in Algorithm 2 in more detail.

Algorithm 2 ***combined-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 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**
        $C \leftarrow C \bigcup (p_1, p_2)$;
        $U' \leftarrow U' - (nodes(p_1) \bigcup nodes(p_2))$;
    else
        find a pair of paths (p_1, p_2), where $p_1 \in P_1$ and $p_2 \in P_2$ that share at least one sensor;
        if (p_1, p_2) exists **then**
            $C \leftarrow C \bigcup (p_1, p_2)$;
            $U' \leftarrow U' - (nodes(p_1) \bigcup nodes(p_2))$;
        else
            break;
        end if
    end if
end while
return C

VI. Experimental Evaluation

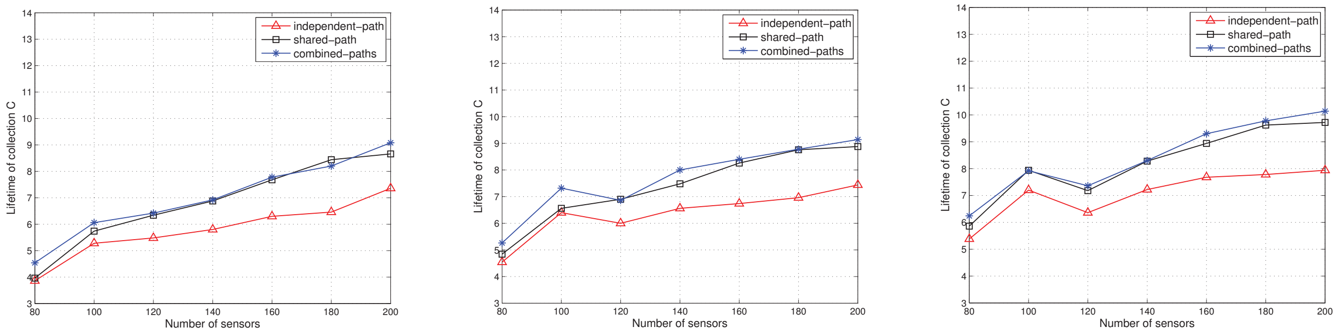
We next evaluate the performance of the three approaches of Section V. We simulated our various experiments in a square-shaped area of size 500 × 500 m^2 where U sensor nodes are randomly deployed in the region initially. Each experiment represents the average result of 100 different graphs. The number of sensors ranges from 80 to 200 and the transmission radius of sensors ranges from 70 to 100. Interestingly, *combined-paths* outperforms *independent-path* and *shared-path* when we have checked the performance as a whole. Now, we describe the results by largely two different scenarios.

As the first performance analysis in our contribution, we compare *independent-path*, *shared-path* and *combined-paths* for lifetime of a collection C, respectively. As we explained lifetime of a collection C or its cardinality $|C|$ in Section IV, $|C|$ is a total network lifetime of reinforced barrier-coverage. Therefore, maximizing value $|C|$ results in maximizing lifetime of r-barriers. Fig. 4 shows results for three approaches by different radius g = 70, 75 and 80, respectively. In this simulations, we have checked *combined-paths* shows better performance than *independent-path* and *shared-path*. Also, as the number of sensor nodes increases, lifetime of a collection C increases for three approaches.

In our second group of simulations, three approaches are compared by different number of sensor nodes. Fig. 5 represents results for three approaches by different total number of sensor nodes n = 100, 150 and 200, respectively. As you can see the Fig. 5, as a transmission radius of sensor increases, lifetime of a collection C increases in all approaches. Moreover, we have checked that as the number of sensor increases, *combined-paths* shows significantly better performance than other approaches. By Fig. 5, we can finally verify that *combined-paths* outperforms *independent-path* and *shared-path* by different number of sensors.

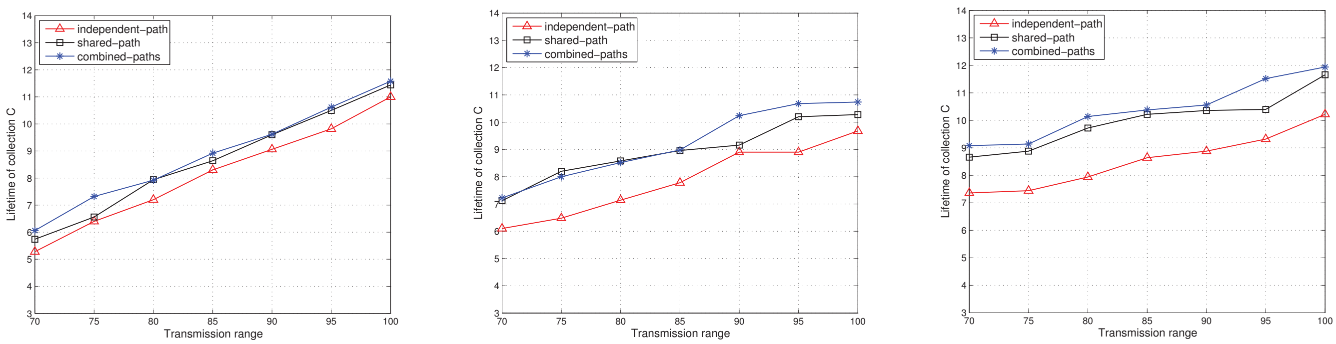
VII. Concluding Remarks

In this paper, we have introduced a new barrier-coverage problem whose objective is to maximize network lifetime such that any penetration type of the intruder is detected. To solve the problem, we created a new type of barriers, r–barriers, which is able to detect any penetration variation of the intruder. Furthermore, we have proposed three different approaches to maximize network lifetime for r–barriers and then have evaluated their performances through extensive experiments, respectively. As a future work, we plan to extend our reinforced barrier-coverage to camera sensor networks.



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

Fig. 4. Comparison for lifetime of collection C for different radii by three approaches in 500×500 region



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

Fig. 5. Comparison for lifetime of collection C for different number of sensors by three approaches in 500×500 region

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: a Survey," *Computer Networks*, vol 38, issue 4, pp. 393-422, March, 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network Survey," *Computer Networks*, vol. 52, no. 12, pp. 2292-2330, Aug., 2008.

[3] C. Huang and Y. Tseng, "The Coverage Problem in a Wireless Sensor Network," in *Proc. of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2003.

[4] H. Zhang and J. Hou, "On Deriving the Upper Bound of α-lifetime for Large Sensor Networks," in *Proc. of the ACM International Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc)*, 2004.

[5] M. Cardei, M. Thai, Y. Li and W. Wu, "Energy-Efficient Target Coverage in Wireless Sensor Networks," *IEEE Annual International Conference on Computer Communications (INFOCOM)*, March, 2005.

[6] S. Gao, X. Wang, and Y. Li, "p-Percent Coverage Schedule in Wireless Sensor Networks," *The International Conference on Computer Communications and Networks (ICCCN)*, August, 2008.

[7] C.T. Vu, G. Chen, Y. Zhao, and Y. Li, "A Universal Framework for Partial Coverage in Wireless Sensor Networks," *IEEE Int'l Performance Computing and Communications Conf. (IPCCC)*, Dec., 2009.

[8] Y. Li, C. Vu, C. Ai, G. Chen, and Y. Zhao, "Transforming Complete Coverage Algorithms to Partial Coverage Algorithms for Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 22, issue 4, pp. 695-703, 2011.

[9] S. Kumar, T.H. Lai, and A. Arora, "Barrier Coverage with Wireless Sensors," in *Proc. of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 284-298, 2005.

[10] A. Saipulla, C. Westphal, B. Liu, and J. Wang, "Barrier Coverage of Line-based Deployed Wireless Sensor Networks," in *Proc. of the 28th IEEE Conference on Computer Communications (INFOCOM)*, 2009.

[11] S. Kumar, T.H. Lai, M.E. Posner, and P. Sinha, "Maximizing the Lifetime of a Barrier of Wireless Sensors," *IEEE Transactions on Mobile Computing (TMC)*, vol. 9, no. 8, August, 2010.

[12] D. Gage. "Command control for many-robot systems," in *Proc. of the Nineteenth Annual AUVS Technical Symposium (AUVS-92)*, 1992.

[13] B. Liu, O. Dousse, J. Wang, and A. Saipulla, Strong barrier coverage of wireless sensor networks, in *Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2008.

[14] D. Ban, Q. Feng, G. Han, W. Yang, J. Jiang, and W. Dou, "Distributed Scheduling Algorithm for Barrier Coverage in Wireless Sensor Networks," in *Proc. of the 2011 Third International Conference on Communications and Mobile Computing (CMC)*, April, 2011.

[15] D. Kim, J. Kim, D. Li, S. Kwon, and A. O. Tokuta,"On Sleep-wakeup Scheduling of Non-penetrable Barrier-coverage of Wireless Sensors," in *Proc. of IEEE Global Communications Conference (Globecom)*," 2012.

[16] A. Chen, S. Kumar, T.H. Lai,"Barrier Coverage with Wireless Sensors," in *Proc. of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom)*," 2005.

[17] C. Wang, B. Wang and W. Liu,"Movement strategies for improving barrier coverage in wireless sensor networks: A survey," in *Proc. of the 13th International Conference on Communication Technology (ICCT)*,", pp. 938-943, 2011.

[18] B. Bhattacharya, M. Burmester, Y. Hu, E. Kranakis, Q. Shi, and A. Wiese, "Optimal movement of mobile sensors for barrier coverage of a planar region," *Theoretical Computer Science (Elsevier)*, vol. 410, pp.5515-5528, 2009.

[19] K. Hung and K. Lui, "Perimeter Coverage Scheduling in Wireless Sensor Networks Using Sensors with a Single Continuous Cover Range," *EURASIP Journal on Wireless Communications and Networking*, 2011.

[20] J. Edmonds and R.M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," *Journal of the ACM*, vol. 19, issue 2, pp. 248-264, 1972.