

Article

# An Effective Sensor Deployment Scheme that Ensures Multilevel Coverage of Wireless Sensor Networks with Uncertain Properties

Yu-Ning Chen, Wu-Hsiung Lin and Chiuyuan Chen \*

Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan; abc89051@gmail.com (Y.-N.C.); wuhsiunglin@nctu.edu.tw (W.-H.L.)

\* Correspondence: cychen@mail.nctu.edu.tw

Received: 10 February 2020; Accepted: 23 March 2020; Published: 25 March 2020



**Abstract:** The coverage problem is a fundamental problem for almost all applications in wireless sensor networks (WSNs). Many applications even impose the requirement of multilevel ( $k$ ) coverage of the region of interest (ROI). In this paper, we consider WSNs with uncertain properties. More precisely, we consider WSNs under the probabilistic sensing model, in which the detection probability of a sensor node decays as the distance between the target and the sensor node increases. The difficulty we encountered is that there is *no* unified definition of  $k$ -coverage under the probabilistic sensing model. We overcome this difficulty by proposing a “reasonable” definition of  $k$ -coverage under such a model. We propose a sensor deployment scheme that uses less number of deployed sensor nodes while ensuring good coverage qualities so that (i) the resultant WSN is connected and (ii) the detection probability satisfies a predefined threshold  $p_{th}$ , where  $0 < p_{th} < 1$ . Our scheme uses a novel “zone 1 and zone 1–2” strategy, where zone 1 and zone 2 are a sensor node’s sensing regions that have the highest and the second highest detection probability, respectively, and zone 1–2 is the union of zones 1 and 2. The experimental results demonstrate the effectiveness of our scheme.

**Keywords:** sensor deployment; sensor coverage problem; network planning; probabilistic sensing model; wireless sensor network

## 1. Introduction

A wireless sensor network (WSN) has various applications in health-care, smart home, security, environmental exploration, and the military [1,2]. A sensor node (or simply node) is the basic component of a WSN. A WSN usually consists of numerous nodes deployed in a region of interest (ROI). Two nodes can communicate with each other if each is within the transmission range of the other, in which case we say that there is a link between them or that they are neighbors. Each node is able to collect data and process information and communicate with neighboring nodes.

Among various issues in WSNs, the coverage problem and the connectivity problem have been regarded as crucial foundations because many applications rely on them. Surveys for the coverage problem can be found in [3,4]. A good sensor deployment strategy should consider both coverage and connectivity. Sensor deployment not only determines the cost of constructing the network but also affects how well the given ROI will be monitored. This paper assumes that each node’s sensing region is of a disk shape (see Figure 1a) and all nodes have the same sensing range  $r_s$  and communication range  $r_c$ .

Let  $u$  be a location in the ROI,  $s_i$  be a node in the WSN, and  $d(u, s_i)$  be the Euclidean distance between  $u$  and  $s_i$ . Most of the past researches use the binary sensing model [5–7], where nodes are assumed to be accurate in detecting targets within their sensing ranges. More precisely, under the

binary sensing model (see Figure 1b for an illustration), the detection probability  $p(u, s_i)$  of  $u$  by  $s_i$  is defined as

$$p(u, s_i) = \begin{cases} 1 & \text{if } d(u, s_i) \leq r_s, \\ 0 & \text{otherwise.} \end{cases}$$

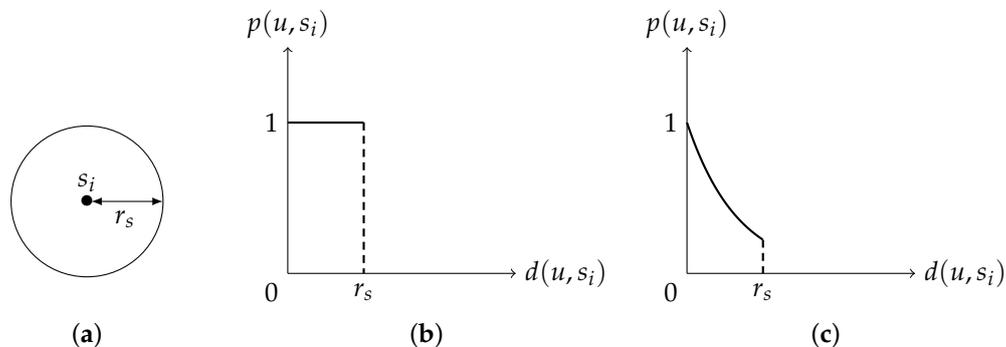


Figure 1. (a) Disk shape. (b) Binary sensing model. (c) Probabilistic sensing model.

In physical scenarios, the distance decay effect cannot be ignored. That is, the detection probability of a node decays as the distance between the target and the node increases. Based on this, the coverage problem under a more realistic model, called the probabilistic sensing model, has been investigated; see [7–11]. A survey for the coverage problem with uncertain properties can be found in [12]. Do notice that unlike the binary sensing model, which has a unified definition of  $p(u, s_i)$ , there is no unified definition of  $p(u, s_i)$  under the probabilistic sensing model; see Section 2 for details. We now give the definition of  $p(u, s_i)$  used in [7] and in this paper. Under the probabilistic sensing model (see Figure 1c for an illustration), the detection probability  $p(u, s_i)$  of  $u$  by  $s_i$  is defined as

$$p(u, s_i) = \begin{cases} e^{-\lambda d(u, s_i)} & \text{if } d(u, s_i) \leq r_s, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where  $\lambda$  is a sensor-dependent parameter.

Many real-world applications impose the requirement of multilevel  $k > 1$  coverage. For example,  $k \geq 2$  for military or surveillance applications,  $k \geq 3$  for positioning protocols using triangulation [13], conducting data fusion [14], and minimizing the impact of sensor failure [15]. Sensor deployment with multilevel ( $k$ ) coverage have been discussed in [7,15–17].

Under the binary sensing model, an ROI is said to be  $k$ -covered if every location in the ROI can be detected by at least  $k$  nodes (i.e., every location in the ROI is within  $k$  nodes' sensing regions). Unfortunately, to the best of our knowledge, there is no unified definition of  $k$ -coverage under the probabilistic sensing model. We now are ready to elaborate this issue. Let  $\mathcal{A} = \{s_1, s_2, \dots, s_n\}$  be the set of sensor nodes deployed in the ROI. In [7], a location  $u$  in the ROI is considered as  $k$ -covered if the probability that there are at least  $k$  nodes that can detect  $u$  is not smaller than a predefined threshold  $p_{th}$ , where  $0 < p_{th} < 1$ . More precisely, in [7], a location  $u$  in the ROI is said to be  $k$ -covered if

$$\prod_{s_i \in \mathcal{A}'} p(u, s_i) \geq p_{th}, \text{ for some } \mathcal{A}' \subseteq \mathcal{A} \text{ with } |\mathcal{A}'| = k. \quad (2)$$

For clarity, we call this definition of  $k$ -coverage the  $k$ -threshold coverage. In [18], a location  $u$  in the ROI is said to be  $k$ -covered if

$$\sum_{s_i \in \mathcal{A}} p(u, s_i) \geq k. \quad (3)$$

For clarity, we call this definition of  $k$ -coverage the  $k$ -expectation coverage. In both [7] and [18], the ROI is said to be  $k$ -covered if every location  $u$  inside the ROI is  $k$ -covered. The  $p(u, s_i)$  used in [18]

is defined as (4), which is a generalization of (1) and is introduced in Section 2. By (3), the event of  $u$  detected by  $s_i$  is independent of the event of  $u$  detected by  $s_j$  for  $j \neq i$ . Reference [18] proves that the ROI is  $k$ -covered by  $\mathcal{A}$  if, among all locations inside the ROI, the minimum expected number of nodes is  $k$  (there is a typo and  $k$  should be *at least*  $k$ ).

Three main considerations in WSNs' coverage are: maximizing coverage quality, maximizing network lifetime, and minimizing the number of deployed nodes. The coverage quality and lifetime are two conflict factors with respect to energy consumption. Reference [12] points out that the  $k$ -coverage requirement makes the problem more sophisticated because more nodes are needed; deploying less nodes by decreasing the overlap region and prolonging the network lifetime become complicated. The  $k$ -expectation coverage [18] has the drawback that the user cannot specify his/her preference for the threshold  $p_{th}$ , meaning that the user cannot specify their desired coverage quality. Moreover, it is possible that the entire ROI is regarded as  $k$ -covered but some locations inside it are detected by a lot of nodes each with a small detection probability. The  $k$ -threshold coverage [7] has the drawback that it tends to use too many sensor nodes; we give the calculation details in Section 2. In short, Wang and Tseng [7] first calculate  $r_{th}$  (notice that  $r_{th}$  is denoted as  $r_s^p$  in [7]) and then replace the original  $r_s$  with  $r_{th}$  in the deployment.

The objective of this paper is to propose a sensor deployment scheme to use less number of nodes while ensuring the following two important coverage qualities: (i) the resultant WSN is connected and (ii) the detection probability satisfies a predefined threshold  $p_{th}$ , where  $0 < p_{th} < 1$ . Although  $k$ -threshold coverage [7] achieves the same objective, we find that the nature of  $k$ -threshold coverage makes  $r_{th}$  tend to become *much smaller than* the original  $r_s$ . For example, suppose  $k = 3$  and  $m$  denotes meters. Then:

- If  $\lambda = 0.05$  and  $r_s = 30$  m, then  $r_{th}$  for  $p_{th} = 0.7, 0.8, 0.9$  are 2.377 m, 1.487 m, and 0.702 m, respectively.
- If  $\lambda = 0.08$  and  $r_s = 30$  m, then  $r_{th}$  for  $p_{th} = 0.7, 0.8, 0.9$  are 1.486 m, 0.929 m, and 0.439 m, respectively.

These very small  $r_{th}$ 's cause a large number of nodes to be deployed. Since the number of nodes is very large, the overall detection probability might be underestimated.

In this paper, we try to propose a "reasonable" solution to the  $k$ -coverage problem under the probabilistic sensing model. The main contributions of this paper are as follows.

- We propose a "reasonable" definition of  $k$ -coverage under the probabilistic sensing model, called the  **$k$ -layer coverage**, and propose a scheme to achieve  $k$ -layer coverage.
- We propose a novel "zone 1 and zone 1–2" strategy to fulfill  $k$ -layer coverage scheme and ensure good coverage quality. We propose an efficient algorithm to calculate the radius  $r_1$  of zone 1, which takes at most 18 iterations when error tolerance  $\epsilon = 10^{-6}$ .
- Experimental results shows that our  $k$ -layer coverage scheme indeed uses less sensor nodes, thereby demonstrating the effectiveness of our scheme.

Our  $k$ -layer coverage scheme partitions the nodes into  $k$  subsets, each forming one layer of coverage, and ensures that for every location  $u$  inside the ROI, the detection probability for  $u$  by nodes in "each layer" is not smaller than  $p_{th}$ . In particular, our  $k$ -layer coverage scheme ensures that every location  $u$  inside the ROI is within zone 1 of at least  $k$  nodes and within zone 1–2 of at least another  $2 \cdot k$  nodes (zone 1 and zone 1–2 are defined in Section 4). When coverage quality is considered, our  $k$ -layer coverage scheme provides a good solution. Different from the  $k$ -threshold coverage [7], which replaces the original  $r_s$  with  $r_{th}$ , our  $k$ -layer coverage replaces the original  $r_s$  with  $r_1$ . We prove that as long as  $\sqrt{3}r_1 < r_s$ , we have  $r_1 > k \cdot r_{th}$ , meaning that our  $k$ -layer coverage will use less nodes. When the number of nodes is considered, our  $k$ -layer coverage scheme also provides a good solution.

The rest of this paper is organized as follow. Section 2 introduces the related works. Section 3 gives preliminaries, assumptions, and objectives. Section 4 gives the basics of the  $k$ -layer coverage.

Section 5 gives our  $k$ -layer coverage scheme. Section 6 illustrates experimental results. Concluding remarks are given in the final section.

### 2. Related Works

Recall that a good sensor deployment should consider both coverage and connectivity. Assuming that the ROI is a convex set, Zhang and Hou [19] investigate the relationship between coverage and connectivity and prove that  $r_c \geq 2r_s$  is both necessary and sufficient to ensure that coverage implies connectivity. With such a proof, one can then focus only on the coverage problem. As long as  $r_c \geq \sqrt{3}r_s$ , nodes can be deployed according to the regular triangular lattice pattern (triangular pattern for short; see Figure 2) so that both coverage and connectivity can be ensured [7,20]. Using the triangular pattern, neighboring nodes will be regularly separated by a distance of  $\sqrt{3}r_s$ . A deployment using the triangular pattern is sometimes called an *optimal deployment* since it is asymptotically optimal in terms of the number of nodes needed to achieve full coverage of the ROI.

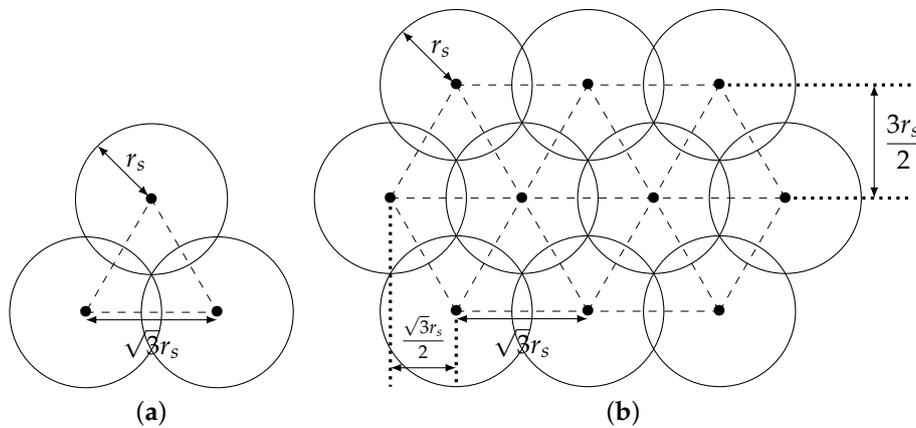


Figure 2. Triangular pattern: (a) three neighboring nodes, (b) the general case.

In some applications,  $r_c \geq 2r_s$  or  $r_c \geq \sqrt{3}r_s$  may not hold. Wang and Tseng [7] therefore consider an arbitrary relationship between  $r_s$  and  $r_c$ , thus relaxing the limitations of existing results. In [7], for the the binary sensing model, two solutions to achieve  $k$ -coverage are proposed: the naive duplicate placement scheme (duplicate scheme for short) and the interpolating placement scheme (interpolating scheme for short). The idea of the duplicate scheme is to use a good sensor placement method to ensure 1-coverage and connectivity and then duplicate  $k$  nodes on each designated location. However, since the duplicate scheme may result in some regions in the ROI having a much higher coverage levels than  $k$ , the interpolation scheme is therefore being proposed to “reuse” these regions to generate a multilevel coverage. When  $k = 1$  or  $k = 2$  or  $(k \geq 3 \text{ and } r_c > \frac{2+\sqrt{3}}{3}r_s)$ , it is found that the interpolation scheme will not save nodes compared to the duplicate scheme, thereby adopting the duplicate scheme. For clarity, we summarize the schemes used in [7] in Table 1. Notice that the interpolation schemes used in the  $(r_c \leq \frac{\sqrt{3}}{2}r_s)$ -case and the  $(\frac{\sqrt{3}}{2}r_s < r_c \leq \frac{2+\sqrt{3}}{3}r_s)$ -case are different.

Table 1. Scheme used by [7]: “ $k$ ” means “to make the ROI  $k$ -covered”, “duplicate” means “duplicate scheme”, “interpolation” means “interpolation scheme”, and “tri. pattern” means “triangular pattern”.

$k$	Case $r_c \leq \frac{\sqrt{3}}{2}r_s$	Case $\frac{\sqrt{3}}{2}r_s < r_c \leq \frac{2+\sqrt{3}}{3}r_s$	Case $\frac{2+\sqrt{3}}{3}r_s < r_c < \sqrt{3}r_s$	Case $r_c \geq \sqrt{3}r_s$
$k = 1$	duplicate	duplicate	duplicate	duplicate and tri. pattern
$k = 2$	duplicate	duplicate	duplicate	duplicate and tri. pattern
$k \geq 3$	interpolation	interpolation	duplicate	duplicate and tri. pattern

Wang and Tseng [7] adapt the schemes of the binary sensing model to the probabilistic sensing model. Set  $(\star) = (k \geq 3 \text{ and } r_c \geq \sqrt{3}r_s)$  for convenience. We now use the  $(\star)$ -case as an illustration

to show how [7] performs the adaption. According to Table 1 (shown in Section 2), under the binary sensing model, [7] will use the duplicate scheme and triangular pattern in the (★)-case. Under the probabilistic sensing model, in the (★)-case, [7] first calculates  $r_{th}$  (notice that  $r_{th}$  is denoted as  $r_s^p$  in [7]). Then, [7] replaces the original  $r_s$  with  $r_{th}$  in the deployment to ensure that every location inside the ROI is  $k$ -covered under the probabilistic sensing model. Since the duplicate scheme places  $k$  nodes on each designated location, [7] calculates  $r_{th}$  by

$$\prod_{s_i \in \mathcal{A}'} p(u, s_i) \approx e^{-k\lambda r_{th}}$$

and

$$e^{-k\lambda r_{th}} \geq p_{th} \Rightarrow r_{th} \leq \frac{\ln p_{th}}{-k\lambda}$$

where  $u$  is a location in the ROI having the minimum  $k$ -covered probability and  $u$  is detected by a set  $\mathcal{A}'$  of  $k$  nodes placed at a designated location with distance  $r_{th}$  to  $u$ .

Let  $u, s_i, d(u, s_i)$ , and  $p(u, s_i)$  be defined as in Section 1. In [11,18,21,22], four parameters  $(\lambda, \beta, r, r_e)$  are used to specify a probabilistic sensing model and  $p(u, s_i)$  is defined as

$$p(u, s_i) = \begin{cases} 1 & \text{if } d(u, s_i) \leq r - r_e, \\ e^{-\lambda(d(u, s_i) - (r - r_e))^\beta} & \text{if } r - r_e < d(u, s_i) \leq r + r_e, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where  $\lambda$  and  $\beta$  are sensor-dependent parameters (see Figure 3). That is, if  $d(u, s_i) \leq r - r_e$ , then a target at  $u$  will definitely be detected by  $s_i$ ; if  $d(u, s_i) > r + r_e$ , then the detection probability will be too small and will be totally ignored. If  $r - r_e < d(u, s_i) \leq r + r_e$ , then the behavior of the detection probability obeys the function  $e^{-\lambda(d(u, s_i) - (r - r_e))^\beta}$ . By taking  $\beta = 1$  and  $r = r_e = \frac{r_s}{2}$ , (4) coincides with (1).

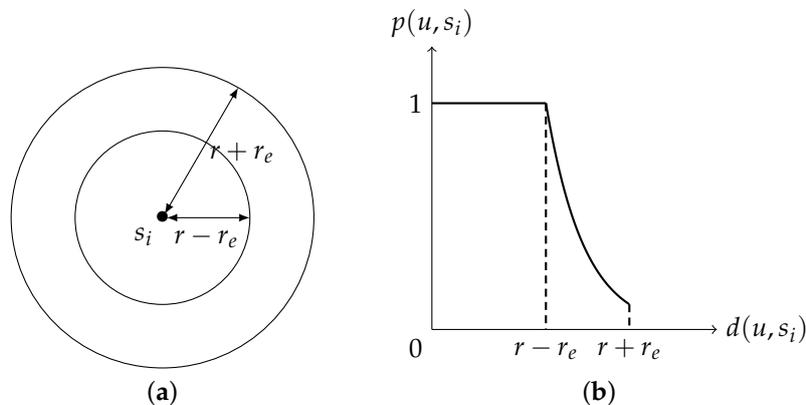


Figure 3. (a)  $r - r_e$  and  $r + r_e$ . (b) The generalized probabilistic sensing model.

Notice that besides the binary sensing model and the probabilistic sensing model, some researchers consider the evidence-based sensor coverage model and use the theory of belief functions to solve the coverage problem; see [23–26]. Notice that [23–26] consider 1-coverage and their ROI is assumed as a two- or three-dimensional grid of points. In this paper, we consider  $k$ -coverage and our ROI contains every location inside it. Some other references related to  $k$ -coverage can also be found in [27–31]. Before ending this section, for clarity, we summarize our most related works in Table 2.

**Table 2.** Definitions of  $k$ -coverage:  $\mathcal{A} = \{s_1, s_2, \dots, s_n\}$  is the set of sensor nodes deployed in the ROI and “det. prob.” means “detection probability”.

$k$ -Coverage by	Definition	det. prob. $p$ Given by
$k$ -expectation [18]	$\sum_{s_i \in \mathcal{A}} p(u, s_i) \geq k$	Equation (4)
$k$ -threshold [7]	$\prod_{s_i \in \mathcal{A}'} p(u, s_i) \geq p_{th}$ , for some $\mathcal{A}' \subseteq \mathcal{A}$ with $ \mathcal{A}'  = k$	Equation (1)
$k$ -layer (this paper)	see Definition 1	Equation (1)

### 3. Preliminaries, Assumptions, and Objectives

Typical types of coverage are target coverage, area coverage, and barrier coverage. The purpose of target coverage is to cover (monitor) a set of specific targets (or points), that of area coverage is to cover the entire ROI, and that of barrier coverage is to detect intruders who intend to cross a long belt region. In a WSN, nodes are deployed in ad-hoc or pre-planned manner. In pre-planned deployment, nodes are placed to designated locations in the ROI so that fewer nodes can provide satisfactory coverage with lower network maintenance and management cost. Area coverage is usually more difficult and can be solved by ad-hoc or pre-planned deployment, depending on the application scenario.

This paper discusses “area coverage” by “pre-planned deployment”; the following assumptions are made:

1. The ROI is of a rectangular shape.
2. Sensor nodes are homogeneous, i.e., with the same sensing range  $r_s$  and communication range  $r_c$ .
3.  $r_c \geq \sqrt{3}r_s$ .
4. The application that we are interested in allows nodes to be deployed in pre-planned manner.
5. The detection probability,  $p(u, s_i)$  of  $u$  by  $s_i$ , used in this paper is (1).

The objective of this paper is to propose a “reasonable” definition of  $k$ -coverage under the probabilistic sensing model and to develop a sensor deployment scheme that uses less number of nodes while ensuring the following two coverage qualities: (i) the resultant WSN is connected and (ii) the detection probability satisfies a predefined threshold  $p_{th}$ , where  $0 < p_{th} < 1$ .

### 4. Basics of the $k$ -Layer Coverage

#### 4.1. The Definition of $k$ -Layer Coverage

**Definition 1.** Given an ROI and a set  $\mathcal{A}$  of a finite number of nodes deployed in the ROI, a location  $u$  in the ROI is  $k$ -layer covered (or  $k$ -covered for short) if  $\mathcal{A}$  can be partitioned into  $k$  subsets  $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_k$  such that

$$1 - \prod_{s_{i,j} \in \mathcal{A}_i} (1 - p(u, s_{i,j})) \geq p_{th}, \text{ for each } i = 1, 2, \dots, k \quad (5)$$

for a predefined threshold  $p_{th}$ , where  $0 < p_{th} < 1$ . The ROI is  $k$ -layer covered (or  $k$ -covered for short) if every location inside it is  $k$ -layer covered.

For convenience, we call each  $\mathcal{A}_i$  one “layer” and call the above coverage the  $k$ -layer coverage. See Figure 4 for an illustration. The  $k$ -layer coverage ensures that the detection probability for  $u$  contributed by each layer (i.e., each  $\mathcal{A}_i$ ) is not smaller than  $p_{th}$ . Thus,  $u$  is  $k$ -layer covered if and only if it is 1-layer covered by each  $\mathcal{A}_i, i = 1, 2, \dots, k$ . Consequently, the ROI is  $k$ -layer covered if and only if it is 1-layer covered by each  $\mathcal{A}_i, i = 1, 2, \dots, k$ .

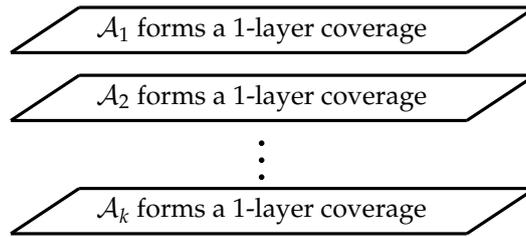


Figure 4. The concept of  $k$ -layer coverage.

4.2. “Zone 1 and Zone 1–2” Strategy

We regard each node’s sensing region as the composition of three concentric circular zones: zone 1, zone 2, and zone 3, as shown in Figure 5. We denote by  $r_1$  the radius of zone 1 and  $r_2$ , the outer contour radius of zone 2, where  $r_1 < r_2 \leq r_s$ . We take  $r_2 = \sqrt{3}r_1$  so that the calculations can be greatly simplified. Zone 1 has a higher detection probability than zone 2 which further has a higher detection probability than zone 3. Zone 3 might be empty and our coverage scheme will not take it into account. For convenience, denote by zone 1–2 the union of zones 1 and 2.

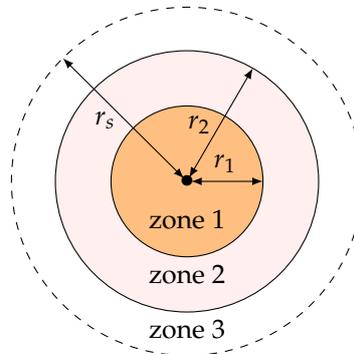


Figure 5. Zone 1, zone 2, zone 3,  $r_1$ ,  $r_2$ , and  $r_s$ . We take  $r_2 = \sqrt{3}r_1$ .

The following theorem states the “zone 1 and zone 1–2” strategy used in our  $k$ -layer coverage scheme, whose proof is in Section 5.

**Theorem 1.** Our  $k$ -layer coverage scheme ensures that for each  $\mathcal{A}_i$ , every location inside the ROI is within zone 1 of one node in  $\mathcal{A}_i$  and within zones 1–2 of at least another two nodes in  $\mathcal{A}_i$ .

4.3. An Algorithm for Calculating the Radius  $r_1$  of Zone 1

To use the “zone 1 and zone 1–2” strategy, we need to calculate  $r_1$ . Let  $u$  be an arbitrary location inside the ROI. By Theorem 1, there exist three distinct nodes  $s_a, s_b$ , and  $s_c$  in  $\mathcal{A}_i$  such that  $u$  is within zone 1 of  $s_a$  and zones 1–2 of  $s_b$  and  $s_c$ . By (5), we have

$$\begin{aligned}
 1 - \prod_{s_{i,j} \in \mathcal{A}_i} (1 - p(u, s_{i,j})) &\geq 1 - \underbrace{(1 - p(u, s_a))}_{\text{node } s_a} \underbrace{(1 - p(u, s_b))}_{\text{node } s_b} \underbrace{(1 - p(u, s_c))}_{\text{node } s_c} \\
 &\geq 1 - (1 - e^{-\lambda r_1})(1 - e^{-\lambda r_2})(1 - e^{-\lambda r_2}).
 \end{aligned}$$

On the one hand, to minimize the number of nodes,  $r_1$  should be as large as possible. On the other hand, to ensure coverage quality,  $r_1$  cannot be too large because the following inequality is needed:

$$1 - (1 - e^{-\lambda r_1})(1 - e^{-\lambda r_2})(1 - e^{-\lambda r_2}) \geq p_{\text{th}}. \tag{6}$$

Therefore, we should find the largest possible  $r_1$  that satisfies (6). The following algorithm finds such an  $r_1$  and we now briefly describe the idea behind our algorithm.

Since the detection probability of any target outside  $r_s$  is defined as zero,  $r_1$  and  $r_2$  must satisfy  $r_1 < r_2 \leq r_s$ . Since we take  $r_2 = \sqrt{3}r_1$ , the largest possible  $r_1$  is therefore  $\frac{r_s}{\sqrt{3}}$ . Substituting  $r_1$  by  $\frac{r_s}{\sqrt{3}}$  into (6), we have

$$1 - (1 - e^{-\lambda \frac{r_s}{\sqrt{3}}})(1 - e^{-\lambda r_s})(1 - e^{-\lambda r_s}) \geq p_{\text{th}}.$$

Thus, our  $k$ -layer coverage has an unavoidable lower bound  $p_{\text{th}}^{\text{min}}$  of  $p_{\text{th}}$ , which is defined as

$$p_{\text{th}}^{\text{min}} = 1 - (1 - e^{-\lambda \frac{r_s}{\sqrt{3}}})(1 - e^{-\lambda r_s})(1 - e^{-\lambda r_s}).$$

This unavoidable lower bound  $p_{\text{th}}^{\text{min}}$  is due to the system assumption that the detection probability of any target outside  $r_s$  is zero. If  $p_{\text{th}} \leq p_{\text{th}}^{\text{min}}$ , then our algorithm enlarges  $p_{\text{th}}$  to be  $p_{\text{th}}^{\text{min}}$  and returns  $\frac{r_s}{\sqrt{3}}$  for  $r_1$ . In the remaining part of this paragraph, we assume  $p_{\text{th}} > p_{\text{th}}^{\text{min}}$  and show how to calculate  $r_1$  when  $p_{\text{th}} > p_{\text{th}}^{\text{min}}$ . Since

$$1 - (1 - e^{-\lambda r_2})^3 \leq 1 - (1 - e^{-\lambda r_1})(1 - e^{-\lambda r_2})(1 - e^{-\lambda r_2}) \leq 1 - (1 - e^{-\lambda r_1})^3,$$

we have

$$1 - (1 - e^{-\lambda r_2})^3 \leq p_{\text{th}} \leq 1 - (1 - e^{-\lambda r_1})^3. \quad (7)$$

For convenience, denote by  $r_1^*$  the the largest possible  $r_1$  that satisfies (6) under the assumption that  $p_{\text{th}} > p_{\text{th}}^{\text{min}}$ . We say that a value is an  $\epsilon$ -approximation of  $r_1^*$  if it satisfies (6) and satisfies a given error tolerance  $\epsilon$ . The following method finds either  $r_1^*$  or an  $\epsilon$ -approximation of  $r_1^*$  by using (7). Substituting  $\sqrt{3}r_1$  for  $r_2$  in (7), we have

$$1 - (1 - e^{-\lambda \sqrt{3}r_1})^3 \leq p_{\text{th}} \leq 1 - (1 - e^{-\lambda r_1})^3.$$

Thus

$$1 - e^{-\lambda r_1} \leq (1 - p_{\text{th}})^{\frac{1}{3}} \leq 1 - e^{-\lambda \sqrt{3}r_1}.$$

Hence

$$(e^{-\lambda r_1})^{\sqrt{3}} = e^{-\lambda \sqrt{3}r_1} \leq 1 - (1 - p_{\text{th}})^{\frac{1}{3}} \leq e^{-\lambda r_1}.$$

By using  $1 - (1 - p_{\text{th}})^{\frac{1}{3}} \leq e^{-\lambda r_1}$ , we can obtain an approximate  $r_1$ . Since  $1 - (1 - p_{\text{th}})^{\frac{1}{3}}$  is a lower bound of  $e^{-\lambda r_1}$ , let

$$\text{lower} = 1 - (1 - p_{\text{th}})^{\frac{1}{3}}.$$

By using  $e^{-\lambda r_1} \leq (1 - (1 - p_{\text{th}})^{\frac{1}{3}})^{\frac{1}{\sqrt{3}}}$ , we can obtain another approximate  $r_1$ . Since  $(1 - (1 - p_{\text{th}})^{\frac{1}{3}})^{\frac{1}{\sqrt{3}}}$  is an upper bound of  $e^{-\lambda r_1}$ , let

$$\text{upper} = (1 - (1 - p_{\text{th}})^{\frac{1}{3}})^{\frac{1}{\sqrt{3}}}.$$

By keeping

$$\text{lower} \leq e^{-\lambda r_1} \leq \text{upper},$$

when our algorithm stops, it finds either  $r_1^*$  or an  $\epsilon$ -approximation of  $r_1^*$ . Our full algorithm is now shown in Algorithm 1.

**Algorithm 1**

**Input:**  $p_{th}$ ,  $r_s$ , and  $\lambda$ , where  $p_{th}$  is the threshold,  $r_s$  is the sensing range, and  $\lambda$  is the sensor-dependent parameter given in (1).

**Output:**  $\frac{r_s}{\sqrt{3}}$  if  $p_{th} \leq p_{th}^{\min}$ ;  $r_1^*$  or an  $\epsilon$ -approximation of  $r_1^*$  if  $p_{th} > p_{th}^{\min}$ .

```

1: if  $p_{th} \leq p_{th}^{\min}$  then  $r_1 \leftarrow \frac{r_s}{\sqrt{3}}$ 
2: else
3:    $found \leftarrow false$ ;
4:    $\epsilon \leftarrow 10^{-6}$ ;
5:    $lower \leftarrow 1 - (1 - p_{th})^{\frac{1}{3}}$ ;
6:    $upper \leftarrow (1 - (1 - p_{th})^{\frac{1}{3}})^{\frac{1}{\sqrt{3}}}$ ;
7:   while  $((found = false) \text{ and } (upper - lower) \geq \epsilon)$  do
8:      $mid \leftarrow \frac{lower + upper}{2}$ ;
9:      $val \leftarrow 1 - (1 - mid)(1 - mid^{\sqrt{3}})(1 - mid^{\sqrt{3}})$ ;
10:    if  $(val = p_{th})$  then  $found \leftarrow true$ ;
11:    else if  $(val < p_{th})$  then  $lower \leftarrow mid$ ;
12:    else  $upper \leftarrow mid$ ;
13:    end if
14:  end while
15:  if  $(found = true)$  then
16:     $r_1 \leftarrow \frac{\ln mid}{-\lambda}$ ; //  $r_1^*$ 
17:  else
18:     $r_1 \leftarrow \frac{\ln upper}{-\lambda}$ ; // an  $\epsilon$ -approximation of  $r_1^*$ 
19:  end if
20: end if
21: return  $r_1$ ;

```

**Theorem 2.** If  $p_{th} \leq p_{th}^{\min}$ , then Algorithm 1 returns  $r_1 = \frac{r_s}{\sqrt{3}}$  in  $O(1)$  time. If  $p_{th} > p_{th}^{\min}$ , then Algorithm 1 returns either  $r_1^*$  or an  $\epsilon$ -approximation of  $r_1^*$  for  $\epsilon = 10^{-6}$  after at most 18 iterations.

**Proof.** The first statement is obvious. Assume that  $p_{th} > p_{th}^{\min}$  and consider the second statement. Then, Algorithm 1 returns either  $r_1 = \frac{\ln mid}{-\lambda}$  or  $r_1 = \frac{\ln upper}{-\lambda}$ . Consider the former case. In this case,  $mid = e^{-\lambda r_1}$ . By lines 9~10,  $1 - (1 - mid)(1 - mid^{\sqrt{3}})(1 - mid^{\sqrt{3}}) = p_{th}$ . Thus  $r_1$  is exactly  $r_1^*$ . Now, consider the latter case. In this case,  $r_1 < r_1^*$ . Hence

$$1 - (1 - e^{-\lambda r_1})(1 - e^{-\lambda \sqrt{3} r_1})(1 - e^{-\lambda \sqrt{3} r_1}) > 1 - (1 - e^{-\lambda r_1^*})(1 - e^{-\lambda \sqrt{3} r_1^*})(1 - e^{-\lambda \sqrt{3} r_1^*}) = p_{th}.$$

Thus  $r_1$  satisfies (6). The execution of the while-loop ensures that  $r_1$  satisfies the error tolerance  $\epsilon$ . Therefore  $r_1$  is an  $\epsilon$ -approximation of  $r_1^*$ . We now prove that when  $\epsilon = 10^{-6}$ , Algorithm 1 takes at most 18 iterations. Set  $x = 1 - (1 - p_{th})^{\frac{1}{3}}$  for easy writing. Since  $0 < p_{th} < 1$ , we have  $0 < x < 1$ . By line 5, initially  $lower = x$ . By line 6, initially  $upper = x^{\frac{1}{\sqrt{3}}}$ . Therefore, initially  $upper - lower = x^{\frac{1}{\sqrt{3}}} - x$ . Let  $f(x) = x^{\frac{1}{\sqrt{3}}} - x$ . The function  $f(x)$  achieves its maximum value when  $f'(x) = \frac{1}{\sqrt{3}} x^{\frac{\sqrt{3}-3}{3}} - 1 = 0$ , which occurs when

$$x = \sqrt[3]{\frac{3}{\sqrt{3}-3}} = 3^{\frac{1}{2} \cdot \frac{3}{\sqrt{3}-3} \cdot \frac{\sqrt{3}-3}{\sqrt{3}+3}} = 3^{\frac{3+\sqrt{3}}{-4}}.$$

Note that  $f(3^{\frac{3+\sqrt{3}}{4}}) \doteq 0.199572 < 0.2$ . Moreover, each iteration of the while-loop cuts the search space by half. Thus after the execution of  $i$ -th iteration of the while-loop,

$$(upper - lower) < f(3^{\frac{3+\sqrt{3}}{4}}) \times 2^{-i} < 0.2 \times 2^{-i}.$$

When  $0.2 \times 2^{-i} < \epsilon$  (i.e., when  $i > \log_2 \frac{0.2}{\epsilon}$ ), Algorithm 1 terminates its while-loop. Since  $\epsilon = 10^{-6}$ , we have  $\log_2 \frac{0.2}{\epsilon} \doteq 17.61 < 18$ . Thus Algorithm 1 takes at most 18 iterations.  $\square$

The following corollary follows from the proof of the above theorem.

**Corollary 1.** For an arbitrary error tolerance  $\epsilon$ , Algorithm 1 takes at most  $\lceil \log_2 \frac{0.2}{\epsilon} \rceil$  iterations. In particular, when  $\epsilon = 10^{-x}$ , where  $x$  is a positive integer, Algorithm 1 takes at most  $\lceil (x - 1) \log_2 10 + 1 \rceil$  iterations.

Suppose  $m$  denotes meters. Algorithm 1 obtains:

- If  $\lambda = 0.05$ ,  $r_s = 30m$ , then  $r_1$  for  $p_{th} = 0.7, 0.8, 0.9$  are 15.685m, 12.391m, and 8.749m, respectively.
- If  $\lambda = 0.08$ ,  $r_s = 30m$ , then  $r_1$  for  $p_{th} = 0.7, 0.8, 0.9$  are 9.801m, 7.743m, and 5.468m, respectively.

To handle the probabilistic sensing model, Wang and Tseng [7] replace the original  $r_s$  with  $r_{th} = \frac{\ln p_{th}}{-k\lambda}$  in their deployment. To handle the probabilistic sensing model, we replace the original  $r_s$  with  $r_1$ . The relationship between  $r_1$  and  $r_{th}$  is given in the following lemma.

**Lemma 1.** If  $\sqrt{3}r_1 < r_s$  and  $1 - (1 - e^{-\lambda r_1})(1 - e^{-\lambda\sqrt{3}r_1})(1 - e^{-\lambda\sqrt{3}r_1}) = p_{th}$ , then  $r_1 > k \cdot r_{th}$ .

**Proof.** Since  $r_{th} = \frac{\ln p_{th}}{-k\lambda}$ , it is true that  $e^{-\lambda k r_{th}} = p_{th}$ . Now assume that  $r_1$  satisfies the assumption of this lemma. Then,

$$\begin{aligned} 1 - (1 - e^{-\lambda k r_{th}})(1 - e^{-\lambda\sqrt{3}k r_{th}})(1 - e^{-\lambda\sqrt{3}k r_{th}}) &> 1 - (1 - e^{-\lambda k r_{th}}) = e^{-\lambda k r_{th}} = p_{th} \\ &= 1 - (1 - e^{-\lambda r_1})(1 - e^{-\lambda\sqrt{3}r_1})(1 - e^{-\lambda\sqrt{3}r_1}). \end{aligned}$$

Hence  $r_1 > k \cdot r_{th}$ .  $\square$

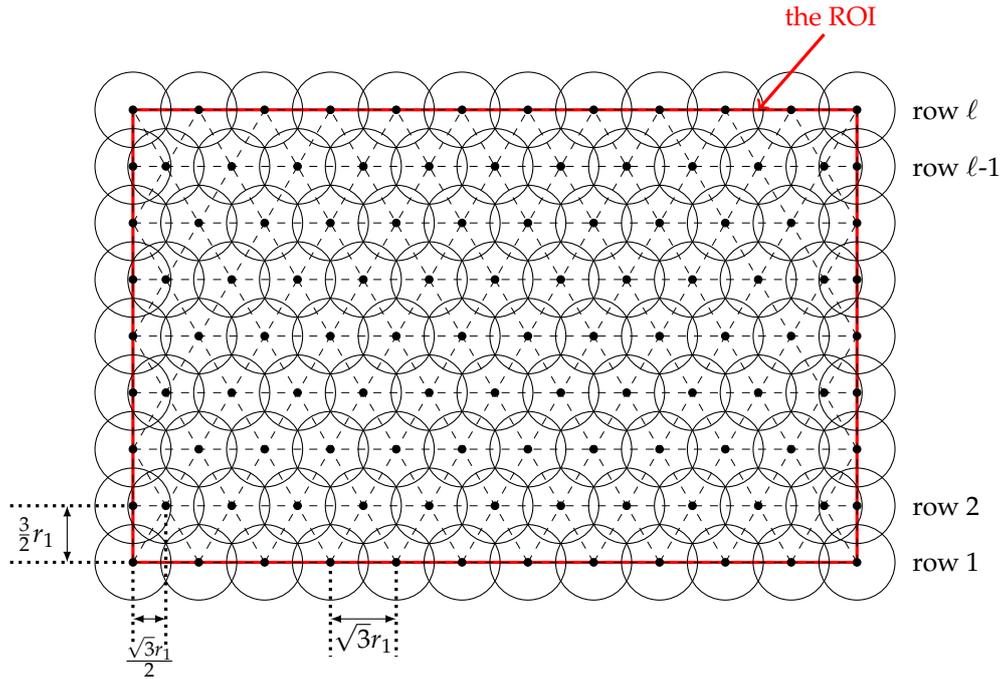
## 5. Our $k$ -Layer Coverage Scheme

### 5.1. The $k = 1$ Case

Our 1-layer coverage scheme deploys nodes according to a “pseudo” triangular pattern deployment, in which neighboring nodes will be regularly separated by a distance of  $\sqrt{3}r_1$  (i.e.,  $r_2$ ) except that some of them will be separated by a distance of  $\leq \frac{\sqrt{3}r_1}{2}$  or  $\leq \frac{3r_1}{2}$  and these exceptions only occur at the boundary of the ROI. See Figure 6 for an illustration.

Let  $L$  and  $H$  be the length and the height of the ROI, respectively. We assume that the ROI's lower left corner is located at coordinate  $(0, 0)$ . Then, the coordinates (i.e., locations) of nodes will be calculated row-by-row, with row 1 containing  $(0, 0)$ , and row  $\ell$  containing the upper boundary of the ROI. Our full 1-layer coverage scheme is shown in Algorithm 2. Since all nodes have to be within the ROI, we may need to adjust the locations of the following nodes:

- the *first* node in each even-numbered row (handled by line 6 of Algorithm 2),
- the *last* node in each row (handled by line 9 of Algorithm 2), and
- *all* nodes in row  $\ell$  (handled by lines 11–18 of Algorithm 2).



**Figure 6.** Sensor placement in 1-layer coverage scheme; notice that only zone 1 of each node is shown in this figure. This ROI has  $\sqrt{3}r_1 \mid L$  and  $\frac{3r_1}{2} \mid H$  and therefore no adjustment for rows 1, 3,  $\dots$ ,  $\ell$ .

---

#### Algorithm 2 sha cheng

---

**Input:** The  $r_1$  derived by Algorithm 1, the length  $L$  of the ROI, and the height  $H$  of the ROI.

**Output:** The designated locations of nodes.

```

1:  $r_2 \leftarrow \sqrt{3}r_1$ ;  $\ell \leftarrow \lceil \frac{2H}{3r_1} \rceil + 1$ ;  $n_1 \leftarrow \lceil \frac{L}{r_2} \rceil + 1$ ;  $n_2 \leftarrow \lceil \frac{2L-r_2}{2r_2} \rceil + 2$ ;
2: for  $i = 1$  to  $\ell - 1$  do //row 1, row 2,  $\dots$ , row ( $\ell - 1$ )
3:   if ( $i$  is odd) then
4:     output the location  $(j \cdot r_2, \frac{3(i-1)}{2}r_1)$  for each  $j = 0$  to  $n_1 - 1$ ;
5:   else //  $i$  is even
6:     output the location  $(0, \frac{3(i-1)}{2}r_1)$ ; //location of the first node in this row
7:     output the location  $(\frac{2j+1}{2}r_2, \frac{3(i-1)}{2}r_1)$  for each  $j = 0$  to  $n_2 - 2$ ;
8:   end if
9:   output the location  $(L, \frac{3(i-1)}{2}r_1)$ ; //location of the last node in this row
10: end for
11: //row  $\ell$  (i.e., the last row)
12: if ( $\ell$  is odd) then
13:   output the location  $(j \cdot r_2, H)$  for each  $j = 0$  to  $n_1 - 1$ ;
14: else //  $\ell$  is even
15:   output the location  $(0, H)$ ; //location of the first node in row  $\ell$ 
16:   output the location  $(\frac{2j+1}{2}r_2, H)$  for each  $j = 0$  to  $n_2 - 2$ ;
17: end if
18: output the location  $(L, H)$ ; //location of the last node in row  $\ell$ 

```

---

Let  $\ell$ ,  $n_1$ , and  $n_2$  denote the number of rows, the number of nodes deployed in an odd-numbered row, and the number of nodes deployed in an even-numbered row by Algorithm 2, respectively. Three claims are made.

**Claim 1.**  $\ell = \left\lceil \frac{2H}{3r_1} \right\rceil + 1$ .

**Proof.** This claim follows from the fact that the distance between row  $i$  and row  $(i + 1)$ ,  $1 \leq i < \ell - 1$ , is exactly  $\frac{3}{2}r_1$ , and the distance between row  $(\ell - 1)$  and row  $\ell$  is  $\leq \frac{3}{2}r_1$ .  $\square$

**Claim 2.**  $n_1 = \left\lceil \frac{L}{r_2} \right\rceil + 1$ .

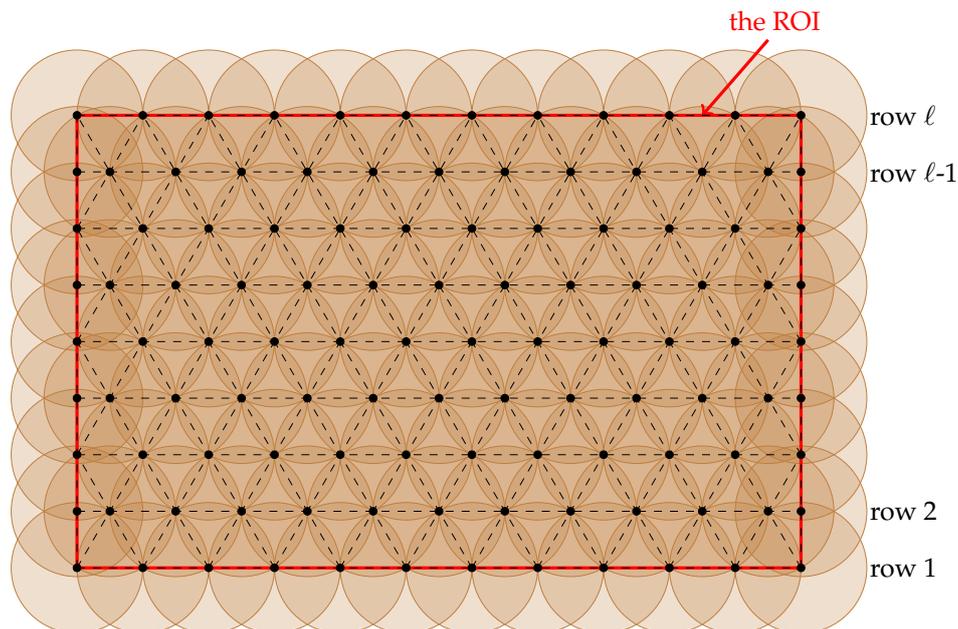
**Proof.** Consider an arbitrary odd-numbered row and let node 1, node 2, ..., node  $n_1$  be nodes in this row. Then, the distance between node  $i$  and node  $(i + 1)$ ,  $1 \leq i < n_1 - 1$ , is exactly  $r_2$ , and the distance between node  $(n_1 - 1)$  and node  $n_1$  is  $\leq r_2$ . Hence the claim.  $\square$

**Claim 3.**  $n_2 = \left\lceil \frac{2L - r_2}{2r_2} \right\rceil + 2$ .

**Proof.** Consider an arbitrary even-numbered row and let node 1, node 2, ..., node  $n_2$  be nodes in this row. Then, the distance between node 1 and node 2 is  $\frac{r_2}{2}$ , the distance between node  $i$  and node  $(i + 1)$ ,  $2 \leq i < n_2 - 1$ , is exactly  $r_2$ , and that between node  $(n_2 - 1)$  and node  $n_2$  is  $\leq \frac{r_2}{2}$ . Hence the claim.  $\square$

We now are ready to prove Theorem 1.

**Proof.** It can be easily observed from Figure 6 that every location inside the ROI is within zone 1 of at least one node. Thus, to prove this theorem, it suffices to prove that every location inside the ROI is within zones 1–2 of at least three nodes. This assertion can be easily verified by using Figure 7 and hence we have this theorem.  $\square$



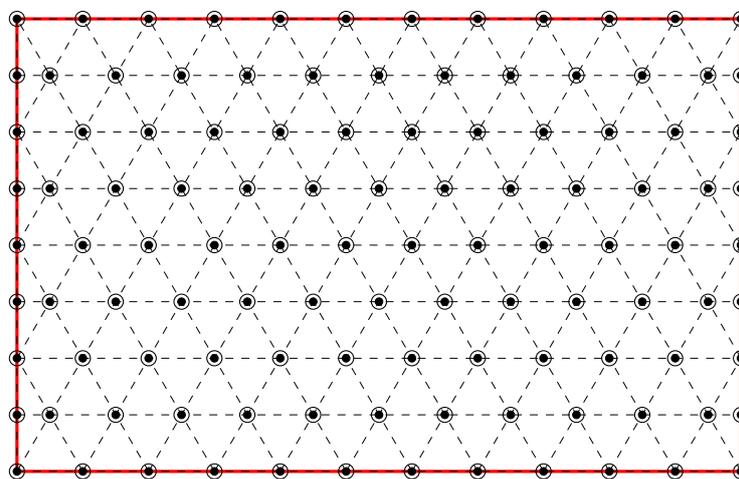
**Figure 7.** Sensor placement in 1-layer coverage scheme; notice that only zone 1–2 of each node is shown in this figure.

## 5.2. The General Case

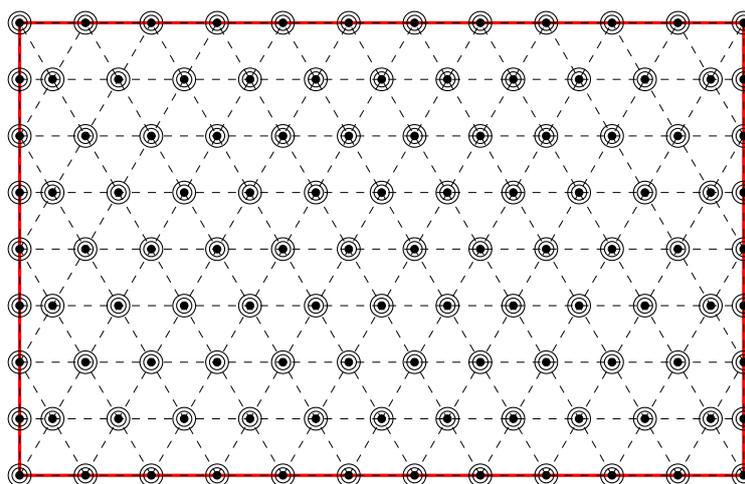
To achieve  $k$ -layer coverage for  $k > 1$ , we begin with the 1-layer coverage and duplicate  $k$  sensor nodes on each designated location of the 1-layer coverage. See Figure 8 for an illustration. See also Appendix A and Figure A1 for possible improvements. The following corollary follows from Claims 1~3.

**Corollary 2.** The number  $N$  of nodes deployed by our  $k$ -layer coverage scheme equals to

$$N = \begin{cases} k \cdot (n_1 + n_2) \lfloor \frac{\ell}{2} \rfloor + k \cdot n_1 & \text{if } \ell \text{ is odd,} \\ k \cdot (n_1 + n_2) \lfloor \frac{\ell}{2} \rfloor & \text{if } \ell \text{ is even.} \end{cases}$$



(a)



(b)

**Figure 8.** (a) 2-layer coverage. (b) 3-layer coverage.

### 6. Experimental Results

Our experimental results are obtained by using Visual C++ programming language under the environment of a 64-bit personal computer with win 10 operating system. Parameters used in our experimental results are listed in Table 3.

**Table 3.** Parameters used: m denote meters.

Size of the ROI	1000 m × 1000 m
sensing range $r_s$	30 m
error tolerance $\epsilon$ used in Algorithm 1	$10^{-6}$
sensor-dependent parameter $\lambda$ used in (1)	either 0.05 or 0.08

Our experimental results are shown in Tables 4–6. These results demonstrate  $r_{th}$  used in  $k$ -threshold coverage [7] and  $r_1$  used in our  $k$ -layer coverage. Recall that both  $r_{th}$  and  $r_1$  are served as a

substitution radius for the original sensing radius  $r_s$ . As can be observed in Tables 4–6,  $r_{th}$  is much smaller than  $r_s$  and  $r_1$  is more reasonable. Tables 4–6 also demonstrate the number of nodes required by  $k$ -threshold coverage [7] and our  $k$ -layer coverage. It is not difficult to see that the number of nodes required by  $k$ -threshold coverage [7] is quite large, especially when  $k$  is large. On the other hand, the number of nodes required by our  $k$ -layer coverage is much less.

**Table 4.** The number of nodes for  $k = 1$  coverage: “# nodes” means “the number of nodes required”.

		$\lambda = 0.05$			$\lambda = 0.08$				
		<i>k</i> -Threshold [7]		Our <i>k</i> -Layer		<i>k</i> -Threshold [7]		Our <i>k</i> -Layer	
$p_{th}$	$r_{th}$	# nodes	$r_1$	# nodes	$r_{th}$	# nodes	$r_1$	# nodes	
0.7	7.133	7790	15.685	1672	4.458	19,781	9.803	4200	
0.8	4.462	19,781	12.391	2640	2.789	50,369	7.744	6688	
0.9	2.107	87,768	8.749	5226	1.317	223,520	5.468	13,161	

**Table 5.** The number of nodes for  $k = 3$  coverage: “# nodes” means “the number of nodes required”.

		$\lambda = 0.05$			$\lambda = 0.08$				
		<i>k</i> -Threshold [7]		Our <i>k</i> -Layer		<i>k</i> -Threshold [7]		Our <i>k</i> -Layer	
$p_{th}$	$r_{th}$	# nodes	$r_1$	# nodes	$r_{th}$	# nodes	$r_1$	# nodes	
0.7	2.377	206,424	15.685	5016	1.486	526,500	9.803	12,600	
0.8	1.487	526,500	12.391	7920	0.929	1,343,811	7.744	20,064	
0.9	0.702	2,350,872	8.749	15,678	0.439	6,005,520	5.468	39,483	

**Table 6.** The number of nodes for  $k = 5$  coverage: “# nodes” means “the number of nodes required”.

		$\lambda = 0.05$			$\lambda = 0.08$				
		<i>k</i> -Threshold [7]		Our <i>k</i> -Layer		<i>k</i> -Threshold [7]		Our <i>k</i> -Layer	
$p_{th}$	$r_{th}$	# nodes	$r_1$	# nodes	$r_{th}$	# nodes	$r_1$	# nodes	
0.7	1.426	952,070	15.685	8360	0.891	2,433,750	9.803	21,000	
0.8	0.892	2,430,505	12.391	13,200	0.557	6,217,600	7.744	33,440	
0.9	0.421	10,881,000	8.749	26,130	0.263	27,857,950	5.468	65,805	

## 7. Concluding Remarks

This paper considers the multilevel ( $k$ ) coverage problem in WSNs with uncertain properties (i.e., under the probabilistic sensing model). We find that the nature of the previous  $k$ -threshold coverage [7] makes its substitution radius  $r_{th}$  tend to be much smaller than the original sensing range  $r_s$  and therefore will use too many nodes. We thus try to propose a more reasonable solution and yield the  $k$ -layer coverage. In  $k$ -layer coverage, each node’s sensing region is regarded as the composition of zone 1, zone 2, and zone 3. Although regarding each node’s sensing region as the composition of several zones has been proposed in the literature, to the best of our knowledge, we are the first ones to use the “zone 1 and zone 1–2” strategy to solve the probabilistic multilevel ( $k$ ) coverage problem. By using such a “zone 1 and zone 1–2” strategy, our  $k$ -layer coverage achieves a better coverage quality and uses less nodes. A preliminary version of this paper was in [32]; see also [33]. One future work for this research is to extend  $k$ -layer coverage to the probabilistic sensing model in (4) and to more realistic regions of interest. Another future work is to explore the relationship between coverage and energy consumption.

**Author Contributions:** Conceptualization, Y.-N.C. and C.C.; methodology, Y.-N.C. and C.C.; validation, Y.-N.C., W.-H.L. and C.C.; investigation, Y.-N.C., W.-H.L. and C.C.; data curation, Y.-N.C.; writing—original draft preparation, Y.-N.C. and C.C.; writing—review and editing, W.-H.L. and C.C.; supervision, C.C.; project administration, C.C.; funding acquisition, W.-H.L. and C.C. All authors have read and agreed to the published version of the manuscript.

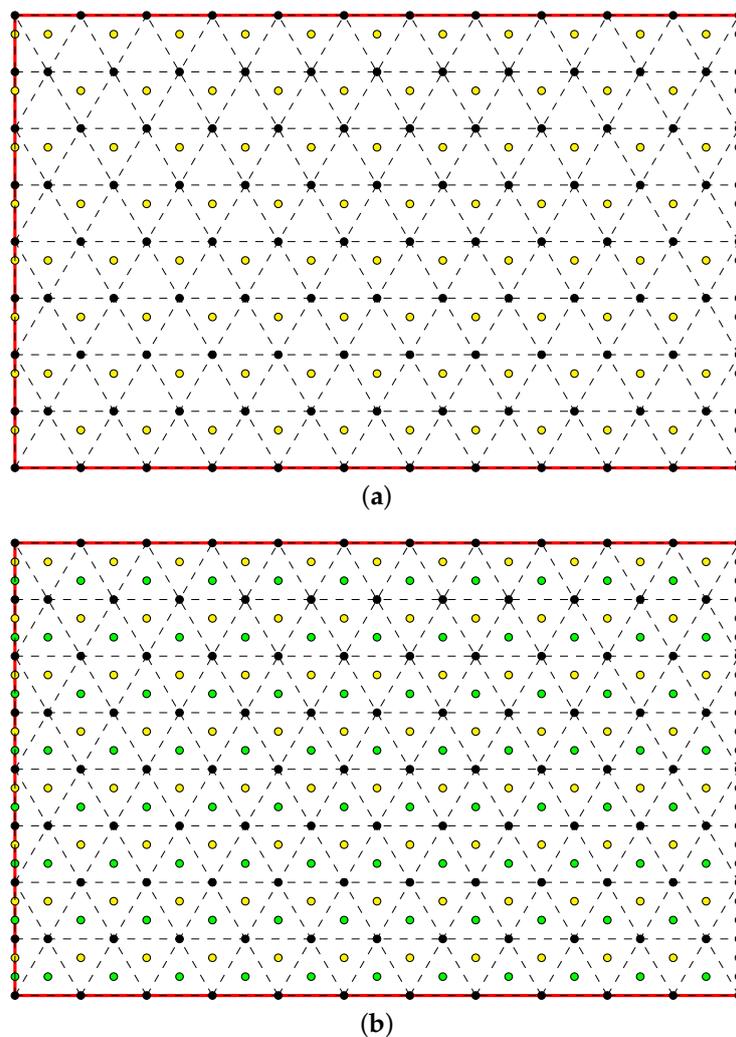
**Funding:** This research was partially supported by the Ministry of Science and Technology of Taiwan under grants MOST-108-2115-M-009-006 and MOST-108-2115-M-009-013.

**Acknowledgments:** The authors would like to thank Yu-Chee Tseng for his valuable comments that greatly improve the presentation of this paper. The authors would also like to express their deepest gratitude to the anonymous reviewers for their constructive comments and time spent to analyze this paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Appendix: Balanced $k$ -Layer Coverage

In Figure 8, some locations in the ROI get a much higher coverage than the other locations. We show how to obtain a more balanced  $k$ -layer coverage. Let  $\mathcal{L}$  denote the 1-layer coverage obtained by Algorithm 2. The balanced 1-layer coverage is  $\mathcal{L}$ . Instead of duplicating each node twice, the balanced 2-layer coverage is  $\mathcal{L}$  together with a shift  $\mathcal{L}'$  of  $\mathcal{L}$ ; see Figure A1a. Instead of duplicating each node thrice, the balanced 3-layer coverage is  $\mathcal{L}$  together with two shifts  $\mathcal{L}'$  and  $\mathcal{L}''$  of  $\mathcal{L}$ ; see Figure A1b. Notice that some nodes have to be added to  $\mathcal{L}'$  (respectively,  $\mathcal{L}''$ ) (all of these nodes are close to the ROI's boundary) so that  $\mathcal{L}'$  (respectively,  $\mathcal{L}''$ ) can ensure a 1-layer coverage; we omit the details. For  $k > 3$ , let  $q = \lfloor \frac{k}{3} \rfloor$  and  $r = k \% 3$ . The balanced  $k$ -layer coverage is applying the balanced 3-layer coverage  $q$  times and applying the balanced  $r$ -layer coverage once. For example, if  $k = 5$ , then apply the balanced 3-layer coverage once and the balanced 2-layer coverage once.



**Figure A1.** (a) The balanced 2-layer coverage. (b) The balanced 3-layer coverage.  $\mathcal{L}$  consists of all nodes colored black,  $\mathcal{L}'$  consists of all nodes colored yellow, and  $\mathcal{L}''$  consists of all nodes colored green.

## References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayichi, E. Wireless sensor network: A survey. *Comput. Netw.* **2002**, *38*, 393–422. [[CrossRef](#)]
2. Huo, H.; Xu, Y.; Zhang, H.; Chuang, Y.H.; Wu, T.C. Wireless-sensor-networks-based healthcare system: A survey on the view of communication paradigms. *Int. J. Ad Hoc Ubiquitous Comput.* **2011**, *8*, 135–154. [[CrossRef](#)]
3. Farsi, M.; Elhosseini, M.A.; Badawy, M.; Ali, H.A.; Eldin, H.Z. Deployment techniques in wireless sensor networks, coverage and connectivity: A survey. *IEEE Access* **2019**, *7*, 28940–28954. [[CrossRef](#)]
4. Wang, B. Coverage problems in sensor networks: A survey. *ACM Comput. Surv.* **2011**, *43*, 32. [[CrossRef](#)]
5. Mini, S.; Udgata, S.; Sabat, S. Sensor deployment and scheduling for target coverage problem in wireless sensor networks. *IEEE Sens. J.* **2014**, *14*, 636–644. [[CrossRef](#)]
6. Rebai, M.; Snoussi, H.; Hnaïen, F.; Khoukhi, L. Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks. *Comput. Oper. Res.* **2015**, *59*, 11–21. [[CrossRef](#)]
7. Wang, Y.C.; Tseng, Y.C. Distributed deployment scheme for mobile wireless sensor networks to ensure multilevel coverage. *IEEE Trans. Parallel Distrib. Syst.* **2008**, *19*, 1280–1294. [[CrossRef](#)]
8. Dhillon, S.S.; Chakrabarty, K. Sensor placement for effective coverage and surveillance in distributed sensor networks. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'03), New Orleans, LA, USA, 16–20 March 2003; pp. 1609–1614.
9. Yang, Q.; He, S.; Li, J.; Chen, J.; Sun, Y. Energy-efficient probabilistic area coverage in wireless sensor networks. *IEEE Trans. Veh. Technol.* **2015**, *64*, 367–377. [[CrossRef](#)]
10. Yun, Z.; Bai, X.; Xuan, D.; Lai, T.H.; Jia, W. Optimal deployment patterns for full coverage and  $k$ -connectivity wireless sensor networks. *IEEE/ACM Trans. Netw.* **2010**, *18*, 934–947.
11. Zou, Y.; Chakrabarty, K. Sensor deployment and target localization based on virtual forces. In Proceedings of the IEEE Infocom Conference (INFOCOM 2003), San Francisco, CA, USA, 30 March–3 April 2003; pp. 1293–1303.
12. Wang, Y.; Wu, S.; Chen, Z.; Gao, X.; Chen, G. Coverage problem with uncertain properties in wireless sensor networks: A survey. *Comput. Netw.* **2017**, *123*, 200–232. [[CrossRef](#)]
13. Nicules, D.; Nath, B. Ad-hoc positioning system (APS) using AoA. In Proceedings of IEEE International Conference on Computer Communications (INFOCOM'03), Dallas, TX, USA, 2003; pp. 1734–1743.
14. Klein, L.A. A Boolean algebra approach to multiple sensor voting fusion. *IEEE Trans. Aerosp. Electron. Syst.* **1993**, *29*, 317–327. [[CrossRef](#)]
15. Sun, T.; Chen, L.J.; Han, C.C.; Gerla, M. Reliable sensor networks for planet exploration. In Proceedings of the IEEE International Conference on Networking, Sensing, and Control (ICNSC'05), Tucson, AZ, USA, 19–22 March 2005; pp. 816–821.
16. Chakrabarty, K.; Iyengar, S.S.; Qi, H.; Cho, E. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Trans. Comput.* **2002**, *51*, 1448–1453. [[CrossRef](#)]
17. Huang, C.F.; Tseng, Y.C. The coverage problem in a wireless sensor network. *Mob. Netw. Appl.* **2005**, *10*, 519–528. [[CrossRef](#)]
18. Wang, H.L.; Chung, W.H. The generalized  $k$ -coverage under probabilistic sensing model in sensor networks. In Proceedings of the IEEE Wireless Communications and Networking Conference: Mobile and Wireless Networks (WCNC), Shanghai, China, 1–4 April 2012; pp. 1737–1742.
19. Zhang, H.; Hou, J.C. Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc Sens. Wirel. Netw.* **2005**, *1*, 89–124.
20. Kershner, R. The number of circles covering a set. *Am. J. Math.* **1939**, *61*, 665–671. [[CrossRef](#)]
21. Zou, Y.; Chakrabarty, K. Uncertainty-aware and coverage-oriented deployment for sensor networks. *J. Parallel Distrib. Comput.* **2004**, *64*, 788–798. [[CrossRef](#)]
22. Zou, Y.; Chakrabarty, K. A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks. *IEEE Trans. Comput.* **2005**, *54*, 978–991. [[CrossRef](#)]
23. Senouci, M.R.; Mellouk, A.; Oukhellou, L.; Aissani, A. Uncertainty-aware sensor network deployment. In Proceedings of the IEEE Globecom, Houston, TX, USA, 5–9 December 2011; pp. 1–5.
24. Senouci, M.R.; Mellouk, A.; Oukhellou, L.; Aissani, A. An evidence-based sensor coverage model. *IEEE Commun. Lett.* **2012**, *16*, 1462–1465. [[CrossRef](#)]

25. Senouci, M.R.; Mellouk, A.; Oukhellou, L.; Aissani, A. Efficient uncertainty-aware deployment algorithms for wireless sensor networks, In Proceedings of the IEEE Wireless Communications and Networking Conference: Mobile and Wireless Networks, Paris, France, 1–4 April 2012; pp. 2163–2167.
26. Senouci, M.R.; Mellouk, A.; Oukhellou, L.; Aissani, A. WSNs deployment framework based on the theory of belief functions. *Comput. Netw.* **2015**, *88*, 12–26. [[CrossRef](#)]
27. Ammari, H.M.; Das, S.K. Centralized and clustered  $k$ -coverage protocols for wireless sensor networks. *IEEE Trans. Comput.* **2012**, *61*, 118–133. [[CrossRef](#)]
28. Elhoseny, M.; Tharwat, A.; Farouk, A.; Hassanien, A.E.  $K$ -coverage model based on genetic algorithm to extend WSN lifetime. *IEEE Sens. Lett.* **2017**, *1*, 7500404. [[CrossRef](#)]
29. Liu, Q.  $K$ -coverage reliability evaluation for wireless sensor networks using 2 dimensional  $k / r \times s / m \times n$ :F system. In Proceedings of the 2018 12th International Conference on Reliability, Maintainability, and Safety (ICRMS), Shanghai, China, 17–29 October 2018; pp. 83–87.
30. Özdağ, R. The solution of the  $k$ -coverage problem in wireless sensor networks, In Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU), Zonguldak, Turkey, 16–19 May 2016; pp. 873–876.
31. Rezaee, A.A.; Zahedi, M.-H.; Dehghan, Z. Coverage optimization in wireless sensor networks using gravitational search algorithm. *J. Soft Comput. Inf. Technol. (JSCIT)* **2019**, *8*, 20–31.
32. Chen, Y.-N. Multilevel ( $k$ ) Coverage Based on Probabilistic Sensing Model in Wireless sensor Networks. Master's Thesis, National Chiao Tung University, Hsinchu, Taiwan, June 2019.
33. Chen, Y.-N.; Chen, C. Sensor deployment under probabilistic sensing model. In Proceedings of the 2nd High Performance Computing and Cluster Technologies Conference (HPCCT), Beijing, China, 22–24 June 2018; pp. 33–36.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).