Sink-Connected Barrier Coverage Optimization for Wireless Sensor Networks

Yung-Liang Lai and Jehn-Ruey Jiang

National Central University
Jhongli, 32001, Taiwan
945402011@cc.ncu.edu.tw jrjiang@csie.ncu.edu.tw

***Abstract*—This paper addresses the sink-connected barrier coverage optimization problem, which is concerned with how to select randomly deployed sensor nodes of a wireless sensor network (WSN) to reach two optimization goals: (1) to maximize the degree of barrier coverage by the minimum number of detecting nodes, and (2) to make the detecting nodes sink-connected by the minimum number of forwarding nodes. The detecting nodes are those for detecting intruders crossing the boundary of a monitored region. On detecting intruders, they send intruding event notifications to one of the sink nodes with the help of the forwarding nodes relaying. An algorithm, called optimal node selection algorithm (ONSA), is proposed for solving the problem on the basis of the maximum flow minimum cost algorithm. We perform simulations for ONSA and compare the results with those of a related algorithm, the global determination algorithm (GDA). The simulation results show that ONSA is better than GDA in terms of the number of nodes required in constructing sink-connected barrier coverage.**

Keywords-Wireless sensor networks; Barrier coverage; Maximum flow minimum cost algorithm; Sink connectivity

I. INTRODUCTION

A *wireless sensor network (WSN)* consists of a large number of sensor nodes with the capabilities of sensing, computing, storing, and communicating data. Each sensor can sense physical phenomena, such as light, temperature, sound, vibration, or electromagnetic field strength, and can transmit sensed data to one or more *sink nodes* through a multiple-hop transmission link. WSNs are self-organizing in the sense that they can be formed without human intervention, adapt to sensor failure and degradation, and react to task changes. They have wide applications like battlefield surveillance, environment monitoring, and so on. Some recent research uses WSNs to establish a *virtual barrier* of sensor nodes for detecting intruders crossing a protected area boundary, such as coastlines, national borders [9], and battlefield boundaries [11].

The barrier coverage problem is concerned with how to deploy WSN sensor nodes to form sensor barrier coverage for detecting intruders crossing a boundary. Several studies [2, 3, 4, 9, 10, 11] address the problem. They try to measure the quality of barrier coverage and/or to design schemes to achieve high-quality barrier-coverage in WSNs. In general, the quality of barrier coverage is measured by the *degree*. A WSN is said to form *k-degree barrier coverage* (or *k-barrier coverage*, for short) if any intruder crossing the barrier is to be detected by at least *k* sensors. To take the WSN in Fig. 1 as an example, it forms 2-barrier coverage and its degree of barrier coverage is 2. This is because any intruder will be detected by at least two different sensor nodes when the intruder crosses the WSN from one border side to the opposite side.

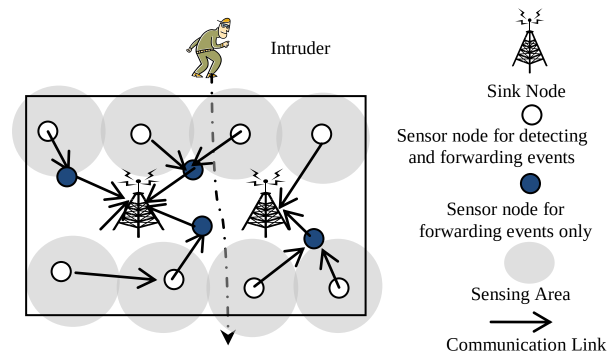


Figure 1. Sink-connected 2-barrier coverage

To the best of our knowledge, no earlier research addresses the barrier coverage problem with both the considerations that the sensor nodes should be connected to the sink node and that the number of the sensor nodes is minimized. In this paper, we take both considerations into account and propose an algorithm to solve the s*ink-connected barrier coverage optimization problem*, which is concerned with how to select nodes from sensor nodes of a randomly deployed WSN to reach following two goals:

Goal 1: Maximizing the degree of barrier coverage using the minimum number of detecting nodes

Goal 2: Minimizing the number of forwarding nodes to make detecting nodes sink-connected

Randomly deployed nodes can be selected to be *detecting nodes* or *forwarding nodes*. The former is selected to be active for detecting intruders and sending *intruding event notifications* towards the sink nodes, and the latter, for forwarding the notifications. It is noted that unselected

 ISBN: 978-1-61208-140-3

nodes can remain inactive to save energy and detecting nodes can also help forward event notifications of other detecting nodes. The first goal is to maximize the degree of WSN barrier coverage while minimizing the number of the detecting nodes. The second goal is to make detecting nodes *sink-connected* (i.e., to make sure that every detecting node can find a path to send intruding event notification to a sink node) by adding a minimum number of forwarding nodes. When the number of forwarding nodes decreases, the collision probability goes down, and the energy consumption in transmissions is thus reduced.

The remainder of this paper is organized as follows. In Sections 2, we introduce some related work. In Section 3, we present the network model and problem definitions. The proposed algorithm is described in Section 4 and simulation results are reported in Section 5. Finally, conclusion is drawn in Section 6.

II. RELATED WORK

The notion of barrier coverage was first introduced by Gage in [6] aiming at sensor-based surveillance of the boundary barrier to minimize the probability of undetected enemy penetration through the boundary barrier. In [10], Liu and Towsley defined *detectability* to be the probability that no path exists for an object to penetrate a barrier. They also characterize the detectability and showed that if the sensor node density is below a critical density, an intruder can almost surely find a path to cross the barrier without being detected. Wang and Cao in [12] also studied how to construct barrier coverage to monitor moving objects in camera sensor networks.

Kumar et al. [9] defined the notion of *k*-barrier coverage for precisely representing a WSN's ability of intruder detection. A WSN is said to have the *k*-barrier coverage property if any intruder crossing the barrier is detected successful by at least *k* sensor nodes. The authors developed theorems and proposed a centralized scheme using the maximum flow algorithm to determine whether a belt boundary region is *k-barrier covered* or not. Besides, they showed that the individual sensors cannot locally decide whether a network can form barrier coverage due to the lack of the global information. Unlike the algorithm in [9] that returns either true or false (0 or 1) for measuring the quality of barrier coverage, the method proposed by Chen et al. in [4] returns a non-binary value for the measurement. They also proposed a method to identify local regions whose qualities do not reach the desired level of quality.

Chen et al. [3] proposed a localized algorithm that guarantees the detection of intruders whose trajectory is confined to a slice of a belt boundary region. Saipulla et al. in [11] studied the barrier coverage of WSNs with line-based deployment, in which sensors are deployed along a line (e.g., sensors are dropped from an aircraft along a given path). Balister et al. [2] estimated the reliable node density that achieves barrier coverage with s-t connectivity in a thin strip with finite length, where *s-t connectivity* means that a connected path exists between the two far ends of the thin strip.

III. NETWORK MODEL AND PROBLEM FORMULATION

In this section, we first describe the network model and then formulate the sink-connected barrier coverage optimization problem to be solved in this paper.

A. Network Model

Consider a WSN consisting of many sensor nodes and few sink nodes, in which sensor nodes are to form a virtual sensor barrier for monitoring a belt region to detect and send intruding events to one of the sink nodes. The sensor nodes are assumed to be randomly deployed; for example, they can be dropped from an aircraft as described in [11]. Each sensor node is equipped with a sensing module with a fixed sensing range to sense intruders and a communication module with a fixed communication range to communicate with other sensor nodes or sink nodes. Initially, a sensor node performs a bootstrapping task to pin point its location, discover its neighboring nodes, and report its information, such as the identification and the location, to one of the sink nodes. The sink nodes are more powerful than sensor nodes. They have more energy, memory, computing power and communication capacity. They can communicate with each other and with sensor nodes; they can also communication with the backend system, which is assumed to have unlimited power supply and enormous computing power to gather all WSN nodes' information and perform the optimization computation.

Let Vs and Vk denote the set of sensor nodes and the set of sink nodes, respectively. Below, we define a *coverage graph* Gc to represent the sensing area coverage relationships of nodes. Moreover, we define a *transmission graph* Gt to represent the nodes' wireless transmission reachability relationships.

(1) Coverage Graph

A coverage graph Gc(Vs$\cup${S,T}, Ec) is an undirected graph, in which Vs is the sensor node set, Ec is the edge set, and {S,T} are two virtual nodes. The edge set Ec represents the sensing area coverage overlap relationships. For two nodes Ni and Nj in Vs, there exists an edge (Ni, Nj) in Ec if Ni's coverage and Nj's coverage have overlap. As shown in Fig. 2, the monitored belt region has the outer side, inner side and lateral sides. Intruders are supposed to cross the belt region from outer side to inner side. The virtual nodes S and T are associated the lateral sides; an edge (Ni, S) or (Ni,

 ISBN: 978-1-61208-140-3

T) exists in Ec if Ni's sensing area overlap either lateral side. Fig. 2 shows the coverage graph Gc of the WSN with 8 sensor nodes N_1,..,N_8, which are represented by solid circles. Note that the gray shades around the solid circles represent the sensing areas of sensor nodes.

Now, we can define the traversable paths in Gc. A traversal path of a coverage graph Gc(Vs∪{S,T}, Ec) is defined to be a path starting from S, going along edges in Ec through nodes in Vs, and stopping at T. Note that a coverage graph is similar to a flow network [1] and a traversable path is similar to a *flow* in the network. In the flowing context, the terms "traversable path" and "flow" will be used alternatively. The coverage graph and its traversal paths are very useful for measuring the degree of barrier coverage. By the theorems developed in [9], a WSN forms *k*-barrier coverage if and only if there exist *k* node-disjoint traversable paths in the coverage graph of the WSN. In the WSN of Fig. 2, there are two node-disjoint traversable paths S-N_1-N_2-N_3-N_4-T and S-N_5-N_6-N_7-N_8-T in the WSN coverage graph, so the WSN forms 2-barrier coverage.

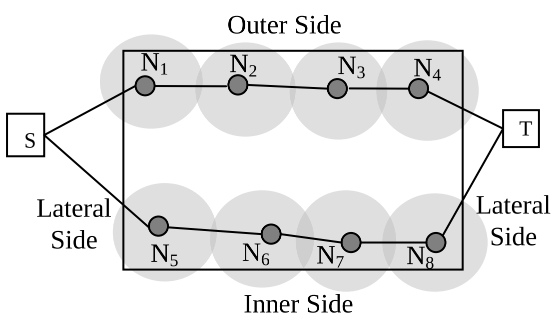


Figure 2. A WSN coverage graph and its 2 node-disjoint traversable paths

(2) Transmission Graph

A transmission graph Gt(Vs∪Vk, Et) is a directed graph, where Vs is the sensor node set, Vk is the sink node set, and Et is an arc (or directed edge) set to represent transmission relationships. Note that we may use "edge" to stand for "arc" in the following context. That is, the two terms are used interchangeably when there is no ambiguity. For two nodes Ni and Nj in Vs, it exists an arc <Ni, Nj> in Et if the node Ni can successfully transmit data (or events) to node Nj over a direct wireless link. Based on the transmission graph Gt of a WSN, we can define the sink-connected property for a set of sensor nodes as follows. For the WSN with the transmission graph Gt(Vs∪Vk, Et), a set S (S⊆Vs) of sensor nodes is *sink-connected* if there exists a path for each node in S going through only nodes in S to reach a node in Vk. For example, for the WSN in Fig. 3 consisting of 14 sensor nodes N_1,…,N_{14} and 2 sink nodes K_1 and K_2, the node sets {N_4}, {N_{11}}, {N_1, N_9}, {N_2, N_3, N_{11}}, {N_4, N_7, N_8, N_{13}}, {N_9,…,N_{13}} and {N_1,...,N_{13}} all satisfy the sink-connected property. However, the node sets {N_1}, {N_6, N_{11}} and {N_1,…,N_8} do not satisfy the sink-connected property.

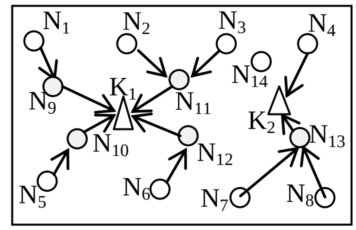


Figure 3. A WSN transmission graph with partial arcs for illustrating the sink-connected property

B. *Sink-Connected Barrier Coverage Optimization Problem*

The objective of the sink-connected barrier coverage optimization problem is to maximize the degree of barrier coverage by selecting the minimum number of nodes, while keeping the selected nodes sink-connected. Below, we formally define the problem.

Given a WSN with the coverage graph Gc(Vs∪{S,T},Ec) and transmission graph Gt(Vs∪Vk, Et), the sink-connected barrier coverage optimization problem is to achieve the following two goals.

Objective 1: To find a minimum sensor node set Vr such that the number of node-disjoint traversable paths of Vr is maximized

Objective 2: To find a minimum forwarding node set Vt such that (Vr∩Vt=∅) and (Vr∪Vt) satisfies the sink-connected property.

According to the above definition, a solution to the sink-connected barrier coverage optimization problem will return two node sets Vr and Vt. We can assume the nodes in Vr as detecting nodes to detect intruding events, and assume the nodes in Vt as forwarding nodes to forward events to one of the sink node. Certainly, since the detecting nodes remain active, they can also forward the intruding events sent by other detecting nodes. In reality, the detecting nodes can form barrier coverage with the highest degree. The solution is optimal in the sense that the degree of barrier coverage is maximized, while the number of detecting nodes and the number of forwarding nodes are both minimized. The solution is also practical in the sense that the detecting nodes are sink-connected with the help of forwarding nodes.

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-140-3

IV. OPTIMAL NODE SELECTION ALGORITHM (ONSA)

In this section, we propose an algorithm, called *optimal node selection algorithm (ONSA)*, to solve the sink-connected barrier coverage optimization problem. Given the sensor nodes Vs, sink nodes Vk, coverage relationship Ec, and transmission relationship Et, ONSA can find the detecting node set Vr and the forwarding node set Vt. ONSA has three main tasks. The first task is to construct the coverage graph Gc and then perform the node-disjoint transformation to generate the graph Gc* such that Gc* is a flow network [5]. The second task is to find the minimum cost maximum flow in Gc*. The third task is to construct the transmission graph Gt based on Gc* and to find a flow plan by executing the maximum flow minimum cost algorithm. The nodes selected in the flow plan will be activated for constructing sink-connected barrier coverage. The details of ONSA are described below.

Optimal Node Selection Algorithm (ONSA)

Input: Vs, Vk, Ec, Et

Output: Vr and Vt

Step 1: Construct a coverage graph Gc(Vs$\cup${S,T}, Ec), where S and T are virtual nodes. Each edge in Ec is associated with one capacity and zero cost.

Step 2: Execute *node-disjoint transformation* to transfer Gc into the new graph Gc*.

Step 3: Execute the maximum flow minimum cost algorithm on Gc* to decide the *minimum cost flow plan* (FP_{MinCt}), and let node set Vr contain the selected nodes in FP_{MinCt}.

Step 4: Construct a transmission graph Gt(Vs$\cup$Vk, Et). Add a virtual source node S and a virtual target node T into Gt.

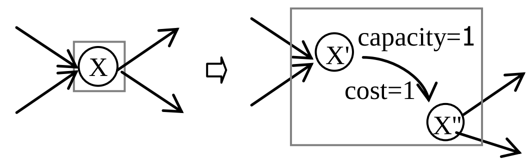
Step 5: For each node in Vr on graph Gt, add an edge from the virtual source node S it to. For each sink node, add an edge from it to the virtual target node.

Step 6: Execute *node-edge transformation* to convert Gt into Gt*.

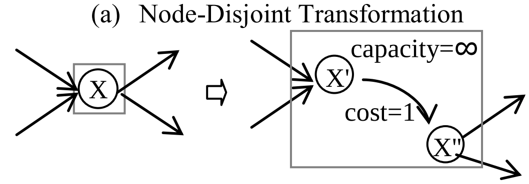
Step 7: Execute the maximum flow minimum cost algorithm to find the *minimum cost flow* (FP_T) on Gt*. Let Vb be the set of the nodes selected in FP_T.

Step 8: Set Vt=Vb-Vr and return Vr and Vt

In step 1, ONSA constructs a coverage graph Gc with a virtual node S and a virtual node T. In step 2, ONSA executes the *node-disjoint transformation* to convert each node with multiple inbound flows and multiple outbound flows into another form. The purpose of the transformation is to make the generated graphs node-disjoint. In the transformation, a node X will be transformed into a pair of nodes X' and X" with one capacity and one unity cost (refer to Fig. 4 (a)).



(a) Node-Disjoint Transformation



(b) Node-Edge Transformation

Figure 4. Two transformations of ONSA

In step 3, ONSA executes the well-known maximum flow minimum cost algorithm, which has two procedures. The first procedure is to find a maximum flow by executing the maximum flow algorithm. The second procedure is to execute the minimum cost flow algorithm. The readers are referred to [5] and [8] for the procedure details. After executing the two procedures on Gc*, a set Vr of detecting nodes is obtained. Since the flow is maximized, the number of node-disjoint traversable paths is also maximized. Moreover, since the cost is minimized, the number of nodes in Vr is also minimized.

In step 4, ONSA constructs a transmission graph Gt(Vs$\cup$Vk, Et) and adds a virtual source node S and a virtual target node T into Gt. In step 5, ONSA adds arcs for connecting the virtual source node S and the nodes in Vr. Each newly added arc is associated with Cost=0 and Capacity=∞.

In step 6, ONSA converts each node (excluding S and T) into two nodes with one arc of Cost=1 and Capacity=∞ by the *node-edge transformation*. Please refer to the Fig. 4(b).

In step 7, ONSA executes the maximum flow minimum cost algorithm on Gt* to decide the minimum cost flow (FP_T). In this step, the nodes containing in FP_T will be added into the node set Vb. Since FP_T has the minimum cost, the number of nodes in Vb will also be minimized.

In step 8, ONSA returns Vr as the set of detecting nodes and returns Vt=Vb-Vr as the set of forwarding nodes.

Below, we take the WSN in Fig. 3 as an example to illustrate the execution of ONSA. In step 1, a coverage graph Gc will be constructed. After step 2, the nodes with multiple inbound flows and multiple outbound flows are transformed by the node-disjoint transformation. The transformation results are shown in Fig. 5.

 ISBN: 978-1-61208-140-3

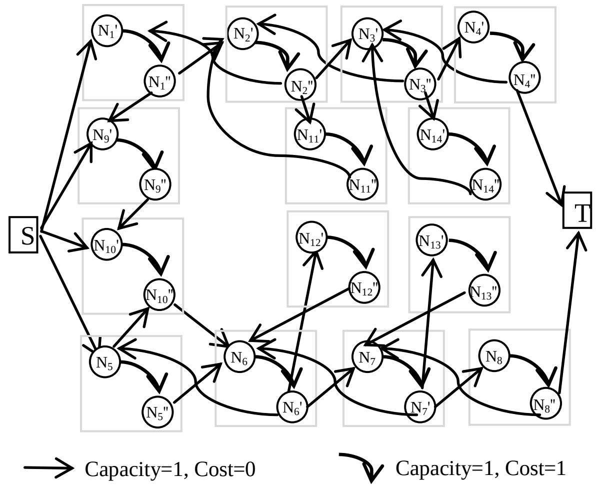


Figure 5. An example of the node-disjoint transformation

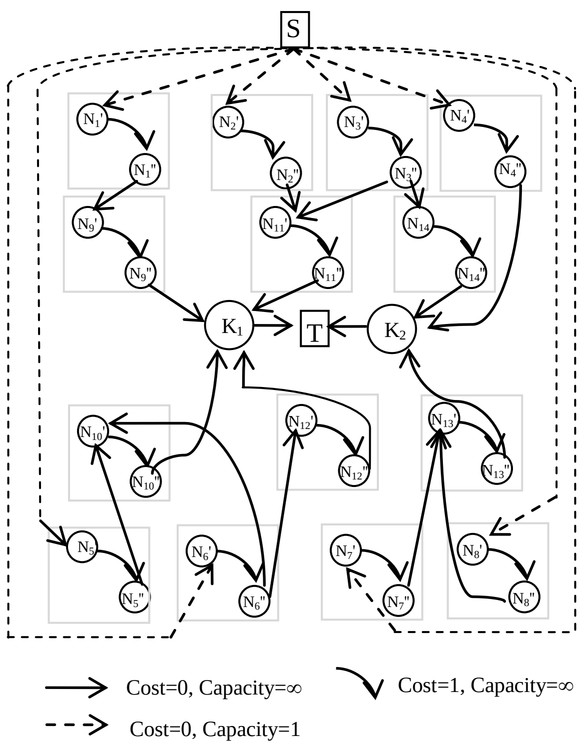


Figure 6. An example of the node-edge transformation

In step 3, a flow plan is decided by the maximum flow minimum cost algorithm. In this example, the maximum number of flows is two. In step 4, the graph transmission graph Gt is constructed and virtual nodes S and T are added into Gt. In step 5, a new arc is added between the virtual source node S and every node selected in step 3 (i.e., every node in Vr). Moreover, a new arc is added between every sink node and the virtual target node.

In step 6, node-edge transformation is performed to generate Gt*, as shown in Fig. 6. In step 7, the maximum flow minimum cost algorithm is executed to obtain Vb. In this example, Vb is $\{N_1, N_2, \ldots, N_{13}\}$, which is a set containing the nodes selected in FP_T. In step 8, $Vr=\{N_1,\ldots,N_8\}$ and $Vt=\{N_9,\ldots,N_{13}\}$ are returned by ONSA, where Vr contains detecting nodes to form 2-barrier coverage for detecting intruding events and Vt contains forwarding nodes to forward events sent by detecting nodes to one of the sink nodes (i.e., either K_1 or K_2). Fig. 7 shows the execution results returned by ONSA.

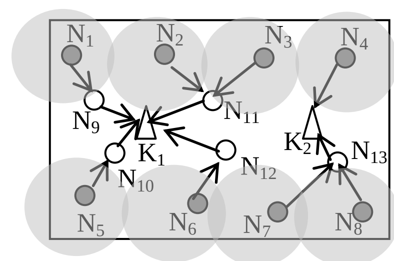


Figure 7. The execution result of ONSA, where $N_1,\ldots,N_8$ are selected as detecting nodes and $N_9,\ldots,N_{13}$ are selected as forwarding nodes

The time complexity of ONSA is dominated by Step 3 and Step 7, which execute the maximum flow minimum cost algorithm on Gc* and Gt*, respectively. The maximum flow minimum cost algorithm is actually the combination of the Edmonds-Karp algorithm [5], which is of $O(V\cdot E^2)$ time complexity for a graph of vetext set V and edge set E, and the minimum cost flow algorithm (MinCostFlow) [8], which is of $O(V\cdot E^2\cdot\log(V))$ time complexity. The time complexity of ONSA is thus $O(Vc^*\cdot E^2c^*\cdot\log(Vc^*) + Vt^*\cdot E^2t^*\cdot\log(Vt^*))$, where Vc* (resp., Vt*) is the size of the vertex set in Gc* (resp., Gt*) and Ec* (resp., Et*) is the size of the edge set in Gc* (resp., Gt*). To execute the optimization computation of ONSA will consume some compuation power and memory storage. Fortunatley, as we have mentioned earlier, ONSA is performed by the backend sytem, which is assumed to have unlimited power supply and enormous computing power. All the sensor nodes in the WSN only need to collaborate to deliver/forward their local information required by ONSA to the sink nodes, which in turn forward the information to the backend system. In other words, ONSA does not impose much computation and memory consumption on normal sensor nodes.

V. SIMULATION

To demonstrate the advantages of ONSA, our proposed algorithm, we conduct simulation experiments. We also compare the simulation results with those of the *global determination algorithm (GDA)*, which is proposed in [9], for determining the degree of barrier coverage by using the maximum flow algorithm. Since GDA does not consider the

 ISBN: 978-1-61208-140-3

sink-connected property, we again use the maximum flow algorithm for GDA to select extra sensor nodes to serve as forwarding nodes to make GDA satisfy the property. In this way, GDA and ONSA can both achieve the highest-degree property and sink-connectivity property of barrier coverage.

We develop a simulator based on the MATLAB software [13] to solve the optimization problem. The simulations are conducted in the following settings. The sensing area coverage radius is 10m. The wireless transmission radius is equal to the coverage radius. All the sensors are randomly deployed by different number of sensors in a rectangle-shaped area of 100m x 10m. The number of sink nodes is 2 and their locations are respectively at ($\frac{100}{3}$m, 5m) and ($\frac{2\times100}{3}$m, 5m) relative to the left-most and lowest position of the rectangle.

We compare ONSA and GDA in terms of the number of nodes selected to achieve the highest-degreed and sink-connected barrier coverage. As shown in Fig. 8, ONSA selects fewer nodes than GDA for all cases. It implies that ONSA needs fewer nodes than GDA to achieve the highest-degreed and sink-connected barrier coverage.

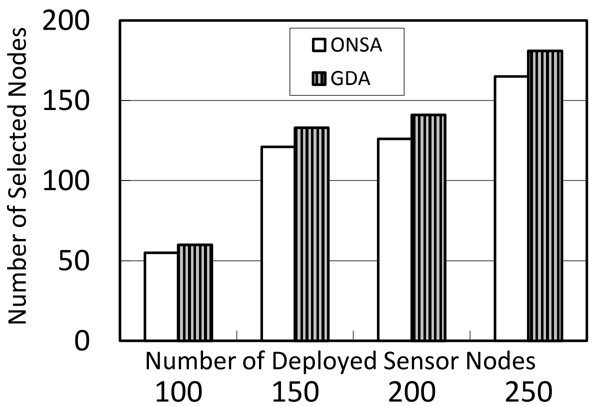


Figure 8. Comparisons of ONSA and GDA in terms of the number of selected nodes

VI. CONCLUSION

In this paper, we studied the sink-connected barrier coverage problem to achieve two goals: (1) to maximize the degree of barrier coverage using the minimum number of detecting nodes and (2) to minimize the number of forwarding nodes to hold the sink-connected property. To the best of our knowledge, this is the first paper to consider the sink-connected property and the barrier coverage quality optimization at the same time. An optimal network flow planning algorithm, called optimal node selection algorithm (ONSA), is proposed to solve the problem. ONSA is based on the well-known maximum flow minimum cost algorithm. We also perform simulation experiments for ONSA and a related algorithm called global determination algorithm (GDA), which uses the maximum flow algorithm to find out the maximum degree of barrier coverage and does not consider the sink-connected property. For the sake of comparison, the maximum flow algorithm is again used to make GDA satisfy the sink-connected property. The simulation results show that ONSA is better than GDA in terms of number of selected nodes.

In the future, we plan to study the optimization of barrier coverage with sink-node connectivity under the lifetime constraint. The lifetime is usually defined as the time span from the time of network deployment to the time when a certain fraction of sensor nodes run out of their energy. If the lifetime of a wireless sensor network is too short, it is likely that the network will soon be partitioned and fail to deliver the sensed data to sink nodes. The lifetime constraint is thus an important factor to be addressed.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, Network Flows: Theory, *Algorithms, and Applications*, Prentice Hall, Upper Saddle River, New Jersey, 1993.

[2] P. Balister, B. Bollobas, A. Sarkar, and S. Kumar, “Reliable density estimates for coverage and connectivity in thin strips of finite length,” *Proceedings of ACM Mobicom*, pp. 75-86, 2007.

[3] A. Chen, S. Kumar, and T.-H. Lai, “Designing localized algorithms for barrier coverage,” *Proceedings of ACM Mobicom*, pp. 63-74, 2007.

[4] A. Chen, T.-H. Lai, and D. Xuan, “Measuring and guaranteeing quality of barrier-coverage in wireless sensor networks,” *Proceedings of The ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 421-430, 2008.

[5] T.-H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (Second Edition)*, MIT Press and McGraw-Hill, 2001.

[6] D. Gage. “Command control for many-robot systems,” *Proceedings of the Nineteenth Annual AUVS Technical Symposium (AUVS-92)*, pp. 28-34, 1992.

[7] M. R. Garey and D. S. Johnson, *Computer and Intractability*, W. H. Freeman and Co., 1979.

[8] A. V. Goldberg,” An efficient implementation of a scaling minimum-cost flow algorithm,” *Journal of Algorithm*, vol. 22, pp. 1-29, 1997.

[9] S. Kumar, T.-H. Lai, and A. Arora, “Barrier coverage with wireless sensors,” *Wireless Networks*, vol. 13, pp. 817–834, 2007.

[10] B. Liu and D. Towsley, “A study on the coverage of large-scale sensor networks,” in *The 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 475-483, 2004.

[11] A. Saipulla, C. Westphal, B. Liu and J. Wang, “Barrier Coverage of Line-Based Deployed Wireless Sensor Networks,” *IEEE INFOCOM 2009*, pp. 127-135, Brazil, 2009.

[12] Y. Wang, G. Cao, “Barrier Coverage in Camera Sensor Networks,” to be presented at *ACM MobiHoc’11*, Paris, France, 2011.

[13] MathWorks Inc, Matlab Software, URL: http://www.mathworks.com. (Accessed at January 8, 2011).