SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2013)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.904

RESEARCH ARTICLE

Surveillance using non-stealthy sensors: A new intruder model

Amitabha Bagchi[1]*, Rajshekar Kalayappan[1] and Surabhi Sankhla[2]

[1] Department of Computer Science and Engineering, Indian Institute of Technology Delhi, New Delhi, India
[2] Boston Consulting Group, Delhi, India

ABSTRACT

We study the problem of intruder tracking with non-stealthy sensors, i.e., sensors whose ON/OFF state can be detected by an intruder, sometimes in advance. The sensor field is assumed to be operating with a sleep schedule to conserve energy. Both motion sensors and presence sensors are considered. We provide a rigorous basis for the study of this scenario by defining a new intruder model, the Ideal Intruder, that knows the entire sleep schedule of all the sensors in the field. More realistic intruders that have spatially and temporally limited knowledge of the sensor states are also defined. We study the well-known Random Independent Sleep (RIS) scheduling scheme with a single parameter p, giving mathematical bounds for the ideal intruder's crossing time in the motion sensor case and showing that the crossing probability in the presence sensor case undergoes a sharp transition as p increases. Further, we show that non-ideal intruders perform almost as well as the ideal intruder against RIS. Motivated by this finding and by the comparison between Barrier Coverage and RIS, we define a new sleep scheduling scheme, Spotlight, that is more robust to faults than Barrier Coverage and more effective than RIS. But more than that, Spotlight is shown to be specifically suited to the non-stealthy case because a non-ideal intruder, that is, one with limited information performs significantly worse against it than the ideal intruder. Spotlight, therefore, apart from being a novel and interesting sleep scheduling scheme in its own right, also illustrates the power of the analytical framework we introduce in this paper.

KEYWORDS
surveillance; intruder detection; sensor networks

***Correspondence**
Amitabha Bagchi, Department of Computer Science and Engineering, Indian Institute of Technology Delhi, New Delhi, India.
E-mail: bagchi@cse.iitd.ernet.in

1. INTRODUCTION

Wireless Sensor Networks are increasingly used for surveillance on various scales—ranging from tracking animals to monitoring the borders between nations. On detecting an intruder, the sensors may send a message to a remote base station or perform some other pre-defined task. The sensors may also be responsible to track the intruder until it leaves the Field of Interest (FoI). This work focuses on intruder detection only.

The engineering problem in this scenario is that of minimizing the cost in terms of battery power spent on running the sensors, an objective that must be achieved without compromising the objective of detecting the intruders presence. This objective function is normally studied by equipping the sensor batteries with a fixed amount of power and then measuring the lifetime of the system under the assumption that the batteries are never recharged.

The key insight that drives this optimization is that for the FoI to be effectively covered, the sensors must be spread all over it, but the intruder can only be detected by a few sensors—the ones closest to it. If it is possible to temporarily shut down the sensors that are unlikely to be in the intruder's range, then significant battery power can be saved. This approach engenders a range of solutions that are commonly called *sleep scheduling* algorithms. Such an algorithm is deemed effective if it prevents the intruder from crossing or intruding deep into the FoI during the network's lifetime. Our paper also studies sleep scheduling but in the somewhat more difficult situation where the intruder can detect and study the sleep patterns of the sensor nodes, what we call the non-stealthy case.

The non-stealthy scenario is very relevant because passive sensors that cannot be detected by an intruder are increasingly found to be ineffective in a number of application areas (see e.g., [1,2] for a case study in submarine detection), so intruder detection is carried out by active sensors that, by emitting one kind of signal or the other, open themselves up to detection by the very intruders they are supposed to be detecting (see e.g., [3,4] for a case

study of obstacle avoidance for helicopters.) In this paper, we focus on this scenario, differentiating between *presence sensors* that can detect an intruder whenever it is present in their sensing range and *motion sensors* that detect an intruder only if it moves while in their sensing range.

The major difference between the non-stealthy and the stealthy scenario is that we need a model for what the intruder knows. This is a tricky matter and is clearly a subject to many factors. But no systematic study of sleep scheduling is possible unless we assign the intruder some detection power. Clearly, the most powerful intruder is one that knows "everything", that is, the state of all the sensors at all times. A sleep scheduling algorithm that can detect this intruder can detect any intruder that is less powerful. This intruder forms a basic benchmark against which every algorithm, and also every less powerful intruder can be measured. We call such an intruder the *Ideal Intruder*. This intruder definition is the main contribution of this paper. We note that the Ideal Intruder is an abstraction in the sense that it is not really possible to implement such an intruder. The motivation for studying such an intruder is that any real intruder will not be as powerful as the Ideal Intruder, and hence, a sleep schedule that can detect an Ideal Intruder will be able to detect a real intruder as well; that is, the Ideal Intruder is a theoretical construction that helps us study the security of the intrusion detection scheme we are using. To demonstrate this, we will also study some non-ideal intruder models that work with limited information.

The ideal intruder is a benchmark in three ways. Firstly, it can throw light on the difference between the non-stealthy and stealthy cases for a particular sleep scheduling algorithm. We illustrate this use in passing in Section 4.4 with Random Independent Sleep (RIS) scheduling as the test case. More importantly, the ideal intruder can be used to categorize sleep scheduling algorithms according to their ability to defeat it (or, more precisely, to be able to detect or retard the progress of such an intruder with a given amount of energy) in the non-stealthy scenario. In Section 4, we study the performance of RIS and against an ideal intruder, using both motion and presence sensors. We characterize the crossing time for such an intruder in the motion sensor case and the crossing probability in the presence sensor case. We present simulations and mathematical arguments based on Percolation theory [5], particularly the directed graph version known as Oriented Percolation [6]. We find that the crossing time of the ideal intruder in the motion sensor case increases linearly to a certain value of the sensor ON probability then increases sharply. This is consistent with the reasoning derived from the study of Oriented Percolation. In the presence sensor case, we find that there is a sharp phenomenon: the crossing probability of an ideal intruder is 1 for lower values of the sensor ON probability, switching abruptly to 0 after a point. This phenomenon is also consistent with the Percolation-based view of the RIS scenario.

A key contribution of our paper is the comparison of the ideal intruder's crossing probability with the crossing probabilities of two weaker intruders for RIS (Section 4.4). We find that limiting the intruder's knowledge does not significantly degrade its ability to cross the grid in the RIS case. This is a weakness of RIS, which is exposed by studying it in the intruder framework. This and the comparison of RIS to Barrier coverage presented in Section 4.5 motivate the second main contribution of this paper: A sleep scheduling algorithm called *Spotlight* that is, to the best of our knowledge, the first sleep scheduling algorithm designed specifically for non-stealthy sensors (Section 5). This algorithm works like a spotlight by moving an aggregation of ON sensor cells around the FoI. This algorithm compares favorably to RIS in terms of crossing probability in the presence sensor case (Section 5.3) but clearly differentiates between ideal and non-ideal intruders (Section 5.4), making it the first sleep scheduling algorithm that specifically requires more information to beat. In order to address the issue that Spotlight requires coordination and information exchange, we discuss implementation issues in Section 5.6 and show that it is relatively lightweight to implement.

2. RELATED WORK

Sleep scheduling to conserve power in surveillance scenarios has been studied extensively [7], with some early schemes mimicking the patrolling behavior of human guards [8]. Turgut *et al.* [9] recognize that sensors may be visible, for example, because the wireless communication between the nodes can be detected.

The problem of coverage is also relevant to the study of the surveillance problem. Huang *et al.* [10–13] discuss the problems of coverage and k-coverage and the methods to determine the existence of coverage. For the surveillance problem, a more relevant and economical concept is Barrier Coverage (see e.g., [14]). This work studies the non-stealthy sensor case as well, concluding that a weak barrier of stealthy sensors is sufficient to detect intruders, whereas if the sensors are non-stealthy, a strong barrier is required. Necessary conditions for the existence of a strong barrier are described in [15] along with a localized algorithm to set up a strong barrier. Several other attempts have been made in this direction based on smoothness in intruder movement [16] and the related concept of Trap coverage that allows an intruder movement up to a fixed distance before being caught [17]. Finally, we mention the paper by Brass [18] that describes the solutions and properties of the many variants of the tracking problem: stationary/mobile sensors, stationary/mobile target, random or optimal sensor deployment, independent or globally coordinated search, and stealthy or visible sensors.

3. PRELIMINARIES

3.1. The FoI and intruder motion models

We assume the FoI is represented by a finite square grid $[L] \times [W]$ (where $[n]$ denotes the set $\{1, \ldots n\}$). This

representation actually incorporates two notions: (i) there are LW sensors that are placed one each at the center of square cells, that is, at locations $\{(i+\frac{1}{2}, j+\frac{1}{2} : 0 \leq i < L, 0 \leq j < W\}$ and (ii) the sensing range of the sensor is limited to the entire cell it inhabits. The y-dimension is the "width" of the field and, in the context of this paper, is the dimension that has to be "crossed" by the intruder. The "length" of the region is in a sense the intruder's room for maneuver, so typically, we will consider situation where $W < L$.

Mathematically, *intruder motion* can be viewed as taking place in an infinite three-dimensional $[L] \times [W] \times \mathbb{N}$ where the $\mathbb{N}$ denotes the natural numbers and represents the time dimension. We note that the source line is $\{(x, 0, 0) : x \in [L]\}$, that is, any point on the source side of the FoI at time 0. The destination plane is $\{(x, W, t) : x \in [L], t \in \mathbb{N}\}$ on the destination side of the FoI at any time. Because this three-dimensional grid has time as its third dimension, the only paths that we are allowed to consider in it are of the form $(x_0, y_0, 0), (x_1, y_1, 1), \ldots, (x_t, y_t, t), \ldots$, with each successive cell having a time value exactly one greater than the previous cell. This is under the assumption that it takes the intruder unit time to cross from one cell to another. The other two dimensions are determined by the square grid imposed on the two-dimensional FoI, that is, if from (x, y, t) the intruder can go to $(x, y, t+1)$ (i.e., not move) or go to $(x', y', t+1)$ where $|x-x'| \leq 1$ and $|y-y'| \leq 1$ and $|x-x'| + |y-y'| \geq 1$ (i.e., move to a neighboring cell, including a diagonal neighbor). Considering each cell to be a vertex and each possible movement described here to be an edge, we have an infinite graph (see Figure 1 for a one-dimensional scenario). In this infinite graph, we add a special vertex o, which is assumed to be where the intruder starts. This vertex has a direct edge to every node of the form $(x, 1, t), x \in [l], t \in \mathbb{N}$ because we assume that the intruder can wait outside the FoI for as long as it wants and then enter its first row at any point. We denote this graph $\mathcal{F}(L, W)$ or just $\mathcal{F}$ when L and W are known. Thus, intruder motion becomes a path in $\mathcal{F}$.

3.2. The surveillance model

We assume that at each time t, each sensor in the FoI is either ON or OFF. Hence, the entire graph $\mathcal{F}$ is not "safe" for the intruder looking to cross. The subgraph of $\mathcal{F}$ that is safe for the intruder depends on the sensor capability model. If we are considering presence sensors and if the intruder's path goes through node (x, y, t) and the sensor in cell (x, y) happens to be on at time t, then the intruder is detected. Hence, given a schedule $U = \{(x, y, t) :$ the sensor at (x, y) is ON at time $t\}$, the *safe graph* $\mathcal{F}_U^P$ is the subgraph of $\mathcal{F}$ induced by removing all the nodes of U.

In the case of motion sensors, the intruder may stay stationary in the sensing range of an active sensor and is not caught. For an intruder to move from cell (x, y, t) to cell $(x', y', t+1)$ (where $x-1 \leq x' \leq x+1$ and $y-1 \leq y' \leq y+1$), the sensors in both cells must be OFF. In addition to this, the intruder has another capability. It can always move

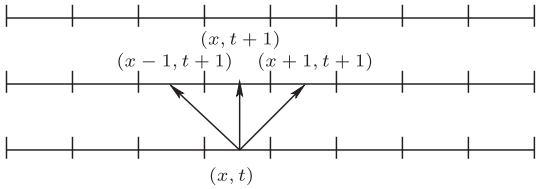


Figure 1. Intruder motion for one-dimensional Field of Interest.

from cell (x, y, t) to cell $(x, y, t+1)$, regardless of whether the sensors in these two cells are ON or OFF. That is, it can stay stationary, at position (x, y), without being caught, even if the sensor is ON. Hence, given a schedule U, the safe graph $\mathcal{F}_U^M$ in this case is the subgraph of $\mathcal{F}$ induced by retaining all the edges of $\mathcal{F}$ that have both endpoints not in U, that is, in OFF state. Note that a basic difference between $\mathcal{F}_U^P$ and $\mathcal{F}_U^M$ is that in the latter, the edges of the form $(x, y, t) \to (x, y, t+1)$ are always present even if both the vertices are ON because they correspond to the intruder staying motionless in the cell (x, y), whereas in $\mathcal{F}_U^P$, the edge may not exist if (x, y) is in ON state at either time t or $t+1$.

3.3. Sleep scheduling

We assume each sensor has a fixed lifetime, τ_s, because it has fixed battery power. Hence, any meaningful schedule U has the property that for a fixed (x, y), the number of ON time slots are exactly τ_s. We call this the *sensor lifetime*. Consider the quantity $\tau_n(U) = \max_t(x, y, t) \in U$ for some (x, y), that is, the maximum time that after which no sensor is ON. This quantity is called the *network lifetime*. A naive method for intruder detection would keep all the sensors in the FOI together as long as they can be kept ON. Clearly in the naive scheme $\tau_s = \tau_n$, whereas a smarter method would use redundancy to ensure $\tau_n > \tau_s$. These smarter methods involve putting sensors to sleep when they are not required and are known as sleep scheduling algorithms. Note that the communication electronics of the sensors are always awake, or may follow some other sleep scheduling scheme that guarantees the required Quality of Service in terms of communication.

We study some well-known sleep scheduling algorithms. The first of these is the RIS. Here, every sensor (x, y) is put to sleep at time t with probability p independent of its own state at all other time instants and of all other sensors at all times. Of course, τ_s is upper bound on the number of time steps for which the sensor can be awake, and so (x, y, t) should be put into sleep mode for all $t > t'$ where t' is the earliest time such that at least τ_s of the cells $(x, y, 0), (x, y, 1), \ldots, (x, y, t')$ are awake. In this case, τ_n is a random variable with expectation $\frac{\tau_s}{p}$, so the lower the p, the better the network lifetime. Of course, lower values of p are not as good as detecting intruders as higher values. The second simple and well-known scheme we study is *Barrier Coverage* (e.g., "Stint" [19], with $k = 1$). Here, the cells $(x, 1, t), x \in [L]$ are kept ON for $0 \leq t < \tau_s$, followed by

the cells $(x, 2, t), x \in [L]$ for $\tau_s \leq t < 2\tau_s$ and so on; that is, in the rectangular FoI, the row closest to the entry side is kept ON till the sensors in it die out, then the next row, then the next till all the sensors are dead. Clearly, here we have $\tau_n = \tau_s \times W$. In Section 5, we describe a new scheme called Spotlight that is specifically designed for the non-stealthy scenario.

Evaluation metrics.

Any intruder can cross the FoI in $\tau_n + W$ steps by crossing after the network dies. In the case of motion sensors, even an intruder that can only detect the ON/OFF state of the sensor in its current cell and its neighbors will *never* be caught because it can hide simply by staying stationary. Hence, in the motion sensor case, the only criterion we measure is the *time taken to cross the FoI* and that too as a fraction of $\tau_n + W$. In the presence sensor case, there is a non-zero probability that an intruder inside the FoI will be apprehended because it is stuck in an area where the cell it is in and all cells around it have sensors in the ON state. Hence, we measure the intruder's *probability of crossing* as well.

3.4. Intruder models

The *ideal intruder* is defined as one who has full knowledge of the safe graph $\mathcal{F}_U$ of any sleep schedule U. For example, in RIS, the set U is randomly constructed, and so $\mathcal{F}_U$ is a random graph, but the ideal intruder knows which edges and nodes are present in this random graphs. We note that this definition gives much power to the intruder and can be thought of as a way of "stress testing" our intrusion detection scheme. In reality, an intruder will probably possess limited knowledge about the functioning of the network, and so any scheme that can detect the ideal intruder will also be able to detect this real, non-ideal, intruder. To demonstrate this, we also introduce non-ideal intruders that have limited spatial and temporal knowledge. Specifically, we discuss intruders that know the ON/OFF schedule of the cell that they currently reside in and the eight cells that surround this cell for the next three time slots. Because the point of entry into the grid affects the performance of these intruders, significantly, we introduce two versions: *Non-Ideal1* that enters the grid at the point through, which is able to cover the maximum possible depth of the FoI, and a more realistic *Non-Ideal2* that enters the FoI through the largest open gap, that is, it considers the nodes $(i, 1, 0), 1 \leq i \leq L$ of $\mathcal{F}_U$ and finds the longest sequence of nodes where the sensors are all OFF, entering through the center of this sequence; that is, if $(i, 1, 0), j \leq i \leq k$ are all OFF and $k - j$ is the max of all such sequences, this intruder enters at $(\lfloor (j + k)/2 \rfloor, 1, 0)$. As benchmarks, we also consider two non-ideal intruders that operate with no information. *Non-Ideal3* chooses a random point of entry and moves in a straight line across the FoI. *Non-Ideal4* enters at a random point and chooses its next step uniformly at random from the three possible choices that move it further across the FoI. These two intruders basically treat the sensors as stealthy; that is, they cannot detect the presence of the sensors and so make their decisions without any information.

4. ANALYZING RIS

4.1. Lifetime

We begin by recalling that the safe graphs $\mathcal{F}_R^P$ and $\mathcal{F}_R^M$ for the case where R is chosen by RIS are random graphs as the choice of R is random. Let us start by studying the lifetime of RIS schemes before moving to the specific types of sensors and intruders.

Proposition 4.1. *For RIS with parameter p on a* $[L] \times [W]$ *grid,*

$$\tau_n = \Omega(\log_{1/(1-p)} LW)$$

with high probability.

Proof. Hence, the network lifetime τ_n is a random variable. The time for which a given sensor (x, y) retains battery power $\tau(x, y)$ can be expressed as $X_1(x, y) + X_2(x, y) + \cdots X_{\tau_s}(x, y)$ where each $X_i(x, y)$ is independent and geometrically distributed with parameter p, and

$$\tau_n = \max_{x \in [L], y \in [W]} \tau(x, y)$$

Now, because $\{\forall i : X_i(x, y) > k/\tau_s\} \Leftarrow \{\tau(x, y) > k\}$ and the $\{X_i(x, y) : 1 \leq i \leq \tau_s\}$ are independent, we can say that

$$\mathrm{P}(\tau(x, y) > k) \geq \left((1-p)^{k/\tau_s}\right)_s^{\tau} = (1-p)^k$$

By the definition of τ_n, we have that

$$\mathrm{P}(\tau_n > k) = \mathrm{P}(\exists (x, y) : \tau(x, y) > k)$$

Because all the LW points of the form (x, y) are independent, we obtain

$$\mathrm{P}(\tau_n > k) \geq 1 - \left(1 - (1-p)^k\right)^{LW}$$

Now, if $k \leq \log_{1/(1-p)} LW - \ln \log_{1/(1-p)} LW$, then $(1-p)^k > (\ln LW)/LW$ and so $\left(1 - (1-p)^k\right) < e^{-(\ln LW)/LW}$. Therefore, for this choice of k, $\mathrm{P}(\tau_n > k) \geq 1 - 1/LW$. □

When we compare this lifetime to that of Barrier Coverage, $\tau_n = \tau_s \cdot W$, we realize that the lifetime could be much smaller if L is comparable to W, for example. However, Barrier Coverage is not robust, even a single faulty sensor opens a hole that the intruder can escape through, especially the ideal intruder who can detect this fault. RIS does not have this problem; in fact, it is highly robust. Even if every sensor is faulty with some small probability ϵ (leading to an expected ϵn faults), it just means RIS runs with a

slightly smaller parameter value $p - \epsilon$. So an advantage in terms of robustness is traded off against lifetime.

4.2. Motion sensors and crossing time

As discussed earlier, because $\mathcal{F}_R^M$ contains all edges of the type $(x, y, t) \rightarrow (x, y, t + 1)$, the crossing probability of the ideal intruder is 1. Indeed, this is true of any intruder that can detect the state of the cells neighboring its current position. So, the problem here is to determine how quickly the ideal intruder can cross the FoI. Let us simplify the problem by assuming that the ideal intruder starts by entering the FoI at some cell $(x, 1)$ and then crosses the FoI by changing only the y coordinate when it can, staying in the same column of the FoI. This intruder makes progress whenever both cell (x, j, t) and $x, j + 1, t)$ are in OFF state, which happens with probability $(1 - p)^2$. It needs to make such progress W times. Hence, the crossing time along a fixed column x, $\tau_c(x) = X_1(x) + X_2(x) + \cdots + X_W(x)$ where each $X_i(x)$, that is, the time taken to make progress from row i to row $i + 1$, is geometrically distributed with parameter $(1 - p)^2$. Hence, $\mathrm{E}(\tau_c(x)) = W/(1 - p)^2$. In actuality, the crossing time in this case will be the minimum of this quantity and $\tau_n + W$.

But the ideal intruder can do better than stick to one column. It can, for example, choose the best of the L columns it can enter. For this intruder, we set $\tau_c = \min_{x \in [L]} \tau_c(x)$. Clearly, $\{\tau_c(x) \geq k\} \Rightarrow \{\exists i : X_i > k/W\}$, and so $\mathrm{P}(\tau_c(x) \geq k) \leq W \cdot (1 - (1 - p)^2)^{k/W}$. From here, given the L columns are independent, we have that $\mathrm{P}(\tau_c > k) \leq W^L \cdot (1 - (1 - p)^2)^{kL/W}$. Setting $k = (W \log_\beta W)/2 - (W \log_\beta WL)/L$, where $\beta = (1 - (1 - p)^2)$, we have $\mathrm{P}(\tau_c > k) \leq 1/(WL)$. This bound implies that if we study the growth of the crossing time with parameter p, it should vary in inverse proportion to $\ln(1/(2p - p^2))$. However, in Figure 2, we see that in simulation for a fixed width, the crossing time grows much more gently in the range [0,0.7] than this bound implies, growing more rapidly after that.

To gain some insight into why this is the case, we turn to the Oriented Percolation model studied by Durrett [6]. We will rely on insights from this theory to study RIS without proving specific results. In Figure 3, we show a two-dimensional graph wherein the intruder is allowed, as in the RIS motion sensor case, to move from (x, i, t) to $(x, i - 1, t + 1)$ or $(x, i + 1, t + 1)$ through solid edges. The dashed edges are those that our intruder is also allowed to use, signifying its non-motion. We call this the RIS Intrusion Graph, which we will introduce more formally in Section 4.3. This graph model is very similar to the Oriented Percolation model presented in [6]. For Oriented Percolation, it is known that there is a critical probability p_c such that the origin of the oriented infinite grid is connected to ∞ with positive probability when $p > p_c$ and with probability 0 when $p < p_c$. Our RIS Intrusion Graph also shows a similar behavior as we will demonstrate formally in Section 4.3, with the difference that in the RIS Intrusion Graph, a low value of the ON/OFF probability corresponds to a high value of the edge retention probability in Oriented Percolation.

In Section 4.3, we will see that just like in Figure 2, there is a change in behavior around 0.7 in the presence sensor case as well (Figure 4). This is due to the change in behavior of the underlying random graph model as we will discuss in more detail ahead.

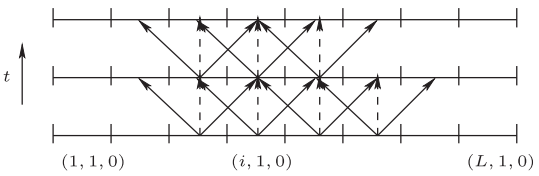


Figure 3. The Random Independent Sleep intrusion Graph.

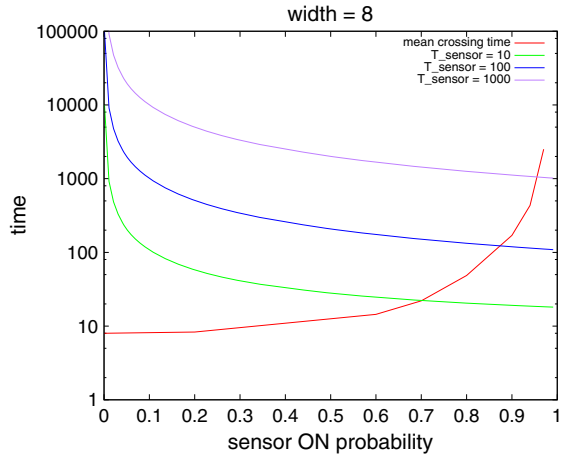


Figure 2. Ideal intruder versus Random Independent Sleep motion sensors.

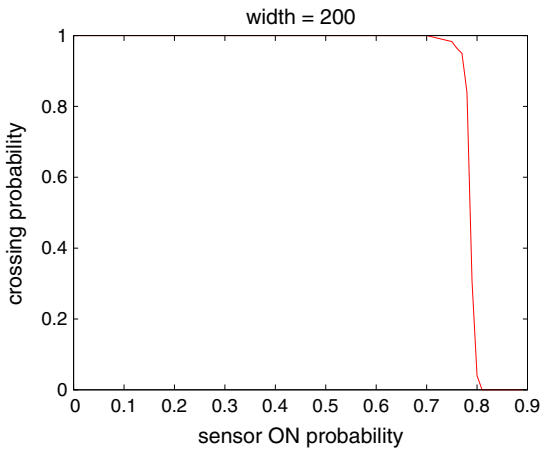


Figure 4. Ideal intruder versus Random Independent Sleep presence sensors.

4.3. Presence sensors and crossing probability

The ideal intruder was pitted against a RIS presence sensor network. The curves obtained are as shown in what follows. Figure 4 shows the variation in crossing probability with increase in sensor ON probability.

On increasing p beyond a certain point, the intruder abruptly finds itself unable to cross undetected. This is completely consistent with the view presented previously that the intruder model demonstrates a critical phenomenon like the oriented percolation model. We now give a partial mathematical treatment of this phenomenon, which will involve the following major steps

- We will define an infinite random graph model, called the *RIS Intrusion Graph*, such that RIS on a finite sized one-dimensional FoI is a subgraph of this model.
- We will show that there is a critical value p_c of the parameter p such that for all $p < p_c$, the intruder can move through the graph undetected for an infinite number of times steps with constant probability, and that for all $p > p_c$, this happens with probability 0.
- We will establish that this value p_c is non-trivial; that is, that it lies strictly between 0 and 1.

Definitions and notation.

For simplicity, let us consider a one-dimensional FoI, that is, a sequence of cells $0, 1, \ldots, W-1$ such that the intruder has to enter at cell 0 and exit from cell $W-1$. We note that when we take time into consideration, this gives rise to a two-dimensional graph structure, as seen in Figure 3, in which the nodes are of the form (i, t) where $0 \leq i < W$, signifying the physical position of the intruder and $t \geq 0$ signifying the time instant at which cell i is being represented. In this graph, directed edges are placed from (i, t) to the nodes $(i-1, t+1)$ signifying a backward step, $(i+1, t+1)$ signifying a step forward and $(i, t+1)$ representing the case where the intruder does not move. We will call this graph structure the *Intrusion Graph* built on the set $S = \{0, 1, \ldots, W-1\}$ and denote it by $\mathcal{I}(S)$.

Let us consider the case where instead of just the cells $0, 1, \ldots, W-1$, our FoI consists of the entire integer line; that is, let us consider the Intrusion Graph built on the set $\mathbb{Z}$, denoted $\mathcal{I}(\mathbb{Z})$. This graph has vertex set $\mathbb{Z} \times \mathbb{Z}_+$ (where $\mathbb{Z}_+$ denotes the non-negative integers) and the edge set as described previously. Applying the RIS schedule with parameter p and $\tau_s = \infty$ on this infinite graph is equivalent in probabilistic terms to considering the probability space $(\Omega, \mathcal{F}, \mathbb{P}_p)$ where $\Omega = \{0,1\}^{\mathbb{Z}\times\mathbb{Z}_+}$ is the set of all possible configurations of the Intrusion Graph with each sensor choosing to be in either an ON state corresponding to 0 (as an intruder cannot pass if the sensor is ON) or an OFF state corresponding to 1. In the following, we will refer to cells in state 1 as being *open* and those in state 0 to be *closed*. For $\omega \in \Omega$, if we denote by $\omega(u)$ the coordinate of ω corresponding to the cell u, then a natural partial order on Ω is defined as follows: Given $\omega_1, \omega_2 \in \Omega$, we say $\omega_1 \preceq \omega_2$ if $\omega_1(u) \leq \omega_2(u)$ for all $u \in \mathbb{Z} \times \mathbb{Z}_+$. Given a σ-algebra $\mathcal{F}$ defined on Ω, we say that $A \in \mathcal{F}$ is an *increasing event* if $\omega \in A$ implies that $\omega' \in A$ for all ω' such that $\omega \preceq \omega'$, that is, if a particular configuration ω satisfies the event A, then all configurations ω' that have the same open cells at those in ω (and may have some other cells also open) are also necessary in A.

The natural σ-algebra on Ω is considered, and the probability measure, $\mathbb{P}_p$, defined on this space is the product measure, which assigns a cell (x, t) value 0 with probability p and a value 1 with probability $1-p$. This definition gives rise to a random graph model, which we will call the RIS Intrusion Graph.

Denote by $L(x, t)$ the number of time steps the ideal intruder that is placed at cell x at time t can stay in the RIS Intrusion Graph without being detected. In graph terms, this is equivalent to saying that $L(x, t)$ is the length of the longest (directed) path of open cells starting at (x, t). We will use only L to denote the length of the longest open path beginning at $(0, 0)$. Further, we denote by $\theta_{x,t}(p)$ the probability of the event $\{L(x, t) = \infty\}$, dropping the subscript when $(x, t) = (0, 0)$.

Critical phenomenon.

Consider the event that there exists a path of infinite length in the RIS Intrusion Graph with parameter p, and denote its probability by $\psi(p)$. Now, clearly it is not possible to determine if this event occurs by looking at any finite subset of the cells and edges of the RIS Intrusion Graph. Therefore, this event is a tail event and Kolmogorov's 0-1 law (cf. [20, Chapter 4]) applies to it, that is, $\psi(p)$ is either 0 or 1. Also, $\psi(p)$ is non-increasing in p; that is,

Proposition 4.2. *Given $p_1, p_2 \in [0, 1]$ such that $p_1 \leq p_2$,*

$$\psi(p_1) \geq \psi(p_2).$$

This follows easily by a coupling argument that allows us to reason about the two models (i.e., RIS Intrusion Graph with parameter p_1 and with parameter p_2) simultaneously. Because the result is quite intuitive and the coupling argument is very standard (c.f. Theorem 2.1 of [5, Thm 2.1, page 33]), we omit it here.

Because $\psi(p)$ is either 0 or 1 and is also non-increasing in p, we can state the following proposition.

Proposition 4.3. *There is a $p_c \in [0, 1]$ such that $\psi(p) = 1$ for all $p < p_c$ and $\psi(p) = 0$ for all $p > p_c$.*

We note that the proposition does not say what happens at $p = p_c$. Such questions tend to be subtle and require deeper investigation.

The link between $\psi(p)$ and $\theta(p)$ is quite tight. In fact, $\theta(p) = 0$ if and only if $\psi(p) = 0$. This can be argued as follows: If $\theta(p) = 0$, then by the symmetry of the RIS Intrusion Graph, $\theta_u(p) = 0$ for all $u \in \mathbb{Z} \times \mathbb{Z}_+$. This means that because the set $\mathbb{Z} \times \mathbb{Z}_+$ is a countable set,

$\sum_{u\in\mathbb{Z}\times\mathbb{Z}_+}\theta_u(p) = 0$. But this sum is an upper bound on $\psi(p)$. This proves one direction of the implication. The other direction is also easy to show, but we omit the argument here. We state this observation as a proposition.

Proposition 4.4. *There is a* $p_c \in [0,1]$ *such that* $\theta(p) > 0$ *for all* $p < p_c$ *and* $\theta(p) = 0$ *for all* $p > p_c$.

Lower bound on critical probability.

The existence of a critical probability is an important fact, but it is not of much use if p_c is 0 or 1, because there is only kind of behavior the model displays. Hence, it is important to demonstrate that the critical value p_c is non-trivial. Here, we show a lower bound on p_c that shows that it is not 0.

Proposition 4.5. *For the RIS Intrusion Graph,*

$$p_c > \frac{2}{3}$$

Proof. Let us consider the component of the RIS Intrusion Graph rooted at the origin $(0,0)$. We note that there are three possible cells that $(0,0)$ is connected to. Each of those cells is, in turn, connected to three possible cells, although at the next level, there is an overlap and hence there are only five cells that are reachable from the origin in two steps.

Let us define a Galton–Watson branching process (GWBP) that begins with one individual. The progeny distribution for each individual mimics the possibility of moving out from a cell of the RIS Intrusion Graph. Each individual can generate at most three progeny, and it chooses each of these three progeny independently with probability $1-p$. Therefore, the mean number of progeny is $3(1-p)$.

Note that if we identify the origin in the RIS Intrusion Graph with the initial individual of the GWBP, then we can couple the cluster of nodes reachable from the origin in the RIS Intrusion Graph to the number of individuals in the entire GWBP in such a way that the number of individuals in the GWBP dominates the number of nodes in the cluster of the origin in the RIS Intrusion Graph. Hence, if the GWBP becomes extinct in finite time with probability 1, then the length of the longest path starting at the origin in the RIS Intrusion Graph is finite with probability 1. Now, as we know, if the expected number of progeny of any individual of a GWBP is ¡ 1 then the GWBP started with one individual becomes extinct in finite time with probability 1. In our case, the condition becomes $3(1-p) < 1$ and the result follows. □

If we compare this lower bound to the curve plotted in Figure 4, we will see that this is consistent with the observation that is made for a finite subgraph of the RIS Intrusion Graph and is in fact quite close to the observed critical value that can be surmised to be between 0.7 and 0.8 from the plotted graph.

The analysis presented here is not complete or comprehensive, it is merely illustrative of the fact that ideas developed in Percolation theory are applicable in our scenario and can explain the observed phenomenon. Developing the analysis in full detail will be repetitive of the techniques demonstrated in [6] and will detract from the main focus of this paper.

4.4. Comparing with non-ideal intruders

First, we note that the non-ideal intruders that use some information, that is, *Non-Ideal1* and *Non-Ideal2*, do significantly better than the *Non-Ideal3* and *Non-Ideal4* that use no information (Figure 5). The performance of the non-ideal intruder that can see only three steps ahead in space and time, as expected, is not as good as that of the ideal intruder, but surprisingly, despite the strong constraints, we find that *Non-Ideal1* and *Non-Ideal2* do not perform significantly worse than the ideal intruder against presence sensors on a grid of width 15. *Non-Ideal1*, which chooses the point on entry that gives it greater initial movement, performs marginally better in terms of crossing probability than *Non-Ideal2*, which chooses the center of the largest open gap as its point of entry, but both of them attain crossing probability 0 at a significantly high value of p.

The reason for this is that RIS sensors do not coordinate in any way, so any small neighborhood (in space and time) may have, with high probability, an escape route for an intruder with non-zero knowledge of what is going to happen in its immediate vicinity. We support this argument with a small calculation. Assume that the non-ideal intruder is a (x,y,t). Now, consider the cube defined by $[x-3,x+3]\times[y-3,y+3]\times[t,t+3]$. For the intruder to be trapped, there must be a boundary of ON cells blocking its path in this cube. But for each of the three time steps under consideration, the probability that the intruder has nowhere to go (including the option of staying in the current cell) is p^9. Hence, the probability that the intruder is trapped in

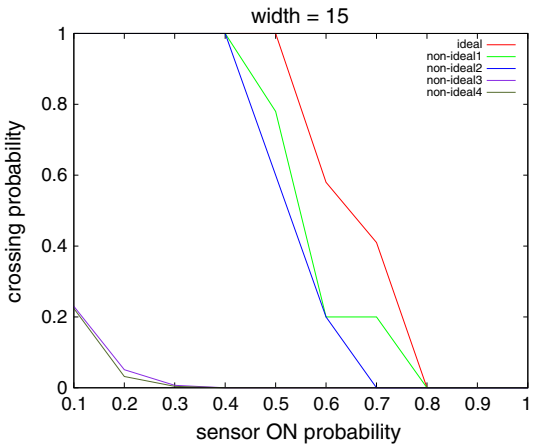


Figure 5. Random Independent Sleep: crossing probability versus p.

this cube is at most $3 \cdot p^9$, which is a very low probability for any reasonable value of p. However, the argument may be made that this is a very local argument in time and space. To take this further, we have to appeal to the concept of k-dependent percolation.

Bollobás and Riordan [21] describe the k-dependent percolation model as being a generalization of percolation where each site is open or closed with marginal probability p, but the probability of a site being open is *not* independent of its neighbors up to a distance of k. The critical phenomenon observed in independent percolation is seen here as well. If we consider each cube described previously as a single point, then we will observe that the probability of each cube being "inescapable" is bounded above by $3 \cdot p^9$ as mentioned, but neighboring cubes, up to a distance of 3 are not independent. Hence, the model becomes a 3-dependent oriented percolation model. Clearly, as we note in Figure 5, when p becomes high enough, the probability of a cube being inescapable does fall and the intruder's crossing probability goes to 0, but this happens quite close to the value where the ideal intruder itself cannot cross, which means that no path exists in any case.

The only difference is that the non-ideal intruder may make non-optimal local choices while hopping around to escape detection. But these do not, for a high enough sensor ON probability, lead to detection. They just mean that the time this intruder takes to cross may be a little longer. In Section 5, we describe the Spotlight scheme that addresses the problem of local escapability in RIS and makes life much more difficult for non-ideal intruders. But before we move on to that, we note in passing from Figure 5 that with stealthy sensors, the crossing probability is so low for width 15 that the curve hugs the x-axis at the scale used.

4.5. Comparison of RIS and Barrier Coverage

If the network is implementing a Barrier Coverage scheme, then it is immaterial if the sensors are motion sensors or presence sensors. In either case, for the lifetime of the network, given by $\tau_n = \tau_s \times W$, an ideal intruder cannot cross the FoI.

The Barrier Coverage scheme seems very attractive as it provides surveillance, with a guarantee of a high network lifetime. Given an FoI of width 200, RIS scheme gives a network lifetime of approximately $\tau_s \times 1.25$. Barrier Coverage, on the other hand, gives a network lifetime of $\tau_s \times 200$. Thus, Barrier Coverage seems the clearest choice.

However, the advantage that RIS provides is robustness. Consider a single faulty sensor in an active row in the Barrier Coverage scheme. The entire FoI behind the barrier is asleep. As a result, the network is completely compromised. An immediate solution would be to suggest having two active rows at a time. Even here, the occurrence of two faulty sensors is not a very rare event. Considering the harsh conditions these sensors are deployed in (mountainous regions have high risk of avalanches, which may compromise tens to hundreds of sensors in a single event), it is likely that sensors across multiple rows are compromised. In such a scenario, the RIS scheme can still provide surveillance.

The aim now is to develop a sleep scheduling algorithm that is robust yet gives a high network lifetime.

5. SPOTLIGHT—A ROBUST, ECONOMICAL SLEEP SCHEDULE

A synergy of the robustness of the RIS scheme and the economy of the Barrier Coverage scheme is desired. In this section, we propose such a scheme: Spotlight coverage. A network of presence sensors is considered here.

5.1. The problem with RIS and a solution

Firstly, the reason behind the poor economy of the RIS scheme is analyzed. Consider the grid shown in Figure 7. The detection of an intruder at time $t + 1$ requires the occurrence of the following scenario.

> If an intruder is in the cell (x, y) at time t ((x, y, t)), then every cell (x', y') at time $t + 1$ ($(x', y', t + 1)$) has its sensor ON, where $x - 1 \leq x' \leq x + 1$ and $y - 1 \leq y' \leq y + 1$.

This is depicted in Figure 7. In RIS, the probability of occurrence of such a configuration is small, given by, p^9. The probability that one of the three cells that lead to forward motion is OFF at time $t + 1$ is $1 - p^3$, which can be quite high for small p. Hence, if p is small, the intruder can not only evade detection at time $t+1$ but also move forward with good probability. This makes *RIS* an uneconomical strategy.

An intuitive approach to addressing this problem is to aggregate the cells that have their sensors ON at any given time. This gives rise to distributions like the one shown in Figure 6. When the ON sensors (20 in number in this example) are arranged this way, the chances of the intrusion detection scenario occurring are higher. In Figure 6, the intrusion detection scenario occurs in six cells (labeled with numerals). If the intruder is in any one of six cells

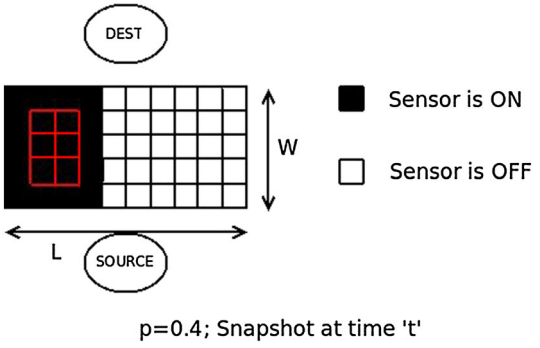


Figure 6. Random Independent Sleep drawback mitigation.

DOI: 10.1002/sec

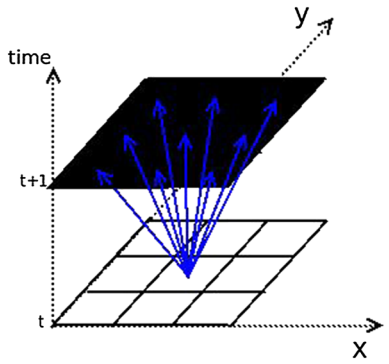


Figure 7. Intrusion detection scenario.

in the previous time step, it is caught. Clearly, aggregation greatly improves the chances of detection.

What must be the shape of the aggregation? In Figure 6, if the intruder, in the previous time step, is in any one of the 14 cells where the sensor is ON, other than those marked with numerals, it may escape undetected. These non-contributing cells always occur on the fringe of the aggregation. The aim is to aggregate sensors in such a way that the number of fringe cells is minimized. This indicates that the aggregation should maximize the size of the interior and minimize the boundary cells. In the grid model, this can be approximated by a square because for a square of n cells, the boundary is $\theta(\sqrt{n})$, which is asymptotically minimum in two dimensions. Hence, the spotlight scheme we describe next uses a square-shaped aggregation.

5.2. The spotlight algorithm

Based on the mitigation strategy explained in Section 5.1, the Spotlight algorithm takes a parameter p and builds a square-shaped aggregation of size pLW. At every time step, this aggregation is centered at a randomly chosen point that is independent of the centers at all other times.

The algorithm is detailed in Figure 8. In other words, at each time, the aggregation is placed at a random position (the number of ON cells remaining the same). This gives rise to the name of the scheme, Spotlight. In theater, a spotlight, capable of illuminating a fixed area of the stage, is focused on different parts of the stage, at different times, as required. Similarly here, a different portion of the FoI is under detection in each time step.

5.3. Comparison with RIS

We analyze Spotlight's performance through simulation. We notice, most importantly, that there is a critical value of the parameter p at which the probability of the ideal intruder crossing falls from 1 to 0 for Spotlight as well. We note that this value is much lower for Spotlight than it is for RIS (Figure 9). To understand why this might be the case, consider the intruder at (x, y, t). Now, under the RIS scheme, the probability of surviving at $t+1$, that is, moving without detection, is given by $1 - p^9$. Let us calculate this

Algorithm **Spotlight**(p)

1. Compute the number of sensors that need to be ON at a given time, $N_{ON} = p \cdot L \cdot W$ where p is the sensor ON probability.
2. Find the dimensions of the aggregation, l_a and w_a
   - If N_{ON} is a perfect square, $l_a = w_a = \sqrt{N_{ON}}$
   - else, $w_a = \lfloor \sqrt{N_{ON}} \rfloor$ and $l_a = \frac{N_{ON}}{w_a}$

   Note : if $w_a > W$, then $w_a = W$ and $l_a = \frac{N_{ON}}{w_a} = \frac{N_{ON}}{W}$
3. For a particular time step t, find a random cell (x_s, y_s, t) to serve as the centre of the aggregation.
4. Build the aggregation : set sensors of all cells (x', y', t) ON, where $x' - x_s \leq l_a/2$ and $y' - y_s \leq w_a/2$.
5. Repeat steps 3 and 4 for all t.

Figure 8. Spotlight algorithm with parameter p.

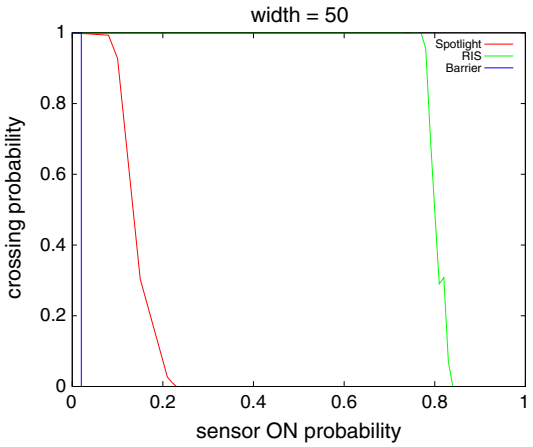
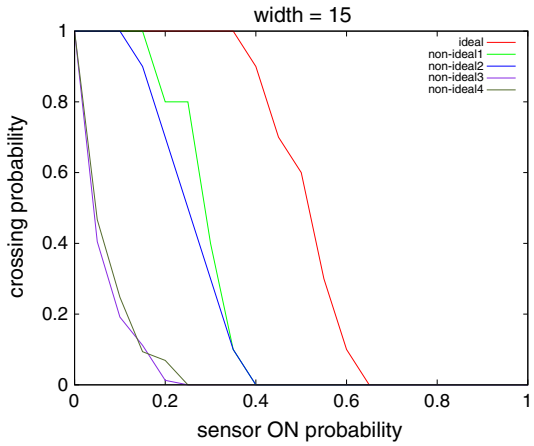


Figure 9. Spotlight: Crossing probability.

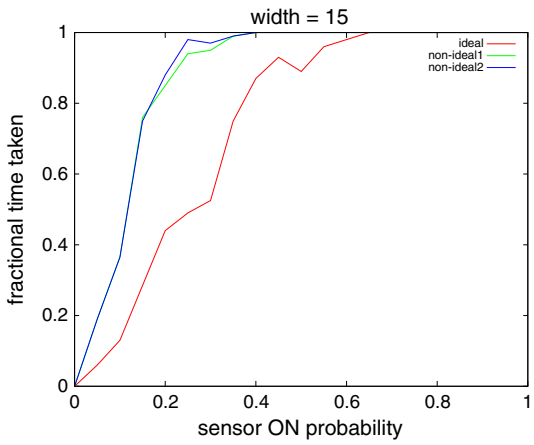
probability under the spotlight scheme. If the center of the spotlight at time $t+1$ is in any of the $(\sqrt{N_{on}} - 1)^2$ cells that form a square centered at (x, y), then the intruder has no option for escape. Plugging in the value of N_{on}, we calculate that the probability of the center of the spotlight being in this region is $\frac{(\sqrt{p \times l \times w} - 1)^2}{L \times W} = p \times \left(1 - \frac{1}{\sqrt{p \times l \times w}}\right)^2$. If we compare this term with the quantity p^9 (i.e., the probability of surviving at $t+1$ in RIS), it is clearly much larger than the corresponding probability under RIS, as long as the values of L and W are reasonably large. This analysis indicates that Spotlight should work better than RIS in practice, although it is by no means a rigorous proof.

5.4. Spotlight and non-ideal intruders

In order to have better handle on Spotlight, we analyzed through simulation the performance of the two non-ideal intruders *Non-Ideal1* (that enters through the point it feels



(a) Crossing probability vs p



(b) Crossing time as a fraction of τ_n vs p

Figure 10. Spotlight in a 20×15 grid with $\tau_s = 10$.

will give best penetration in the next three steps) and *Non-Ideal2* (that enters through the largest "open" gap in the FoI). We observed that the non-ideal intruder models underperform with respect to the ideal intruder (Figure 10) significantly, as opposed to RIS where the gap in performance was much lower. The non-ideal intruders fail at a significantly lower probability and take significantly longer to cross even in the low ranges of p where they succeed in crossing. This is expected as Spotlight is designed to beat the kind of spatially and temporally local knowledge that a realistic intruder may possess. Here too, we see that the performance of *Non-Ideal3* and *Non-Ideal4* is significantly worse than that of *Non-Ideal1* and *Non-Ideal2*, but the information that the latter two non-ideal intruders have is not enough to beat Spotlight.

The bottomline is that beating Spotlight takes *more information* than beating RIS, and this illustrates both the value of Spotlight as a sleep scheduling scheme and of our intruder models as a framework for analyzing such schemes.

5.5. Barrier-Over-Time phenomenon in spotlight

Interestingly, the spotlight scheme also deters the motion of the intruder by means of a barrier. However, this barrier is different from that of Barrier Coverage in that its formation is across multiple time slots. This is elaborated further.

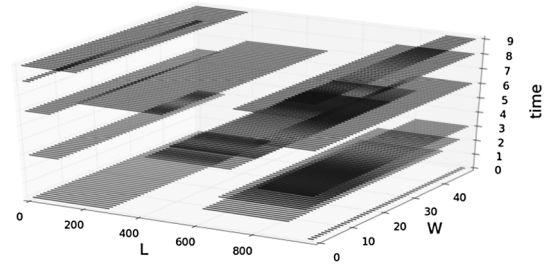
In the spotlight technique, in each time slot, a rectangle (possibly a square) of ON sensors is present. This is a sub-barrier. Let the intruder be at (x, y, t), that is, in the row y, column x at time t. Let a sub-barrier at time t_1 be defined by $(x_1, x_2, y_1, y_2, t_1)$, meaning the sub-barrier extends from column x_1 to x_2, and row y_1 and y_2. Let $t_1 > t$. Note that if (i) $y_2 - y > t_1 - t$, (ii) $x_2 - x > t_1 - t$ and (iii) $x - x_1 > t_1 - t$, then from (x, y, t), the ideal intruder cannot make any forward progress. It has to move backward. In addition to the previous conditions, if $y - y_1 > t_1 - t$, then the intruder cannot go anywhere from (x, y, t), not even backward. It is bound to be caught. Hence, we use the term *sub-barrier*.

Now, if it so happens that a subset of the sub-barriers overlap, over time, in such a way that the ideal intruder at $(x, 0, 0)$ finds it impossible to reach row W, no matter what path it takes, then we say that an *x-barrier* is in place. If for each x there exists an x-barrier, then surveillance is guaranteed, and we say that a *barrier-over-time* is in place. This barrier does not exist in a single time slot. Rather, the combination of multiple sub-barriers, occurring at different time slots, forms the barrier (Figure 11(a), time advances on the z-axis). We studied the location of the barrier-over-time phenomenon through simulation. As expected, when p is below the critical probability p_c, a barrier does not exist (Figure 11(b)). Upon increasing sensor ON probability beyond p_c, we see that the barrier forms closer to row 0. With increasing value of p, the penetration or progress of the intruder decreases. The decrease is sharp for values of p just greater than p_c and becomes more gradual as p increases.

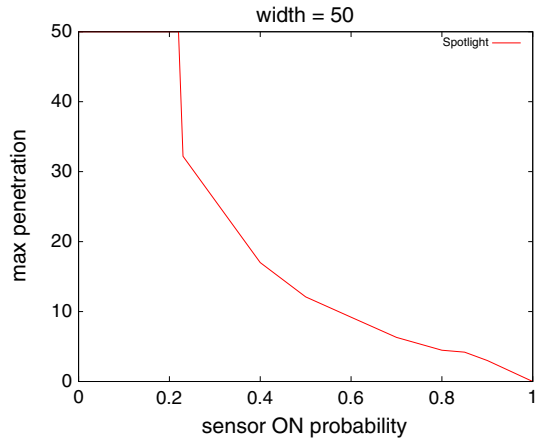
5.6. Implementation of the spotlight technique

The advantage of the RIS scheme over the Spotlight and Barrier schemes is that the latter require the subset of ON sensors at a particular time to follow a certain structural pattern. This requires a mechanism for the nodes to agree on which of them should be ON, which may require some amount of communication between them. Synchronization between nodes is critical. Care must also be taken that security is not compromised. The scheme can be realized in more than one way, each with its own advantages and shortcomings.

One method is to simulate the algorithm given in Section 5.2 over T time steps *prior to* deployment. This gives us a vector of 0's and 1's (0 meaning OFF and 1 meaning ON) of length T for each sensor. Each of these vectors is stored in the corresponding sensor. When deployed, at each time step t, a sensor simply looks up the



(a) FoI($p = 0.3$) over time (along z-axis)



(b) Position of Barrier

Figure 11. Barrier-Over-Time in spotlight.

$t\%T$th coordinate of its vector—if 0, it turns itself OFF for the time step; if 1, it turns itself ON. Because the vector contains long sequences of 0's, separated by short sequences of 1's, efficient encoding schemes may be used to reduce the amount of space required on the sensor.

Two issues need to be discussed at this point. The first one is that of synchronization—each sensor must start a logical time step at the same absolute time. A variety of possible solutions are possible. A Network Time Protocol [22] system is a hierarchical one of time servers that subscribers use to update their clocks. Some of the sensor nodes themselves may take upon the role of time server. Periodically changing the set of time servers may prove beneficial in terms of security and lifetime of the system (due to load sharing). To do this, distributed leader election protocols may be used, which require a pre-defined logical structure of the nodes such as a ring or a tree. These structures can be statically computed, or periodically carried out in the interest of robustness [23].

The second concern is that of randomness. A large value of T increases the degree of randomness perceived by the observer, reducing the chances of profiling the sleeping behavior of the network. This, however, requires a corresponding increase in the storage required. If we choose a short cycle, we save on storage but by repeating it very often, we effectively reveal the sleep schedule to the intruder. The degree of randomness in the schedule is critical—the sensor ON probability required when faced with an ideal intruder is markedly greater than that when faced with an uninformed intruder (Figure 10(a)). Resources spent in improving the degree of randomness will allow a smaller sensor ON probability, thereby increasing the lifetime of the system.

As an alternative to the vector-based approaches, the solution can be inspired by frequency hopping techniques employed in spread spectrum communications. Instead of agreeing on which frequency to use, the nodes now have to agree on who the center of the spotlight is. Example bases for the node algorithm include Reed Solomon codes [24], hyperbolic codes [25], fuzzy theory [26,27] and chaos theory [28].

Explicit synchronization is required here as well and must be addressed. However, these methods achieve greater randomness than the vector-based approach. The randomness can be further enhanced by having the sensors periodically agree to change the seed value used by the algorithm.

Another way to agree on the next spotlight center is through explicit communication. The sensors can participate in a leader election algorithm and decide on the new center. Alternatively, the current center can select the center for the next time step. The next center can also be selected by some pre-defined entity, either one of the sensors or a central base station.

Although an appreciable degree of randomness can be attained, an intruder capable of overhearing the communication and strengthening its chances is not improbable. Thus, investments have to be made in communicating quietly and securely.

6. SIMULATION METHODOLOGY

The discussed intruder models and sleep scheduling algorithms were programmed in *C*. For each configuration (a configuration comprises of FoI width, sensor ON probability, and the sleep scheduling and intruder algorithms), the simulations were performed 1000 times, and the average was taken.

While evaluating a particular configuration, the sleep scheduling algorithm is first simulated over 50 time steps. This yields a three-dimensional matrix of ON and OFF sensors. The intruder is pitted against this matrix, with the matrix being replicated during simulation to represent infinite time. The goal of the intruder, as explained in Section 3.1, is to start from the source line $\{(x, 0, 0) : x \in [L]\}$ and reach the destination plane $\{(x, W, t) : x \in [L], t \in \mathbb{N}\}$. The abilities possessed by the intruder while moving through the matrix depend on the intruder model being evaluated.

REFERENCES

1. Colin M, Beerens S. False-alarm reduction for low-frequency active sonar with BPSK pulses:

experimental results. *Oceanic Engineering, IEEE Journal of* 2011; **36** (1): 52–59, DOI 10.1109/JOE.2010.2094770.
2. Zhou S, Willett P. Submarine location estimation via a network of detection-only sensors. *Signal Processing, IEEE Transactions on* 2007; **55** (6): 3104–3115, DOI 10.1109/TSP.2007.893970.
3. Cheng V, Sridhar B. Integration of active and passive sensors for obstacle avoidance. *Control Systems Magazine, IEEE* 1990; **10** (4): 43–50, DOI 10.1109/37.56277.
4. Cheng VH, Sridhar B. Considerations for automated nap-of-the-earth rotorcraft flight, *American Control Conference*, Atlanta, GA, USA, 1988; 967–976.
5. Grimmett G. *Percolation*. Springer: Berlin, 1999.
6. Durrett R. Oriented percolation in two dimensions. *Annals Of Probability* 1984; **12** (4): 999–1040.
7. Gui C, Mohapatra P. Power conservation and quality of surveillance in target tracking sensor networks. In *Proceedings of mobicom '04*. ACM: New York, NY, USA, 2004; 129–143.
8. Gui C, Mohapatra P. Virtual patrol: a new power conservation design for surveillance using sensor networks. In *Proceedings of IPSN '05*. IEEE Press: Piscataway, NJ, USA, 2005; pp 246–253.
9. Turgut D, Turgut B, Boloni L. Stealthy dissemination in intruder tracking sensor networks, *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, 2009, Zurich, Switzerland; 22–29.
10. Huang CF, Tseng YC. The coverage problem in a wireless sensor network. *Mobile Networks and Applications* 2005; **10**: 519–528.
11. Kumar S, Lai TH, Balogh J. On k-coverage in a mostly sleeping sensor network. In *Proceedings of Mobicom '04*. ACM: New York, NY, USA, 2004; 144–158.
12. Meguerdichian S, Koushanfar F, Potkonjak M, Srivastava M. Coverage problems in wireless ad-hoc sensor networks, *Proceedings of INFOCOM 2001*, Anchorage, Alaska, USA, 2001; 1380–1387 vol.3. DOI 10.1109/INFCOM.2001.916633.
13. Zhou Z, Das S, Gupta H. Connected k-coverage problem in sensor networks, *Proceedings of ICCCN 2004*, 2004; 373–378. DOI 10.1109/ICCCN.2004.1401672.
14. Kumar S, Lai TH, Arora A. Barrier coverage with wireless sensors. In *Proceedings of Mobicom '05*. ACM: New York, NY, USA, 2005; 284–298.
15. Liu B, Dousse O, Wang J, Saipulla A. Strong barrier coverage of wireless sensor networks. In *Proceedings of Mobihoc '08*. ACM: New York, NY, USA, 2008; 411–420.
16. Chen A, Kumar S, Lai TH. Designing localized algorithms for barrier coverage. In *Proceedings of mobicom '07*. ACM: New York, NY, USA, 2007; 63–74.
17. Balister P, Zheng Z, Kumar S, Sinha P. Trap coverage: allowing coverage holes of bounded diameter in wireless sensor networks, *INFOCOM 2009, IEEE*, Rio de Janeiro, Brazil, 2009; 136–144. DOI 10.1109/INFCOM.2009.5061915.
18. Brass P. Bounds on coverage and target detection capabilities for models of networks of mobile sensors. *ACM Transactions on Sensor Networks* 2007; **3** (9).
19. Kumar S, Lai TH, Posner ME, Sinha P. Maximizing the lifetime of a barrier of wireless sensors. *IEEE Transactions on Mobile Computing* 2010; **9**: 1161–1172, DOI http://doi.ieeecomputersociety.org/10.1109/TMC.2010.78.
20. Billingsley P. *Probability and measure*. John Wiley and Sons: New York, USA, 1995.
21. Bollobás B, Riordan O. *Percolation*. Cambridge University Press: Cambridge, UK, 2006.
22. Mills D. Internet time synchronization: the network time protocol. *Communications, IEEE Transactions on* 1991; **39** (10): 1482–1493, DOI 10.1109/26.103043.
23. Gallager RG, Humblet PA, Spira PM. A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems* 1983; **5** (1): 66–77.
24. Einarsson G. Address assignment for a time-frequency-coded, spread-spectrum system. *Bell System Technical Journal* 1980; **59**: 1241–1255.
25. Maric S, Titlebaum EL. A class of frequency hop codes with nearly ideal characteristics for use in multiple-access spread-spectrum communications and radar and sonar systems. *Communications, IEEE Transactions on* 1992; **40** (9): 1442–1447, DOI 10.1109/26.163565.
26. Hangsheng Z, Hang Z, Zhongmin G. Fuzzy theory in frequency hop communication, *Communication Technology Proceedings, 1998. ICCT '98. 1998 International Conference on*, 1998; 5 vol. 1. DOI 10.1109/ICCT.1998.743245.
27. Pacini PJ, Kosko B. Adaptive fuzzy frequency hopper. *Communications, IEEE Transactions on* 1995; **43** (6): 2111–2117, DOI 10.1109/26.387452.
28. Cong L, Songgeng S. Chaotic frequency hopping sequences. *Communications, IEEE Transactions on* 1998; **46** (11): 1433–1437, DOI 10.1109/26.729385.