Imperfection Better Than Perfection: Beyond Optimal Lifetime Barrier Coverage in Wireless Sensor Networks

Haiming Luo*†, Hongwei Du*†, Donghyun Kim‡, Qiang Ye§, Rongrong Zhu*† and Jinglan Jia¶

*Department of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, China

†Shenzhen Key Laboratory of Internet Information Collaboration, China

Emails: cshmluo@gmail.com, hongwei.du@ieee.org, hitzrr13@gmail.com

‡Department of Mathematics and Physics, North Carolina Central University, Durham, NC, USA

Email: donghyun.kim@nccu.edu

§Department of Computer Science and Information Technology, University of Prince Edward Island, Canada

Email: qye@upei.ca

¶Departmant of Mathematics, Central China Normal University, Wuhan, China

Email: windy_lan@163.com

***Abstract*—Barrier coverage based on Wireless Sensor Networks (WSNs) has been widely used to prevent intruder trespassing in monitoring systems. Traditionally, enabling perfect barrier coverage is considered the most important goal of barrier coverage studies. Imperfect coverage has been deemed to be a failure. In our research, we attempted to use the redundant sensor nodes in WSNs to prolong the optimal network lifetime of barrier coverage by adding imperfect barrier coverage. Specifically, we devised two schemes, CIBC-1 and CIBC-2, to construct imperfect barrier coverage in order to improve the performance of the existing optimal network lifetime scheduling algorithms for barrier coverage. Our simulation results indicate that our schemes can significantly extend the network lifetime resulting from the state-of-the-art network lifetime scheduling algorithms.**

***Index Terms*—Wireless sensor networks, imperfect barrier coverage, network lifetime**

I. Introduction

Wireless sensor networks are multi-hop and self-organized networks that are formed by wireless sensor nodes deployed in the monitoring region. Barrier coverage in wireless sensor networks is inspired by the moats which have been used to detect intruders [1]. Of course, the barriers based on wireless sensor networks are formed by wireless sensors instead of the moats. The sensing regions of wireless sensors can overlap in the horizontal direction and form a long barrier. Barrier coverage requires fewer sensors than area coverage because barrier coverage does not have to cover every point [1]. The lifetime of wireless sensors is a bottleneck in barrier coverage because sensor nodes are typically powered by batteries.

Many studies have been working on how to schedule the sensors deployed in a monitoring region to prolong the network lifetime for barrier coverage [2],[3],[4],[5]. Typically, Kumar et al. [2] have proposed an optimal sleep-wakeup scheduling algorithm for k-barrier coverage. This algorithm generates the optimal network lifetime. They proved an important theorem: when the maximum number of node-disjoint paths in the coverage graph is m, the network can provide k-barrier coverage for the maximum time is m/k where $m \geqslant k$. Specially, when k is equal to 1, the optimal network lifetime is m. In Kumar's study, node-disjoint paths between two virtual nodes in the coverage graph form a perfect barrier coverage. However the optimal network lifetime is maintained by the perfect barrier coverage, redundant sensor nodes are not utilized in his study. We analyze how redundant sensor nodes make a contribution to prolonging the network lifetime even though they do not form perfect barriers in Section III.

In this paper, we introduce a new concept called imperfect barrier coverage and show that redundant sensor nodes can form an imperfect barrier coverage which can also prevent the intruder trespassing the monitoring region. We can beyond the optimal lifetime of barrier coverage by adding imperfect barrier coverage. We devise two algorithms CIBC-1 and CIBC-2 to construct imperfect barrier coverage by utilizing redundant sensor nodes. CIBC-1 and CIBC-2 are based on two different assumptions. For simplicity, we set k to 1 and study the network lifetime of both perfect and imperfect barrier coverage. Indeed we improve the optimal sleep-wakeup scheduling algorithm by Kumar et al. [2].

The rest of the paper is organized as follows: Section II presents the related work. In Section III we introduce some preliminaries and some important definitions. We show the motivation of considering imperfect barrier coverage. By considering imperfect barrier coverage, we can beyond optimal lifetime barrier coverage. In Section IV, we devise two algorithms to construct imperfect barrier coverage by utilizing redundant sensor nodes. Section V presents the performance evaluation of our algorithms and analyses the simulation results. Section VI concludes this paper.

II. RELATED WORK

The concept of barrier coverage first appeared in the literature [6] in the context of robotic sensors. They present that the objective of barrier coverage is to achieve a static arrangement of elements that minimizes the probability of undetected enemy penetration through the barrier. The reader may find more related information from a nice survey [7] on the barrier coverage.

According to different service requirements, barrier coverage can be depicted by *k*-barrier coverage in a wireless sensor network where all crossing paths through the region are *k*-covered, that is to say, any path through the region intersects with the sensing range of at least *k* distinct sensors. In [8], it characterizes the asymptotic behavior of the barrier coverage of large-scale sensor networks which lies within a two-dimensional plane and a two-dimensional strip and explores the fundamental limits of the network coverage. Kumar et al. [1] introduce two kinds of barrier coverageweak *k*-barrier coverage and strong *k*-barrier coverage. And they derive the critical conditions for achieving weak *k*-barrier coverage by using minimum number of sensors. However, critical conditions for strong *k*-coverage are still an open problem. Liu et al. [9] derive critical conditions for strong *k*-barrier coverage, filling the gap in the critical conditions for barrier coverage. It also presents an efficient distributed algorithm to construct sensor barriers on long strip areas of irregular shape without any constraint on crossing paths via using the critical conditions.

To prolong the network lifetime, many feasible strategies have been presented in literature [7], [8], [9], [10], [11] including group sensors and turn off the redundant sensor for saving energy consumption. Chen et al. [4] introduce the concept of local barrier coverage to address the limitation that sensors cannot locally determine whether the deployment provides global barrier coverage. And they develop a novel sleep-wake up algorithm for maximizing the network lifetime, called Localized Barrier Coverage Protocol (LBCP). Du et al. [12] focus on maximizing the network lifetime under a novel *k*-discrete barrier coverage model, whose goal is to cover some specific discrete points of interest by deploying sensors in *k* lines to form barriers.

III. PRELIMINARIES

In the following discussion, we set *k* to 1 and assume each sensor's lifetime is one hour for simplicity, introduce perfect barrier coverage and imperfect barrier coverage, analyze how to beyond optimal lifetime barrier coverage by adding the imperfect barrier coverage.

Firstly, we introduce an optimal scheduling algorithm called Stint [2]. It constructs a coverage graph from the sensor network then calls a standard max-flow algorithm Edmonds-Karp to compute maximal node-disjoint paths between two virtual nodes *s* and *t* in a coverage graph. Consider a set of nodes which are randomly deployed in a monitoring region *B* and the disk model is adopted as sensing model. The sensor network *N* is shown as Fig.1. The communication radius is assumed to be twice as the sensing radius.

Definition 1 (Coverage graph *G(V,E)* [2]). The coverage graph is formed by the sensor network *N*. *V* contains all sensor nodes and two virtual nodes while *E* contains all edges. There exists an edge between two sensor nodes in *V* whose sensing range overlaps. Two virtual nodes *s* and *t* are placed to the left of the monitoring region and to the right of the monitoring region respectively. There exists an edge between *s(t)* and a sensor node *u* if *u*'s sensing rage covers the left(right) border of the monitoring region. Fig.2 shows an example of coverage graph. Let *m* be the maximum number of node-disjoint paths between *s* and *t*.

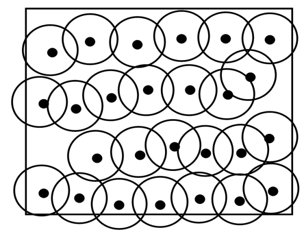

Fig. 1. A set of nodes are randomly deployed in a monitoring region *B*, they form a sensor network *N*.

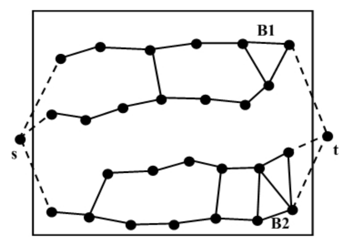

Fig. 2. An example of coverage graph. Two sensor nodes are connected if their sensing ranges overlap.

In Fig.2, we can achieve two node-disjoint paths between *s* and *t* which is the output of Edmonds-Karp algorithm. These two node-disjoint paths construct two barriers B_1 and B_2. According to Stint, we can active B_1 and B_2 in sequence and prevent intruders trespassing the monitoring region for double lifetime of sensor nodes. Such as barriers B_1 and B_2, we call them perfect barriers, the following is a definition about perfect barrier coverage.

Definition 2 (Perfect barrier coverage) A set of sensor nodes $\{s_1, s_2, \ldots, s_n\}$ are deployed in a monitoring region *B*. Some nodes whose sensing ranges overlap cover the length of region *B* completely. These nodes line up in a horizontal direction and form a perfect barrier coverage. For example B_1 and B_2 form perfect barrier coverage in Fig.3. "Perfect" means that any vertical crossing path or curved crossing path of an intruder can be detected by the barriers.

Stint computes the maximum network lifetime only by considering perfect barriers. However, there are redundant sensor nodes between neighboring perfect barriers which can also prevent intruders trespassing monitoring region. We show how redundant sensor nodes play a role in Fig.3. The sensor network in Fig.3 is the same as that in Fig.1. An intruder *I* is located at the top of monitoring region *B*. Applying Stint, we can obtain two perfect barriers(consist of all the nodes in B_1 and B_2). Then we active B_1 for one hour first, after B_1 is expired, we active B_2 which means we can prevent intruders trespassing the monitoring region for two hours. It is proved to be an optimal sleep-wakeup scheduling algorithm in [2].

However, if we utilize the redundant sensor nodes between B_1 and B_2, we can beyond optimal lifetime of the barrier coverage. The redundant sensor nodes construct two barriers B_3 and B_4 that could not cover the length of region *B* completely. If the intruder *I* knows the sensing ranges, it can follow a curved path to invade which is the dotted line shown in Fig.3. The delay we can prolong depends on the location of *I*, the speed of *I* and the size of monitoring region *B*. On the other hand, if *I* knows nothing about sensing ranges, it would prefer the vertical crossing paths that are shortest paths and is detected by B_3 and B_4. Such that, B_3 and B_4 can prevent *I* for at least one hour.

Definition 3 (Imperfect barrier coverage) Imperfect barrier coverage does not guarantee covering the length of monitoring region completely. "Imperfect" means all the vertical crossing paths of an intruder can be detected while some curved crossing paths are still available for an intruder. In this paper, we consider that weak barrier coverage [3] is a special case of imperfect barrier coverage. B_3 and B_4 in Fig.3 form imperfect barrier coverage.

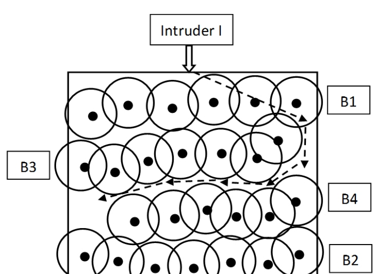

Fig. 3. Redundant sensor nodes prevent the intruders trespassing monitoring region.

IV. CONSTRUCT IMPERFECT BARRIER COVERAGE

In this section, we focus on constructing the imperfect barrier coverage between neighbor perfect barriers which can prolong the network lifetime to prevent intruders trespassing the monitoring region. We assume that the deployed sensor nodes are redundant. Actually, if we attempt to enhance the strength of the barrier coverage, more sensor nodes are needed, then there are more redundant nodes that are not utilized. We divide the monitoring region into many vertical strips with equal length and assume the set of vertical strips are *R*=$\{r_1,r_2,r_3,\ldots,r_m\}$. We assume that the monitoring region *B* is also a coordinate system. The upper left corner of *B* is the origin point of the coordinate system. The right direction of the origin point is the positive direction of X-axis while the below direction of the origin point is the positive direction of Y-axis. Each sensor node has X-coordinate and Y-coordinate. We give some further important definitions as following:

Definition 4 (Vertical strip r_i is covered by sensor node *s*): One vertical strip r_i is said to be covered by a sensor node *s* if and only if any crossing path of *I* in r_i is detected by *s*.

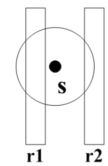

Fig. 4. The vertical strip r_1 is covered by sensor node *s*, while the vertical strip r_2 is not covered by sensor node *s* for there exist crossing paths in r_2 that are not detected by *s*.

Definition 5 (None-crossing imperfect barrier coverage): Suppose in an imperfect barrier covered set, there exists an edge between each pair of sensor nodes. For any two imperfect barrier covered sets S_1 and S_2, there exists no crossing edges between these two sets. S_1 and S_2 form none-crossing imperfect barrier coverage.

For any two imperfect barrier covered sets, if the Y-coordinate of each sensor node in one imperfect barrier covered set is bigger than the largest Y-coordinate of sensor nodes in the other imperfect barrier covered set, then they construct none-crossing imperfect barrier coverage. For example, suppose there are two imperfect barrier covered sets S_1 and S_2, the Y-coordinate of each sensor node in S_1 is bigger than the largest Y-coordinate of sensor nodes in S_2. If we consider the imperfect barrier covered set as a whole, then S_2 is "above" S_1 in the monitoring region which means there exists no crossing edges between S_1 and S_2. According to **Definition 5**, S_1 and S_2 construct none-crossing imperfect barrier coverage.

In the rest of this section, we devise two algorithms CIBC-1 and CIBC-2 based on different assumptions to construct the imperfect barrier coverage. In CIBC-1, we assume intruders know nothing about the deployment of sensor nodes and sensing ranges. They prefer vertical crossing paths that are shortest. Under this assumption, imperfect barrier coverage is similar to weak barrier coverage. Ban et al. [3] proposed a distributed scheduling algorithm for weak barrier coverage called DBCS. DBCS can generate as many cover sets as possible which can prolong the network lifetime. However, in DBCS a sensor node may get to sleep periodically which means that an intruder can escape the sensing region of this sensor node when the sensor node is in sleeping mode. Our algorithm CIBC-1 can avoid this disadvantage of DBCS. We aim to construct none-crossing imperfect barrier coverage. In CIBC-2, intruders are assumed to know the deployment and sensing ranges. They prefer crossing paths which cost less delay. Our goal is to maximize the delay for intruders to trespass a sub-region between two perfect barriers and minimize the energy of redundant sensor nodes.

1) Algorithm CIBC-1: In CIBC-1, we input n redundant sensor nodes between two perfect barriers and vertical strips *R*, output a minimum set of nodes *S* to guarantee any vertical crossing path can be detected by one sensor node. In the initial stage, *S* represents the result of output and *U* represents the set of nodes that cover a certain vertical strip. *S* and *U* are initialized to be empty. In step 2, we construct a matrix $M[R_j][s_i]$ where the vertical strips represent the row coordinates and the sensor nodes represent the column coordinates. The sensor nodes are sorted by increasing order of the Y-coordinate of their location which means if the Y-coordinate of sensor node s_i is smaller than the Y-coordinate of sensor node s_j, then the column coordinate s_i represents is smaller than the column coordinate s_j represents in $M[R_j][s_i]$. In step 3, we assign value to $M[R_j][s_i]$, if a vertical strip R_j is covered by a sensor node s_i, then $M[R_j][s_i]$ is set to be 1 otherwise $M[R_j][s_i]$ is set to be 0. From step 4 to step 17, we traverse each vertical strip R_j. For each R_j, we add sensor nodes that cover R_j to *U*. Then we examine whether there is a sensor node in *U* belonging to *S*. If yes, it means that the current vertical strip is covered by a certain node in *S* and we don't need to add a new sensor node. If no, then we select the sensor node with smallest column coordinate and add it to *S*. Specially, if there is no sensor nodes that cover the current vertical strip, *S* is set to be NULL and the loop is interrupted which means that it fails to construct an imperfect barrier coverage. Finally, if the return is not NULL, we update the redundant sensor nodes by removing the sensor nodes in *S* and the sensor nodes whose Y-coordinate is less than or equal to the largest Y-coordinate of sensor nodes in *S*. After that, we input the remaining redundant sensor nodes to CIBC-1 and find another minimum set of nodes to construct imperfect barrier coverage.

Sensor nodes of each imperfect barrier covered set generated by CIBC-1 have minimal Y-coordinates. Suppose CIBC-1 generate an imperfect barrier covered set *G* and the imperfect barrier covered set whose sensor nodes have minimal Y-coordinates is denoted as *P*. We can divide *P* into two sets P_s and P_d, where P_s are the sensor nodes that also belong to *G* and P_d are the sensor nodes that not belong to *G*. Randomly choose a sensor node *b* in P_d , remove *b* in *P*. Then there must be some vertical strips that are not covered. Assume they are R_{j1}, $R_{j2} \cdots R_{jk}$. For the vertical strips R_{j1}, $R_{j2} \cdots R_{jk}$, we can handle them one by one. For example R_{jk}, we can find a sensor node *a* in *G* covering it. In each iteration of CIBC-1, we select the sensor node with small Y-coordinate, thus the Y-coordinate of sensor node *a* is less than Y-coordinate of sensor node *b*. We can use sensor node *a* to substitute sensor node *b*. Such that, *G* can be transformed to *P* which means *G* has the same effect as *P*. Thus, CIBC-1 can generate an imperfect barrier covered set whose sensor nodes have minimal Y-coordinates in each iteration. After each iteration, there are as many remaining redundant sensor nodes as possible for the next iteration. Such that, CIBC-1 can construct maximum none-crossing imperfect barrier covered sets.

Algorithm 1 CIBC-1

```
1:  Initialize: S=NULL, M[R_j][s_i]=NULL, U=NULL;
2:  Construct a matrix M[R_j][s_i], where sensor nodes are
    sorted by increasing order of Y-coordinate of their loca-
    tion.
3:  Assign to M[R_j][s_i]. If R_j is covered by s_i, M[R_j][s_i]=1
    otherwise, M[R_j][s_i]=0; (1⩽j⩽m, 1⩽i⩽n)
4:  for each j ∈ [1, m] do
5:     Add s_i to U where M[R_j][s_i]==1;
6:     if U==NULL then
7:        S=NULL;
8:        break;
9:     else
10:       if U ∩ S==∅ then
11:          Select the sensor node with smallest column coor-
             dinate in U to S;
12:       else
13:          Do nothing;
14:       end if
15:    end if
16:    Reset U=NULL;
17: end for
18: Return S;
```

2) Algorithm CIBC-2: In CIBC-2, we aim to construct an imperfect barrier coverage that can maximize the delay intruders trespassing the sub-region between neighboring perfect barriers. Based on the assumption that intruders know sensing ranges, CIBC-1 can only prolong less delay. Instead, we utilize connected components to construct the imperfect barrier coverage because a set of connected sensing ranges make a contribution to prolonging more delay. We input n redundant sensor nodes and output a set of nodes *S* for the imperfect barrier coverage. Firstly we initialize some important variables. *S* is the set of nodes for the output, *Q* is a queue of the connected components and U_m is a set of vertical strips. In the second step, we compute all the connected components *ConCompn*. Each connected component C_i contain a set of connected sensor nodes. From step 3 to step 6, for each connected component C_i, we compute the set of vertical strips M_i covered by C_i and compute the average Y-coordinate of nodes in C_i. Then add C_i to *Q*. In step 7, we sort *Q* by cY_i in increasing order. The main part of Algorithm 2 is from step 8 to step 17. Each iteration, we get the first element C_{temp} of *Q*. M_{temp} is the set of vertical strips covered by C_{temp}. Then put the nodes in C_{temp} to *S* and update U_m which is the set of vertical strips covered in the monitoring region. After that, the first element in *Q* is removed. If *Q* is empty or all the vertical strips are covered, the loop is terminated. If the return *S* is NULL, it means although all the connected components are selected, there are some vertical strips that are not covered and the remaining nodes can not construct an imperfect barrier coverage. Otherwise, the nodes in S construct an imperfect barrier coverage and we can update the remaining nodes by removing the nodes in S and call the algorithm again to find another set of nodes for imperfect barrier coverage.

Algorithm 2 CIBC-2

```
Initialize S=NULL, Q=NULL, U_m=NULL;
Compute all the connected components ConCompn={C_1, C_2, C_3, ..., C_k};
for each i ∈ [1, k] do
    Compute M_i and cY_i;(M_i is the set of vertical strips covered by nodes in C_i and cY_i is the average Y-coordinate of nodes in C_i).
    Add C_i to Q;
end for
Sort Q by cY_i in increasing order.
while ∃ r ∉ U_m (r is a vertical strip) do
    if Q.empty() then
        S=NULL;
        break;
    end if
    C_temp=Q.front();
    Put the nodes in C_temp to S;
    U_m=U_m ∪ M_temp;
    Q.pop();
end while
Return S;
```

V. SIMULATION

In this section, we do extensive simulations by Java and Matlab to evaluate the performance of our two algorithms. Kumar et al. [2] present an optimal sleep-wakeup scheduling algorithm for k-barrier coverage. When the number of perfect barriers is m and k is 1, the perfect barriers can be activated one by one in sequence and the optimal network lifetime is equal to m/k. In our simulations, we set k to 1 and construct imperfect barrier coverage between neighboring perfect barriers. The scheduling algorithm is improved like that after a perfect barrier is exhausted, activate the imperfect barrier covered sets in sequence before the next perfect barrier. We compare the improved scheduling algorithms with the optimal scheduling algorithm for barrier coverage in [2]. In our simulations, a wireless sensor network was modeled as a set of nodes randomly deployed in a 400×100m rectangular monitoring region. Without loss of generality, we set sensing radius is 10m and communication radius is 20m initially. The number of sensor nodes in the network vary from 100 to 700 in steps of 100. We assume that an intruder trespasses the monitoring region with a velocity of 2m/s. The performance of improved scheduling algorithm by CIBC-1 is much better than that of the optimal scheduling algorithm because we only consider the vertical crossing paths of the intruder which is a practical special case. In the other hand, the improved scheduling algorithm by CIBC-2 which considers any crossing path of the intruder also outperforms the optimal scheduling algorithm.

Fig.5 presents the performance of the optimal sleep-wakeup scheduling algorithm of a barrier coverage which is represented by "Optimal" and the improved scheduling algorithm after adding imperfect barrier constructed by CIBC-1. First of all, the network lifetime doesn't increase when the number of sensors is 100 which means redundant nodes are not enough to construct imperfect barrier coverage. But with more sensor nodes are deployed, improved scheduling algorithm by CIBC-1 outperforms optimal sleep-wakeup scheduling algorithm. When the number of sensor nodes are 400 and 700, the prolonged network lifetime is the most.

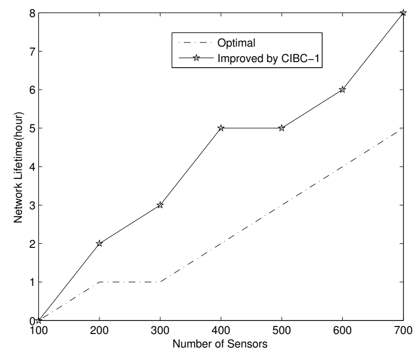

Fig. 5. Network Lifetime vs. Number of Sensor Nodes

Fig.6 presents the performance of the optimal sleep-wakeup scheduling algorithm and improved the scheduling algorithm by CIBC-1 in terms of the network lifetime versus the sensing radius given 400 deployed nodes. The communication radius is twice as sensing radius and they increase in equal proportion. We can see that when the sensing radius reaches 30, the network lifetime of the optimal scheduling algorithm almost not increase the while the network lifetime of improved scheduling algorithm by CIBC-1 still increases rapidly until sensing radius reaches 40. After that, it increases gently. This is reasonable because constructing imperfect barrier coverage needs less nodes, there are still some redundant nodes unutilized when perfect barrier coverage reach saturation.

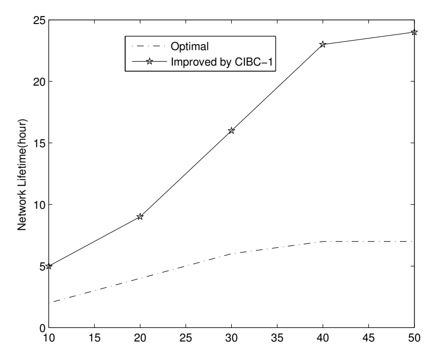

Fig. 6. Network Lifetime vs. Sensing Radius

Fig.7 presents the performance of the optimal sleep-wakeup scheduling algorithm and improved the scheduling algorithm by CIBC-2 in terms of the network lifetime versus the number of sensor nodes. The network lifetime of the improved scheduling algorithm by CIBC-2 is more than that of the optimal scheduling algorithm about 540 seconds in average.

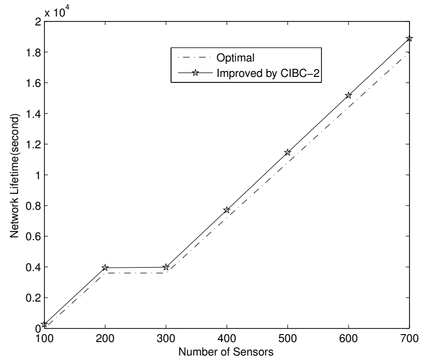

Fig. 7. Network Lifetime vs. Number of Sensor Nodes

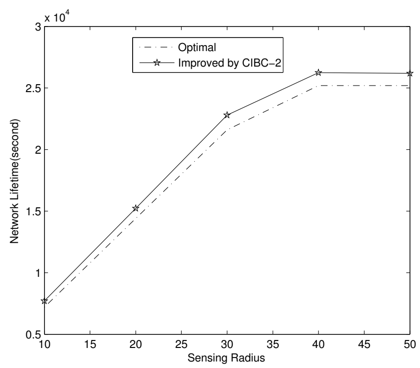

Fig. 8. Network Lifetime vs. Sensing Radius

Fig.8 presents the performance of optimal sleep-wakeup scheduling algorithm and improved the scheduling algorithm by CIBC-2 in terms of the network lifetime versus the sensing radius given 400 deployed nodes. The network lifetime of both the two scheduling algorithm keeps the consistency. It grows linearly when the sensing radius is between 10m to 30m. After that, it grows slowly and reaches to stable status. This is actually expected because under the assumption of the intruder knowing sensing ranges, the imperfect barrier coverage is constructed by connected components. It needs more redundant sensor nodes such that the network lifetime of the improved scheduling algorithm by CIBC-2 reaches to stable status sooner than that of improved scheduling algorithm by CIBC-1 in Fig.6.

VI. CONCLUSION

In this paper, we introduce the imperfect barrier coverage and analyze that using redundant nodes to construct the imperfect barrier coverage can increase the network lifetime of WSNs. We devise two algorithms to construct imperfect barrier coverage under two different assumptions. Simulations shows that both of the two algorithms can improve scheduling algorithm of barrier coverage. By constructing imperfect barrier coverage with redundant sensor nodes, we can outperform the optimal network lifetime barrier coverage. For the future work, we will study the influence of communication overhead on the network lifetime of barrier coverage and study how to devise efficient algorithms to construct barrier coverage by considering both sensing and communication.

ACKNOWLEDGMENT

This research was jointly sponsored by the National Natural Science Foundation of China under Grants 61370216 and 61100191, the Shenzhen Strategic Emerging Industries Program under Grant No. ZDSY20120613125016389. It was also supported in part by US National Science Foundation (NSF) CREST No. HRD-1345219.

REFERENCES

[1] 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), August 2005.

[2] 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.

[3] 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.

[4] A. Chen, S. Kumar and T.H. Lai, “Local barrier coverage in wireless sensor networks,” IEEE Transactions on Mobile Computing(TMC), vol.9, no.4, April 2010.

[5] H. Yang, D. Li, Q. Zhu, W. Chen and Y. Hong, “Minimum energy cost k-barrier coverage in wireless sensor networks,” in Proc. of the 5th International Conference on Wireless Algorithms, Systems, and Applications (WASA), 2010.

[6] D. W. Gage, “Command control for many-robot systems,” In Proc. of the Nineteenth Annual AUVS Technology Symposium(AUVS-92), 1992.

[7] E. Felemban, “Advanced border intrusion detection and surveillance using wireless sensor network technology,” Int’l J. of Communications, Network and System Sciences, vol. 6, no. 5, 2013.

[8] B. Liu, D. Towsley, “A study of the coverage of large-scale sensor networks,” in Proc. of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, 2004.

[9] B. Liu, O. Dousse, J. Wang and A. Saipulla, “Strong barrier coverage of wireless sensor networks,” in Proc. of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing(MobiHoc),2008.

[10] H. Du, Z. Zhang, W. Wu, L. Wu and K. Xing, “Constant-approximation for optimal data aggregation with physical interference,” Int’l J. of Global Optimization 56(4): 1653-1666, 2013.

[11] H. Du, Panos M. Pardalos, W. Wu, L. Wu, “Maximum lifetime connected coverage with two active-phase sensors,” Int’l J. of Global Optimization 56(2): 559-568, 2013.

[12] J.Z. Du, K. Wang, H. Liu and D.K. Guo, “Maximizing the lifetime of k-discrete barrier coverage using mobile sensors,” IEEE Sensors Journal, vol.13, no.12, December 2013.