Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 08-026

VISA: Virtual Scanning Algorithm for Dynamic Protection of Road Networks

Jaehoon Jeong, Yu Gu, Tian He, and David Du

August 22, 2008

VISA: Virtual Scanning Algorithm for Dynamic Protection of Road Networks

Jaehoon Jeong, Yu Gu, Tian He and David Du
Department of Computer Science & Engineering, University of Minnesota
Email: {jjeong,yugu,tianhe,du}@cs.umn.edu

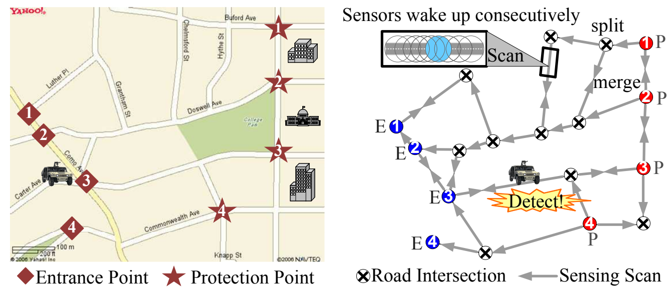
***Abstract*—This paper proposes a <u>VI</u>*rtual* <u>S</u>*canning* <u>A</u>*lgorithm* (*VISA*), tailored and optimized for road network surveillance. Our design uniquely leverages upon the facts that (i) the movement of targets (e.g., vehicles) is confined within roadways and (ii) the road network maps are normally known. We guarantee the detection of moving targets before they reach designated protection points (such as temporary base camps), while maximizing the lifetime of the sensor network. The main idea of this work is *virtual scan* – waves of sensing activities scheduled for road network protection. We provide design-space analysis on the performance of virtual scan in terms of lifetime and average detection delay. Importantly, to our knowledge, this is the first work to study how to *guarantee* target detection while sensor network deteriorates, using a novel hole handling technique. Through theoretical analysis and extensive simulation, it is shown that a surveillance system, using our design, sustains orders-of-magnitude longer lifetime than full coverage algorithms, and as much as ten times longer than legacy duty cycling algorithms.**

I. Introduction

Surveillance for critical infrastructure and areas is regarded as one of the most practical applications of wireless sensor networks (WSNs). So far, most of WSN surveillance systems have focused on surveillance for two-dimensional spaces, such as open battlefields [1]–[4]. Research on road network surveillance, however, is very limited. In modern warfare, roadways (as fast maneuver paths) are vantage areas for military surveillance and operations. Clearly, surveillance in a road network is significantly different, because (i) the movement of targets (e.g., vehicles) is confined within road segments, and (ii) the road network map is normally known (e.g., from Google Earth and Yahoo Maps). We argue that legacy solutions, which are not tailored for road networks, lead to suboptimal performance.

This paper proposes a novel sensing scheduling algorithm for target intrusion detection, utilizing the unique features of road networks. Specifically, we focus on supporting military operations with fast, infrastructure-free deployment. As shown in Figure 1(a), we guarantee the detection of targets, entering from *entrance points*, before they reach one of *protection points*; in modern warfare, battlefield situational awareness requires both entrance points and protections points (e.g., temporary base camps) to be assigned and changed on demand for fast military maneuver within a road network. Therefore, we cannot place sensor gates *a priori* before protection points for intrusion detection. Instead, a road-network-wide deployment is needed.

A straightforward solution for road network surveillance is *duty cycling*, in which nodes wake up simultaneously for w seconds (the minimum working time before reliable detection can be reported) and then the whole network remains silent for T seconds. The detection is guaranteed if it takes more than T seconds for a target to travel along the shortest path



(a) Protected Road Network (b) Concept of Virtual Scanning

Fig. 1. Road Network Surveillance

between any pair of entrance points and protection points; this duty-cycling-based algorithm performs much better in terms of system lifetime than traditional full coverage algorithms [1]–[4] in road networks. This is because the duty cycling algorithm allows the whole network to be silent completely for T seconds every w seconds, but the full coverage algorithms (e.g., the one covers all intersections) require at least one subset of sensors to be active at any given point time, taking no advantage of the linear structure of road networks.

In this paper, we present a novel scan-based algorithm, which improves further energy efficiency of surveillance in road network. As shown in Figure 1(b), sensors wake up one by one for w seconds along road segments, creating waves of sensing activities, called *virtual scanning*. Waves propagate from one (or multiple) protection point P, split at the intersections, and merge along the route until they scan all of the road segments under surveillance. Our study reveals that this scan-based method can achieve significantly better performance (e.g., ten times system lifetime) than duty cycling algorithms. The concept of virtual scanning is simple, however, in-depth design is very challenging due to a set of practical issues we consider in this paper. Particularly, we investigate (i) how to optimize the network-wide silent duration T between scan waves, (ii) how to coordinate the working schedules of individual sensors during the scan, and (iii) how to deal with sensing holes due to unbalanced initial node deployment, node failure and the depletion of node energy over time. Specifically, the intellectual contributions in this paper are as follows:

- A new architecture for surveillance in road networks. *VISA* is the first work tailored for road networks, leading to orders-of-magnitude longer system life for target intrusion detection, using a novel scan-based algorithm.
- A sensing scheduling algorithm for an arbitrary road network. The working schedule of each sensor (i.e., when to wake up) is constructed in a decentralized way. The network-wide silent duration is computed by *VISA* sched-

uler and naturally disseminated along with sensing waves to the nodes in a network.

- An optimal sensing hole handling algorithm for uncovered road segments. The *VISA* scheduler deals with both the initial sensing holes at the deployment time as well as the sensing holes due to the heterogeneous energy budget among sensors by optimally labeling additional *pseudo* protection or entrance points.

The rest of this paper is organized as follows: Section II describes the problem formulation. Section III explains the *VISA* system design. In Section IV, we discuss practical issues. Section V evaluates our algorithm through simulation. We summarize related work in Section VI and then conclude this paper in Section VII.

II. Problem Formulation

The problem is to maximize the lifetime of a sensor network, while ensuring all intruding targets are detected before they reach protection points. For clarity, this section explains the basic idea of virtual scanning, using one road segment, and then we extend our design to arbitrary road networks in Section III.

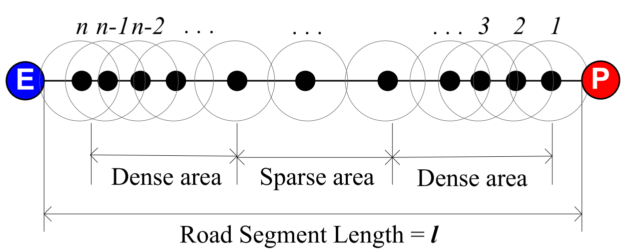


Fig. 2. Randomized Linear Deployment

A. Virtual Scanning for Surveillance

We assume n sensors are placed on a road segment of length l randomly (or uniformly). Each sensor has a nominal (conservative) sensing circle of radius r, which is long enough to over the width of the road. This assumption holds true for most commercial available sensors (e.g., PIR sensors can detect moving car 60~100 feet away). Therefore, we can represent sensing coverage using a linear sensor network model as shown in Figure 2, where n sensors are placed linearly. At the moment, let the left end of the road segment be the entrance point E of targets and the right end of the road segment be the protection point P.

Let w be the minimum working time needed by a sensor in order that the sensor can reliably detect a target over multiple samplings. Let v be a maximum target speed. Suppose that targets enter only from the entrance point and move towards the protection point. In this scenario, we can use the traditional full coverage algorithms where sensors turn on all the time. We call this approach the ***Always-Awake***.

A better design can be built based on the observation that it takes a target at least l/v seconds to pass a road segment of length l at a maximum speed v. Therefore, all sensors in the road segment can sleep *together* for l/v seconds, which is defined as *silent time* of the road network. After this *silent time*, all nodes wake up simultaneously for detection. We call this approach ***Duty Cycling***.

Based on the fact that targets move only along the roadways, we propose a new design called ***Virtual Scanning***. As shown in Figure 3, after all sensors sleep for l/v seconds, we turn sensors on one-by-one for w working time from the rightmost sensor s_1 toward the leftmost one s_n. Clearly, this wave of sensing activities guarantees the detection and allows *additional sleeping time* for individual sensors. Compared with duty-cycling, this additional sleeping time is obtained by the fact that *all sensors but one can sleep during the scan*. We note that the direction of a virtual scan shall be from the protection point to the entrance point. The virtual scan of the opposite direction (i.e., from the entrance point to the protection point) cannot guarantee target intrusion detection, if a very fast target enters right after the beginning of the network-wide silent time.

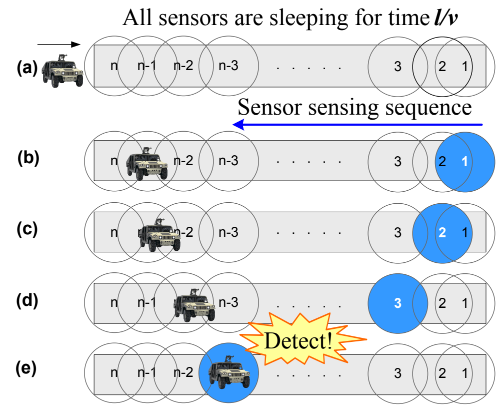


Fig. 3. Sensor Sensing Sequence

B. Analytical Network Lifetime Comparison

To understand key design parameters, this section compares analytically the network lifetime among the *always-awake*, *duty-cycling* and *virtual scanning* methods. For clarity, we summarize the notation in Table I and overall analytical results in Table II.

TABLE I
Notation of Parameters for Analysis

Parameter	Definition
T_{life}	Lifetime that a sensor can work continuously corresponding to its energy budget.
T_{net}	Sensor network lifetime.
T_{work}	Time that a sensor needs to work for reliable detection. Normally $T_{work} = w$.
T_{sleep}	Sleeping time of each sensor.
T_{scan}	Time that a virtual scanning wave moves along the road segment of length l such that $T_{scan} = nw$.
T_{silent}	Time that the whole sensor network remains silent; that is, time that a target passes through the road segment of length l. $T_{silent} = l/v$.
T_{period}	Schedule period of the sensor network. $T_{period} = T_{scan} + T_{silent}$.

Always-awake & Duty-cycling: For the *Always-awake* approach, the network lifetime T_{net} is the same as T_{life}, because sensors work continuously without sleeping. For the *Duty-cycling* approach, the network lifetime T_{net} is the number of periods $\lfloor \frac{T_{life}}{w} \rfloor$ multiplied by the length of the period T_{period}. We have:

$$T_{net} = \lfloor \frac{T_{life}}{w} \rfloor (\frac{l}{v} + w) \quad (1)$$

Virtual scanning: In the virtual scanning, the network lifetime T_{net} is the number of periods $\lfloor \frac{T_{life}}{w} \rfloor$ multiplied by the period

TABLE II
PERFORMANCE ANALYSIS FOR THREE APPROACHES

Approach	Sleeping (T_{sleep})	Working (T_{work})	Network Lifetime (T_{net})	Avg. Detection Time
Always Awake	0	T_{life}	T_{life}	0
Duty Cycling	$\frac{l}{v}$	w	$\lfloor\frac{T_{life}}{w}\rfloor(w+\frac{l}{v})$	$\frac{l^2}{2v(wv+l)}$
Virtual Scanning	$(n-1)w+\frac{l}{v}$	w	$\lfloor\frac{T_{life}}{w}\rfloor(nw+\frac{l}{v})$	$\frac{l}{2v}$

length T_{period}. T_{period} is the sum of the scan time nw and silent time $\frac{l}{v}$ as shown in Figure 4. Therefore, we have:

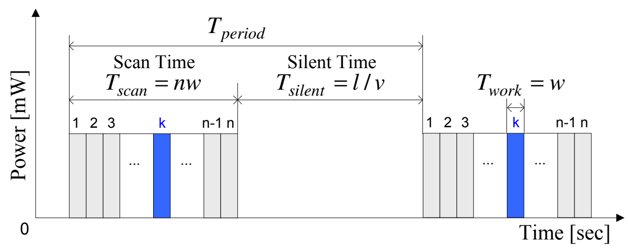


Fig. 4. Scheduling Time Diagram for Node k

$$
\begin{aligned}
T_{net} &= \lfloor\frac{T_{life}}{w}\rfloor(T_{scan}+T_{silent}) \\
&= \lfloor\frac{T_{life}}{w}\rfloor(nw+\frac{l}{v})
\end{aligned}
\tag{2}
$$

Figure 5 shows the comparison of lifetime among these three approaches. For example, for $w = 1$ sec, *Virtual Scanning* has the lifetime of 30 hours, *Duty Cycling* 3.2 hours, and *Always-Awake* 0.14 hour; *Virtual Scanning* has 9.4 times lifetime of *Duty Cycling* and 214 times lifetime of Always-Awake.

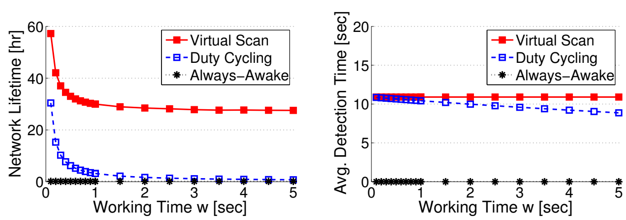


Fig. 5. Performance Comparison according to Working Time w

C. Analytical Detection Time Comparison

This section compares the average detection time after a target entering a road segment among the *always-awake*, *duty-cycling* and *virtual scanning* methods.

Always-awake & Duty-cycling: For *Always-awake*, since a target is detected as soon as it enters the road segments, the average detection time is zero. For the *Duty Cycling*, if a target enters during the working period, detection time is zero. On the other hand, if a target enters during the silent time, average detection time is half of the silent time $l/(2v)$. The percentage of silent time within a period is $l/(wv+l)$, therefore, the overall average detection time of the duty-cycling approach is $l^2/(2v(wv+l))$.

Virtual scanning: We suppose that n sensors are deployed on a road segment, so each sensor covers the length of l/n in average. Also, we suppose that target speed is v and the target can arrive at any time; that is, the arrival time is uniformly distributed. A target can arrive either during *scan time* or *silent time*. We analyze separately the average detection time for each period and then combine them to obtain overall expected delay $l/(2v)$. Please refer to Appendix A for detailed derivation.

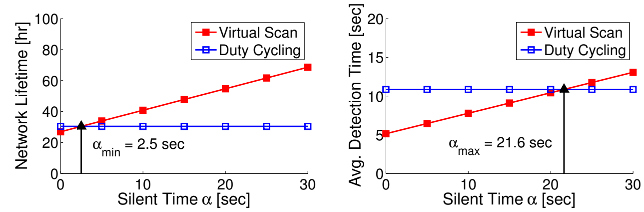


Fig. 6. Performance Comparison under Different α Values

Figure 5 shows the comparison of average detection time among the three approaches. Virtual scanning detects with a constant $l/(2v)$ delay regardless of working time w. On the other hand, the average detection time of the duty cycling tends to decrease slowly while working time w increases. The always-awake method detects without any delay. For example, for working time $w = 0.1$ sec, virtual scanning has similar performance as that of duty cycling, about 10.9 sec. For working time $w = 5$ sec, the virtual scanning detects target within 10.9 sec in average and the duty cycling does within 8.87 sec. The average detection delay ratio between the virtual scanning and the duty cycling is 1.23. However, the ratio of the virtual scanning's network lifetime to the duty cycling's network lifetime is 37, as shown in Figure 5. Thus, even though the average detection time increases slightly with virtual scanning, the benefit of network lifetime is quite remarkable.

D. Configuring VISA for Better Delay and Longer Lifetime

As a reminder, when the network silent time T_{silent} is equal to or smaller than l/v, target detection is guaranteed. Basic VISA design uses l/v as the network silent time T_{silent}. However, if a smaller silent time T_{silent} is used, it is possible to detect the target not only *faster* but also with *less* energy than the duty-cycling algorithm.

Let $T_{silent} = \alpha$ for $\alpha \in [0, l/v]$, in order to outperform duty-cycling in both network lifetime and average detection delay, we shall satisfy the following inequalities:

$$
\begin{array}{ccc}
\textbf{Virtual Scanning} & & \textbf{Duty Cycling} \\
\lfloor\frac{T_{life}}{w}\rfloor(nw+\alpha) & \geq & \lfloor\frac{T_{life}}{w}\rfloor(w+\frac{l}{v}) \\
\frac{l(nw+\alpha)}{2(nwv+l)} & \leq & \frac{l^2}{2v(wv+l)}
\end{array}
\tag{3}
$$

Solving the above inequalities, we have:

$$
\max\{\frac{l}{v}-(n-1)w, 0\} \leq \alpha \leq \min\{\frac{l(nwv+l)}{v(wv+l)}-nw, \frac{l}{v}\}
$$

When α falls into this range, virtual scanning has better performance than duty cycling in both the average detection time and network lifetime. For example, as shown in Figure 6, for $w = 0.1$ sec, when α is less than $\alpha_{max} = 21.6$ sec, the average detection time of virtual scanning is shorter than that of duty cycling. Also, when α is greater than $\alpha_{min} = 2.5$ sec, virtual scanning's lifetime is longer than that of duty cycling. Thus, the range of α achieving better detection delay and lifetime is [2.5, 21.6] sec. We note the results here only illustrate the idea. Detailed study on the performance effect of α is presented in evaluation Section V-B3.

III. Virtual Scanning Algorithm System Design

For the sake of clarity, the previous section presents the basic idea using one road segment. In the rest paper, we demonstrate how to apply the virtual scanning to road networks with arbitrary topology. This section is organized as follows: Section III-A lists definitions and assumptions used in VISA. Section III-B describes the scheduling algorithm, and Section III-C presents the hole handling algorithm.

A. Definitions and Assumptions

Definition 3.1 (Road Network Graph): Let a road network graph be $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of intersections, entrance points, and protection points in the road network under surveillance, and $E = [e_{ij}]$ is a matrix of road segment length e_{ij} for vertices v_i and v_j. Figure 7 shows a graph G corresponding to the road network in Figure 1.

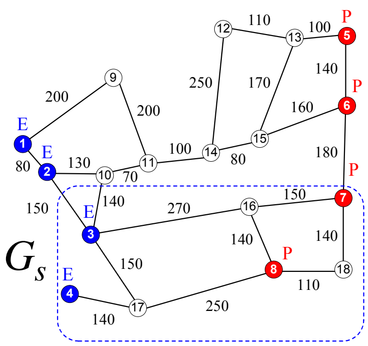


Fig. 7. Road Network Graph G

Definition 3.2 (Network Lifetime): Let *Network Lifetime* be the duration from the starting of a sensor network for surveillance until a target can *possibly* reach one of the protection points without detection. In other words, lifetime ends when there exists a possible breach path between an entrance point to a protection point.

Definition 3.3 (VISA Scheduler): Let *VISA Scheduler* be a sink node that initiates the sensing scheduling algorithm.

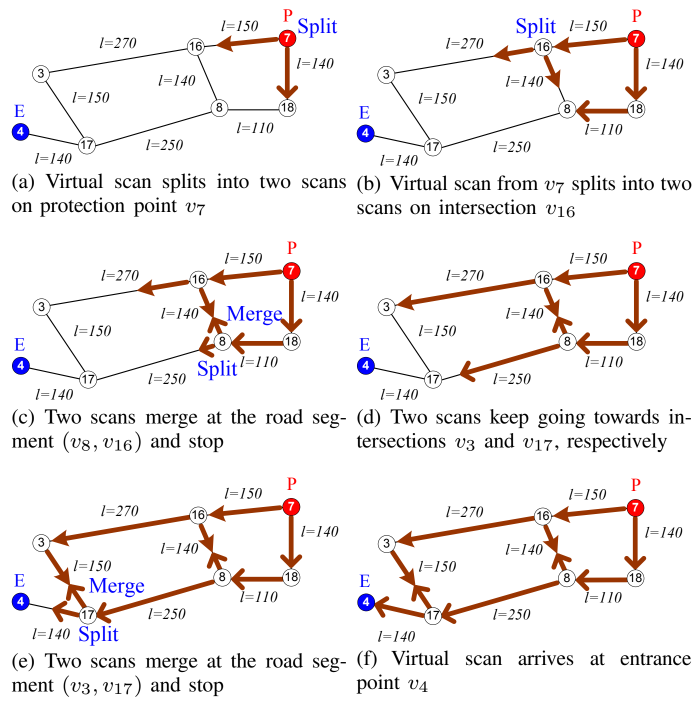
The VISA design is based on the following assumptions:

- Road map and locations of sensor nodes are known to *VISA Scheduler*. The sensor location can be obtained through localization schemes [5].
- Sensors are roughly time-synchronized at tens of millisecond level. It can be easily achieved because existing solutions [6], [7] can achieve microsecond level accuracy.
- Sensors only have simple sensing devices for binary target detection, such as PIR sensors [8]. No sophisticated hardware is available.
- One of existing low-duty-cycle data forwarding schemes, such as DSF [9] and DESS [10] are used to deliver nodes' locations and target detection results to the VISA scheduler.
- Targets move only along predefined roads with the bounded maximum speed.

B. VISA Scheduling on Road Network

This section presents the design of virtual scanning, including schedule establishment and dissemination.

1) Establishment of Working Schedule: For clarity in presentation, we use the subgraph G_s of the graph G shown in Figure 7 where the edge weight means the physical distance of the road segment. First, we will consider a road network with one entrance and one protection point at first, and then will consider a road network with multiple entrance and multiple protection points. Also, for now, we assume that no sensing holes exist in the middle of roadways where targets cannot be detected due to the non-existence of sensors. The sensing hole handling will be discussed in Section III-C.



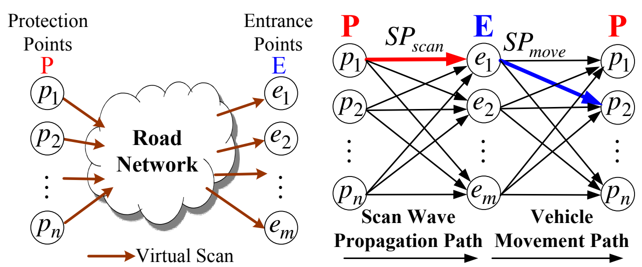
(a) Virtual scan splits into two scans on protection point v_7
(b) Virtual scan from v_7 splits into two scans on intersection v_{16}
(c) Two scans merge at the road segment (v_8, v_{16}) and stop
(d) Two scans keep going towards intersections v_3 and v_{17}, respectively
(e) Two scans merge at the road segment (v_3, v_{17}) and stop
(f) Virtual scan arrives at entrance point v_4

Fig. 8. Virtual Scanning on Road Network for Working Schedule Establishment

Figure 8 shows snapshots of virtual scanning in this road network with one entrance v_4 labeled with *E* and one protection point v_7 labeled with *P*. The virtual scan's propagation time on each road segment is the multiplication of the number of sensors and the individual working time w, instead of the physical distance of a road segment. As shown in Figure 8, by turning on nodes along roads consecutively, virtual scanning waves propagate along multiple routes simultaneously, split at intersections, and disappear when two waves encounter each other in a road segment.

In the case of multiple entrance and protection points, scan operation is similar, except that multiple protection points initiate scanning at the same time. Because the waves merge into each other in virtual scanning, regardless the number of protection points and the locations of the nodes, each node only works for w second per scan, which is a nice feature for energy balance. Clearly, the scan wave arrival time for each sensor can be easily computed with All-Pairs Shortest Path algorithm, such as *Floyd-Warshall algorithm* [11]. We note the scan wave arrival time decides the working schedule of a sensor node. In other words, a sensor shall start to work for w seconds after a virtual scanning wave arrives.

2) Decentralized Implementation: In a centralized implementation, a VISA scheduler calculates the work schedules for all nodes and disseminate the results, which leads to far more messages than necessary. Actually the scan wave arrival time for each sensor can be calculated in a decentralized way. During the initialization phase, all sensors are awake. The sensors at the protection points generate a short message containing a counter with value initialized to one, and pass them to their immediate neighboring sensors. The neighboring sensors only record the minimum counter value ever seen (discard the rests), increment the counter, and then relay the message to their neighboring sensors. If a sensor is located at a road intersection, it duplicates and relays multiple copies of messages to the all its neighboring nodes except the one it received the message from. In this way, the sensors can decide their sensing scanning order (i.e., the minimum counter value) in the distributed way. Given a sensing order of K, a node shall start to work at time Kw and stop at time $(K+1)w$.



(a) Virtual Scanning on an Arbitrary Road Network with Protection Points and Entrance Points

(b) Sleeping Time Computation considering the Shortest Scan and Movement Paths

Fig. 9. Virtual Scanning on Road Networks

3) Establishment of Sleeping Schedule: The previous section discussed how to decide working schedule during the scan. In this section, we will explain how to compute the optimal sleeping length, i.e., the maximum duration sensors can sleep safely after working for w seconds while guaranteeing the detection.

Figure 9 shows the virtual scanning in an arbitrary road network. Let $P = \{p_1, ..., p_n\}$ be the set of protection points. Let $E = \{e_1, ..., e_m\}$ be the set of entrance points. As discussed before, a period T_{period} consists of (i) silent time T_{silent} during which whole network are turned off and (ii) scan time T_{scan} during which scan waves propagate across the network. Since a node only works for fixed $T_{work} = w$ seconds every T_{period}, the longer T_{period} is, the better energy efficiency we have. Therefore, we shall identify the maximum T_{period} value that can guarantee the detection. Before this optimization, we define two important concepts as below:

Definition 3.4 (Shortest Scanning Path): The Shortest Scanning Path $p_{scan}(i,j)$ is the shortest-delay path for wave propagation from v_i to v_j on the graph G, where $v_i \in P$ and $v_j \in E$. Let $l_{scan}(i,j)$ be the number of sensors along the path $p_{scan}(i,j)$. Therefore, the Shortest Scanning Time $T_{scan}(i,j)$ can be computed as $l_{scan}(i,j) * w$.

Definition 3.5 (Shortest Movement Path): The Shortest Movement Path $p_{move}(i,j)$ is the shortest-distance path between vertices v_i and v_j on the virtual graph G where $v_i \in E$ and $v_j \in P$. Let $l_{move}(i,j)$ be the shortest distance. Therefore, the Shortest Movement Time $T_{silent}(i,j)$ can be computed as $l_{move}(i,j)/v_{max}$, where v_{max} is maximum target speed. We note that all of the sensors along the path $p_{move}(i,j)$ can sleep together for the silent time $T_{silent}(i,j)$.

Two shortest paths $p_{scan}(i,j)$ and $p_{move}(i,j)$ for all pairs of vertices can be computed based on G by the All-Pairs Shortest Paths algorithm, such as *Floyd-Warshall algorithm*.

An important principle of computing the optimal sleeping time is that all of vehicles entering during the sleeping time must be detected before their arrival to the protection points. Once a virtual scan wave originating from the protection points have swept an entrance point, the paths from this swept entrance point to the protection points are vulnerable to the target intrusion. This is because the swept paths are not swept again until the next scan period.

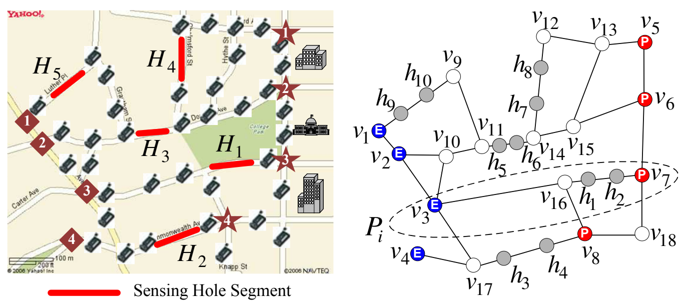
It is noted that we can guarantee detection by setting T_{period} as the sum of **all-pair minimum scanning time** and **all-pair minimum target movement time**. However the resulting T_{period} is shorter than the optimal value, because an intruding target could have to travel a *long* route from an entrance point with the *earliest* scan arriving time, or could have to wait until a *late* scan arrives before it can travel along the *shortest* route, especially when nodes are placed non-uniformly across a network. Therefore, the optimal safe T_{period} shall be the **minimum sum** of the **scanning time** from v_i to v_j and the **vehicle movement time** from v_j to v_k, for $v_i, v_k \in P$ and $v_j \in E$.

Figure 9(b) shows a three-column graph for computing the period T_{period}. The edges between first and second column denote the time for wave propagation and the edges between the second and third column denote the time for target movement. To compute a safe and optimal T_{period}, we need to identify the shortest path from any vertex in the first column to any vertex in the third column. Without loss of generality, suppose $p_1 \Rightarrow e_1 \Rightarrow p_2$ is the shortest path. Once the virtual scanning arrives at the entrance point e_1 with a delay of $T_{scan}(p_1, e_1)$, the path from the entrance point e_1 to the protection point p_2 becomes vulnerable, if the network remains silent for more than $T_{slient}(e_1, p_2)$. Thus, to prevent a target from reaching the protection point p_2, another scan wave must be generated from the protection point p_2 after $T_{slient}(e_1, p_2)$. Therefore, the safe and optimal $T_{period} = T_{scan}(p_1, e_1) + T_{silent}(e_1, p_2)$. Consequently, the sleeping time $T_{sleep} = T_{period} - T_{work}$, because each sensor must work for its duty cycle $T_{work} = w$ per period.

Now, we can formally define the optimization problem of the sleeping time. Let $T_{sleep}(i,j,k) = T_{scan}(i,j) + T_{silent}(j,k) - T_{work}$ for $v_i, v_j, v_k \in V[G]$ where $T_{work} = w$. The optimal sleeping time is chosen as follows:

$$T_{sleep} \leftarrow \min_{\substack{v_i, v_k \in P, \\ v_j \in E}} T_{sleep}(i,j,k). \tag{4}$$

Obviously, the searching for an optimal sleeping time is done in polynomial time $O(mn^2)$. Once the sleeping time value is computed by *VISA scheduler*, it piggybacks in the counter message discussed in Section III-B2 and is disseminated to all



(a) Sensing Hole Segments on Road Network: $H_i = (h_j, h_k)$

(b) Augmented Graph including Sensing Holes: $G_a = (V_a, E_a)$

Fig. 10. Augmentation of Road Network Graph with Sensing Holes

the sensors in the network. If the VISA scheduler changes the locations of protection and entrance points dynamically, it only needs to re-calculate a new sleeping time and re-disseminate it.

Till now, the sensors know when to wake up in order to create virtual scanning (i.e., Working Schedule in Section III-B1) and how long they can safely sleep with optimal efficiency (i.e., Sleeping Schedule in Section III-B3).

C. Handling of Sensing Holes

We have so far discussed the sensor working schedule and sleeping schedule, assuming balanced energy and no initial sensing holes. In this section, we discuss the handling of sensing holes that can exist after the sensor deployment and that can occur due to sensor failure or energy depletion. As shown in Figure 10(a), five sensing hole segments (i.e., H_1,..., H_5) exist in the given road network graph. Our idea to deal with these initial hole segments is that we make an augmented graph by adding the endpoints of the hole segments as shown in Figure 10(b). To ensure the protection, we treat these endpoints as either *pseudo* entrance points or *pseudo* protection points. The hole handling problem is, therefore, reduced to a labeling problem of hole segment endpoints.

Problem Definition: How to optimally determine the role of each hole endpoint (i.e., label as entrance point or protection point) in order to achieve the maximum sleeping time, leading to the maximization of the sensor network lifetime.

In the rest of this section, we present an optimal labeling algorithm for hole handling.

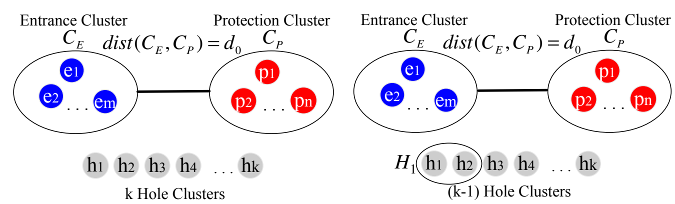
1) Initial Sensing Holes: In reality, there is high probability that some road segments are not covered by sensors even though many sensors are randomly deployed on road network as shown in Figure 10(a). We define these uncovered road segments as the *initial sensing hole segments*; note that each sensing hole segment consists of two hole endpoints.

Suppose that n hole endpoints occur under a uniform sensor density. With an exhaustive search, 2^n cases are required to investigate. This means the time complexity of $O(2^n)$. Since this complexity is intractable, we need an improved way to achieve an optimal labeling for hole endpoints.

We explain here the idea with a simplified example; Figure 10(b) shows one roadway P_i consisting of v_3, v_{16}, and v_7 and a hole segment H_1 with hole endpoints h_1 and h_2, which are closer to a protection point v_7 than an entrance point v_3. If two hole endpoints h_1 and h_2 are labeled differently, this short hole segment determines the shortest sleeping time. To avoid this, h_1 and h_2 should have the same type of label. Furthermore, since h_1 and h_2 near the protection point v_7, in order to get a longer sleeping time, they should be labeled as protection points.

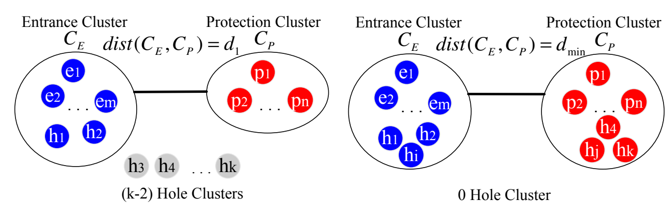
Conceptually, when labeling hole endpoints, we should label each hole endpoint with the same label as the closest point that is already labeled. Rationale behind this insight is: the maximization of the distance between the entrance points and protection points leads to a maximum sleeping time according to Eq. 4.

Formally, let H be the set of hole endpoints such that $H = \{h_1, h_2, \ldots, h_k\}$. Let E be the set of entrance points. Let P be the set of protection points. Let L_H be hole label, L_E be entrance label, and L_P be protection label. We can label the holes in the set H, by partitioning H into two disjoint subsets (called *clusters*) Entrance Cluster (C_E) and Protection Cluster (C_P). Asano et al. proposed such a clustering algorithm for a farthest k-partition based on Minimum Spanning Tree (MST) [12], giving an *optimal clustering* to maximize the inter-cluster distance. We extend *Asano's Clustering* for sensing hole labeling.



(a) Initial Clusters: Entrance Cluster C_E, Protection Cluster C_P, and k Hole Clusters h_s for $s = 1..k$

(b) The closest pair of h_1 and h_2 are merged into cluster H_1 labeled as unknown



(c) The closest pair of C_E and H_1 are merged into C_E with H_1 labeled as entrance

(d) After clustering, all hole clusters are merged into either C_E or C_P, that is, labeled as either L_E or L_P

Fig. 11. Clustering for Sensing Hole Labeling

Figure 11 illustrates the main idea. Let $dist(C_E, C_P)$ be the inter-cluster distance between C_E and C_P. Our objective is to partition the set H into two disjoint sets C_E and C_P such that the inter-cluster distance between C_E and C_P is maximized. The initial inter-cluster distance is $dist(C_E, C_P) = d_0$, as shown in Figure 11(a). In this example, suppose that two hole clusters h_1 and h_2 consists of the closest pair of two clusters. In this case, these hole clusters are merged into one hole cluster H_1 with the same, unknown label, as shown in Figure 11(b). The reason two clusters h_1 and h_2 are merged into one hole cluster with the same label is to let the inter-cluster distance between C_E and C_P be maximized. Otherwise, the inter-cluster distance between h_1 and h_2 can make the inter-cluster distance shorter than the initial inter-cluster distance $dist(C_E, C_P) = d_0$. As shown in Figure 11(c), two clusters C_E and H_1 are the closest pair, so H_1 is merged into C_E with hole endpoints h_1 and h_2 labeled as entrance. In this way, we can cluster all of the hole endpoints into either C_E or C_P to maximize the inter-cluster distance $dist(C_E, C_P)$, as shown in Figure 11(d). Similar to Asano's algorithm [12], our clustering gives an optimal hole labeling because it satisfies the *greedy choice property* and *optimal substructure* [11].

As an important difference from *Asano's Clustering*, during the clustering, we maintain multiple hole clusters H_i labeled as *unknown* in addition to one Entrance Cluster C_E and one Protection Cluster C_P. Through the MST construction, we merge one hole cluster H_i to either C_E or C_P such that the inter-cluster distance between C_E and C_P is maximized. We call this new labeling algorithm the *MST-based Labeling*.

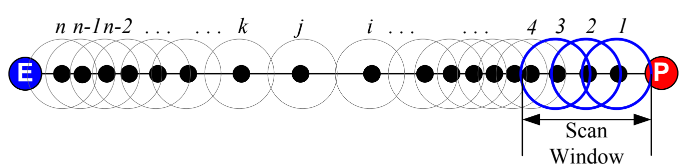
2) Sensing Holes due to Energy Depletion or Failure: In the previous section, we discussed the initial sensing hole issue. However, since in reality, the sensors deployed on road network may not have the same amount of energy initially, we need to consider the sensing holes caused by this unbalanced sensor energy budget. Also sensor could fail over time. We can deal with these sensing holes in the same way as with the initial holes; we can either completely relabel all holes or incrementally relabel new holes by using *MST-based Labeling*. The former is optimal, but the latter introduces less computation.

IV. Practical Issues

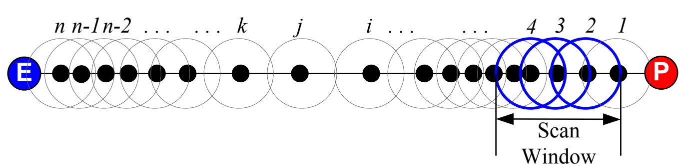
In this section, we consider three practical issues for the deployment of our *VISA* system in real road networks: (i) Detection error probability, (ii) Time synchronization error, and (iii) Communication design for detection report.

A. Detection Error Probability

In reality, there exists sensing error in sensor node. We need to relax the assumption that every vehicle within the sensing range of some sensors can be detected with probability one. Let p be the sensing failure probability of in each sensor. As shown in Figure 12, there exist n sensors. In order to reduce the detection failure under this condition, we perform the virtual scan consisting of multiple sensors, which constructs *scan window*. As shown in Figure 12(a), the right-most k sensors (i.e., s_1, s_2, and s_3) turn on together and work for their duty cycle w. After w, the *scan window* moves to the left, letting the next sensor s_4 turn on and the right-most sensor s_1 in the current scan window turn off, as shown as Figure 12(b).



(a) Scan Window consisting of Sensors s_1, s_2, and s_3



(b) Scan Window consisting of Sensors s_2, s_3, and s_4

Fig. 12. Moving Window for Detection Error Handling

The failure probability $P_{fail}(k)$ of *scan window size* k (i.e., the probability that a vehicle passes this scan window without being detected) is p^k. Thus, according as we increase the *scan window*'s size k, the probability $P_{fail}(k)$ will be very small.

B. Time Synchronization Error

Sensors in *VISA system* are roughly time-synchronized as long as there is no time gap between two neighboring sensors during the scan time for vehicle detection. Many state-of-art solutions [6], [7] can provide sensors with the time synchronization at the microsecond level. When a maximum time error is known as ϵ_t, each sensor is required to have a margin of ϵ_t for its working start time t_s and working end time t_e such that the working schedule is $[t_s - \epsilon_t, t_e + \epsilon_t]$. This guarantees duty cycle overlap with its neighboring sensors.

C. Communication Design for Detection Report

We assume that sensors deployed on a target road network can construct an ad-hoc network for the detection report delivery to the *VISA scheduler*. We also suppose that multiple sink nodes are located near by entrance points and protection points and the sink nodes can communicate with each other through wired or wireless links. In this setting, during the virtual scanning for road surveillance, sensors wake up earlier and sleep later with some margin of time than its original working schedule. The margin is set up to guarantee that two neighboring sensors can exchange messages during the scanning as in the case of time synchronization error handling. In other ways, for the quicker delivery of target detection results, we can use existing low-duty-cycle data forwarding schemes, such as DSF [9] and DESS [10], considering sensor working schedules.

V. Performance Evaluation

In this section, we analyze performance of VISA, comparing with other schemes for road network surveillance.

- **Performance Metrics:** We use *network lifetime* and *average detection time* as the performance metrics.
- **Baselines:** Since the road network surveillance is a new research area, to the best of our knowledge, there exist no other state-of-art sensing schemes for road network surveillance. We compare VISA with two approaches: *Duty Cycling* and *Always-Awake*.
- **Parameters:** In the performance comparison, we investigate the effect of the following three parameters: (i) working time w, (ii) sensor density, and (iii) energy budget. In addition, we reveal (i) effect of sleeping time duration and (ii) effect of sensing hole labeling.

Simulation uses the map of a real road network as shown in Figure 7. The system parameters are selected based on a typical military scenario [13]. Unless mentioned otherwise, the default values in Table III are used.

For network lifetime measurement, the default energy budget (50 kJ) is used, but for the average detection time measurement, to obtain high statistical confidence, a full-day energy budget is used for the comparison among three approaches: (i) *Virtual Scanning*, (ii) *Duty Cycling*, and (iii) *Always-Awake*. The vehicle arrival time is uniformly distributed during the system lifetime with mean inter-arrival time 60 sec.

TABLE III
SIMULATION CONFIGURATION

Parameter	Description
Sensing range R	$R = 2r =$ 20meters (i.e., 66feet) where r is sensing radius.
Sensor working time w	Nine points in [0.1,0.9] with step time 0.1sec and nine points in [1,5] with step time 2.5sec. The default of W is 1sec.
Sensor density d	$d \sim N(\mu_d, \sigma_d^2)$ where $\mu_d = \{2, 4, ..., 20\}$ and $\sigma_d = \{0, 1, ..., 6\}$. The default of (μ_d, σ_d) is $(10, 0)$.
Energy budget b	$b \sim N(\mu_b, \sigma_b^2)$ where $\mu_b =$ 50kilo-joule (kJ) and $\sigma_b = \{0, 2, ..., 18\}$kJ. The default of σ_b is 5kJ.
Vehicle speed v	$v \sim N(\mu_v, \sigma_v^2)$ where $\mu_v = \{15, 20, ..., 60\}$MPH and $\sigma_v = \{0, 5\}$MPH. Maximum speed is 70MPH and minimum speed 10MPH. The default of (μ_v, σ_v) is $(40, 0)$.

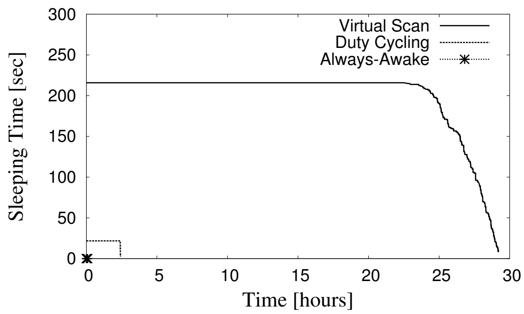


Fig. 13. Comparison of Sleeping Time T_{sleep}

A. System Behavior Over time

All three methods *Virtual Scanning*, *Duty Cycling* and *Always-Awake* can guarantee the detection of targets. Their difference lies in the network lifetime. Clearly, the longer a node can sleep safely per period, the more energy efficiency is. Figure 13 shows how the sleeping time T_{sleep} changes before network lifetime ends. As shown in the figure, *Virtual Scanning* has by far the largest sleeping time and hence the longest network lifetime. For example, *Virtual Scanning* sustains for 29.2 hours, compared with 2.4 hours in *Duty Cycling* and 5.4 minutes in *Always-Awake*. This is because of the significant energy saving during the scanning process.

B. Performance Comparison

In this section, we compare three approaches: (i) *Virtual Scanning*, (ii) *Duty Cycling* and (iii) *Always-Awake* in terms of Network Lifetime and Average Detection Time under several user-level parameters, such as working time duration, energy budget, and sensor density.

1) The Impact of Working Time w : Since w is the minimum working time before reliable detection can be reported, this evaluation reveals how different hardware response speeds and sensing algorithms affect the VISA and other baselines. We use non-uniform $50kJ$ energy budget with the energy variation $5kJ$. Clearly, VISA provides significantly longer system lifetime than the baselines, especially when w is large as shown in Figure 14(a). For example, when w is 1 second, VISA extends network lifetime by 12 times, compared with *Duty Cycling* and 158 times, compared with *Always-Awake*. As shown Figure 14(b), the average detection time of *Virtual Scanning* is about 11.75 sec, which is slightly longer than that of duty cycling. This is because we set the silent time exact as the target moving time along the shortest path.

2) Impact of Sensor Density: As we expected from the formula of the network lifetime in Eq. 2, the high sensor density provides the longer network lifetime for *Virtual Scanning*. This is because with a higher density, we have a longer scanning time T_{scan}, which allows sensor nodes to sleep longer. However, the high sensor density does not contribute much to the network lifetime to *Duty Cycling* and *Always-Awake*, since their sleeping time is independent of the number of sensors. For the average detection time, in both *Virtual Scanning* and *Duty Cycling*, e.g., under sparse sensor density less than 8, the lower density lets the sensors close to entrances detect vehicle earlier. This is because many sensor network clusters occur due to initial sensing holes, so the sleeping time becomes short. Thus, the sensors close to entrances wake up early and detect targets, leading to shorter detection time. In summary, at all sensor density settings, *Virtual Scanning* provides the longest network lifetime with a slight increase in detection time. The performance gain of *Virtual Scanning* is also higher when sensor density becomes higher.

3) Achieving Shorter Delay and Longer Lifetime Simultaneously: In Section II-D, we showed analytically how VISA achieves a shorter delay and a longer network lifetime simultaneously by adjusting the *silent time* ($T_{silent} = \alpha$) within the range that satisfies Eq. 3. To confirm our design empirically, Figure 16 shows the performance effect of *Virtual Scanning* according to α. For example, as shown in Figure 16, when *Virtual Scanning* reduces α from T_{silent} to $T_{silent}/2$ in the working time interval $[0.1, 0.5]$, it has better performance in both the network lifetime and average detection time than *Duty Cycling*.

C. The Effect of Hole handling

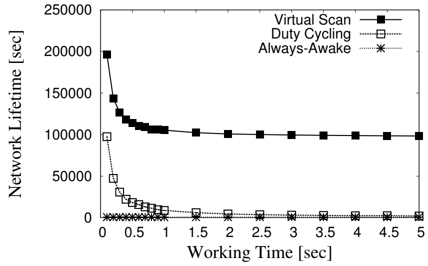
This section compares three different methods for hole handling as follows:

- *MST-based Labeling*: our hole labeling scheme discussed in Section III-C.
- *Random Labeling*: a new hole is randomly labeled with either *pseudo* entrance point or *pseudo* protection point.
- *No Labeling*: when a new hole occurs, it is not handled, leading to the end of system lifetime.

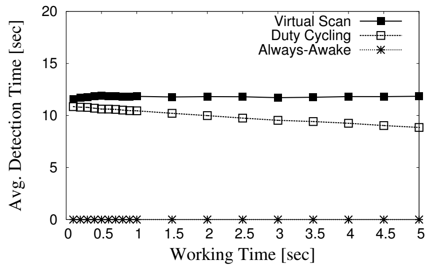
We use the same *Virtual Scanning* for these three labeling algorithms. As shown in Figure 17, *MST-based Labeling* gives longer lifetime than both *Random Labeling* and *No Labeling*. *Random Labeling* and *No Labeling* have the similar lifetime, because *Random Labeling* cannot label holes appropriately to prevent a breach path (i.e., path vulnerable to vehicle intrusion to protection points) from existing. Since *No Labeling* does not handle sensing hole, one sensing hole creates a breach path, leading to the end of system. For the average detection time, these three labeling algorithms have similar performance whose curves are the same as the curve of *Virtual Scanning* in Figure 14(b).

VI. RELATED WORK

Most research on coverage for detection has so far focused on *Full Coverage* [1]–[4], [14]–[18] in a two-dimensional space. In [4], authors use the off-duty eligibility rule to turn on/off a node as long as the neighboring nodes can cover the sensing area of this node. The Coverage Configuration Protocol (CCP) [19] provides an energy-efficient sensing coverage, integrated with SPAN for connectivity. In [20], surveillance

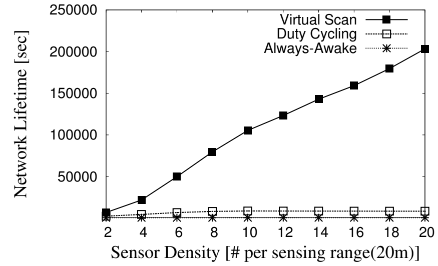


(a) Network Lifetime vs. Working Time

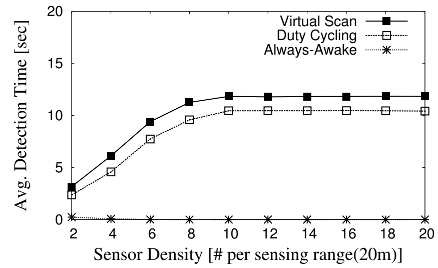


(b) Avg. Detection Time vs. Working Time

Fig. 14. The Impact of Working Time w

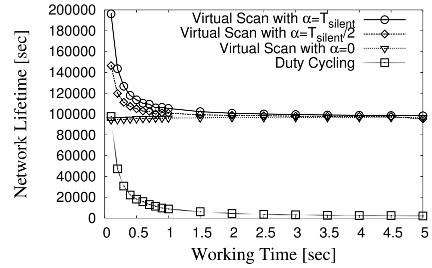


(a) Network Lifetime vs. Sensor Density

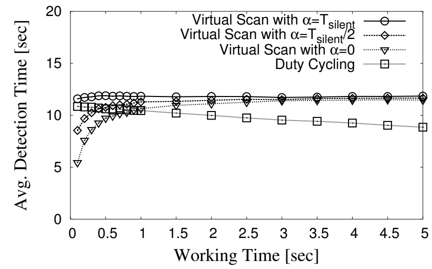


(b) Avg. Detection Time vs. Sensor Density

Fig. 15. The Impact of Sensor Density



(a) Network Lifetime vs. Working Time



(b) Avg. Detection time vs. Working Time

Fig. 16. The Impact of Sleep Time α

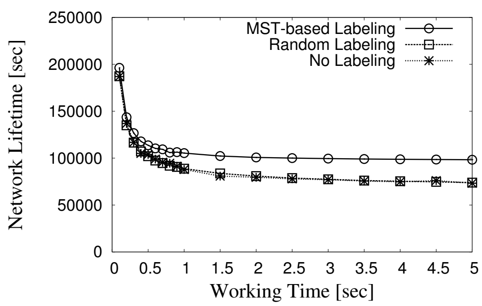


Fig. 17. Performance Comparison of Hole Labeling Algorithms

coverage is achieved through probing. DiffSurv [21] provides differentiated surveillance to an area with a certain degree of coverage, up to the limitation imposed by the number of sensor nodes deployed. Kumar et al. [3] identify a critical bound for k-coverage in a network, assuming a node is randomly turned on with a certain probability. In [2], Cardei et al. propose two heuristic algorithms to identify a maximum number of set covers to monitor a set of static targets at known locations. In [1], Abrams et al. propose three approximation algorithms for a relaxed version of the previously defined SET K-COVER problem [22].

To aggressively reduce energy consumption, partial coverage through *Duty Cycling* has been studied as well. In [23], [24], authors provide a theoretical analysis and simulation on the delay (or stealth distance) before a target is detected. In [23], the Quality of Surveillance (QoSv) is defined as the reciprocal value of the expected travel distance before mobile targets are first detected by any sensor. In [25], nodes coordinate among each other to guarantee the worst-case detection delay and minimize the average detection delay. In [26]–[28], the theoretical foundations for laying barriers with stealthy and wireless sensors are proposed in order to detect the intrusion of mobile targets approaching the barriers from the outside.

The closest related work is *virtual patrol* [29], in which a *virtual patrol* moves along the predefined path in 2-dimensional space and triggers sensors adjacent to the virtual patrol's path for detection. This virtual patrol is similar to the concept of our virtual scan. However, the uniqueness of our work can be clearly identified from the following respects: (i) our work focuses on surveillance in road network, where legacy two-dimensional solutions cannot directly apply, and (ii) we are the first to formally guarantee target detection while sensor network deteriorates, using a hole handling technique.

VII. Conclusion

Specially tailored for road networks, this work introduces *VISA* based on the concept of virtual scanning. *VISA* propagates sensing waves along the roadways and detects vehicles entering into the target road network before they reach the protection points. We demonstrate analytically and empirically the feasibility of achieving longer network lifetime and shorter detection delay simultaneously. In addition, we propose an optimal algorithm to deal with the initial sensing holes at the deployment time as well as the sensing holes due to node failure and the heterogeneous energy budget among sensors by optimally labeling additional *pseudo* protection or entrance points. Evaluation shows orders-of-magnitude longer network lifetime than the always-awake method, and as much as ten times longer than the duty cycling algorithms. We believe this work opens a promising direction of road network surveillance. Future works, for examples, include (i) the perimeter protection of road networks, (ii) protection design with bounded detection delay and (iii) optimal sensor placement with minimal detection delay.

Acknowledgment

This research was supported by the Department of Computer Science and Engineering, and the Digital Technology Center at the University of Minnesota, and also in part by NSF grant CNS-0626614, CNS-0615063, and CNS-0626609.

References

[1] Z. Abrams, A. Goel, and S. Plotkin, “Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks,” in *IEEE IPSN*, 2004.

[2] M. Cardei, M. T. Thai, Y. Li, and W. Wu, “Energy-Efficient Target Coverage in Wireless Sensor Networks,” in *IEEE INFOCOM*, 2005.

[3] S. Kumar, T. H. Lai, and J. Balogh., “On k-coverage in a mostly sleeping sensor network,” in *Mobicom*, 2004.

[4] D. Tian and N. Georganas, “A Node Scheduling Scheme for Energy Conservation in Large Wireless Sensor Networks,” *Wireless Communications and Mobile Computing Journal*, May 2003.

[5] A. Savvides, C. C. Han, and M. B. Srivastava, “Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors,” in *MOBICOM*. Rome, Italy: ACM, Jul. 2001.

[6] J. Elson, L. Girod, and D. Estrin, “Fine-Grained Network Time Synchronization using Reference Broadcasts,” in *OSDI*. ACM, Dec. 2002.

[7] M. Maróti, B. Kusy, G. Simon, and Ákos Lédeczi, “The Flooding Time Synchronization Protocol,” in *SENSYS*. Baltimore, Maryland, USA: ACM, Nov. 2004.

[8] Lin Gu et al., “Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments,” in *SENSYS*. San Diego, California, USA: ACM, Nov. 2005.

[9] Y. Gu and T. He, “Data Forwarding in Extremely Low Duty-Cycle Sensor Networks with Unreliable Communication Links,” in *SENSYS*. Sydney, Australia: ACM, Nov. 2007, pp. 321–334.

[10] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, “Delay Efficient Sleep Scheduling in Wireless Sensor Networks,” in *INFOCOM’05*, 2005.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms (2nd Edition)*. MIT Press, 2003.

[12] T. Asano, B. Bhattacharya, M. Keil, and F. Yao, “Clustering Algorithms based on Minimum and Maximum Spanning Trees,” in *Proceedings of the Fourth Annual Symposium on Computational Geometry*, 1988, pp. 252–257.

[13] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, “An Energy-Efficient Surveillance System Using Wireless Sensor Networks,” in *MobiSys’04*, June 2004.

[14] W. Wang, V. Srinivasan, and K.-C. Chua, “Trade-offs between Mobility and Density for Coverage in Wireless Sensor Networks,” in *MOBICOM*. ACM, Sep. 2007.

[15] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, “Coverage problems in wireless ad-hoc sensor networks,” in *INFOCOM*. Anchorage, Alaska: IEEE, Apr. 2001.

[16] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration in wireless sensor networks,” in *SENSYS*. Los Angeles, CA, USA: ACM, Nov. 2003.

[17] H. Zhang and J. Hou, “Maintaining sensing coverage and connectivity in large sensor networks,” *Ad Hoc & Sensor Wireless Networks*, vol. 1, Mar. 2005.

[18] S. Shakkottai, R. Srikant, and N. Shroff, “Unreliable sensor grids: Coverage, connectivity, and diameter,” in *INFOCOM*. San Francisco, CA, USA: IEEE, Apr. 2003.

[19] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks,” in *First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, November 2003.

[20] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, “PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks,” in *ICDCS*. IEEE, May 2003.

[21] T. Yan, T. He, and J. Stankovic, “Differentiated Surveillance Service for Sensor Networks,” in *SENSYS*. Los Angeles, CA, USA: ACM, Nov. 2003.

[22] S. Slijepcevic and M. Potkonjak, “Power Efficient Organization of Wireless Sensor Networks,” in *IEEE ICC*, 2001.

[23] C. Gui and P. Mohapatra, “Power conservation and quality of surveillance in target tracking sensor networks,” in *MOBICOM*. Philadelphia, PA, USA: ACM, Sep. 2004.

[24] S. Ren, Q. Li, H. Wang, X. Chen, and X. Zhang, “Analyzing Object Tracking Quality under Probabilistic Coverage in Sensor Networks,” *ACM Mobile Computing and Communications Review*, vol. 9, no. 1, January 2005.

[25] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, “Towards Optimal Sleep Scheduling in Sensor Networks for Rare Event Detection ,” in *The Fourth International Conference on Information Processing in Sensor Networks*, 2005.

[26] S. Kumar, T. Lai, and A. Arora, “Barrier coverage with wireless sensors,” in *MOBICOM*. Cologne, Germany: ACM, Aug. 2005.

[27] A. Chen, S. Kumar, and T.-H. Lai, “Designing Localized Algorithms for Barrier Coverage,” in *MOBICOM*. ACM, Sep. 2007.

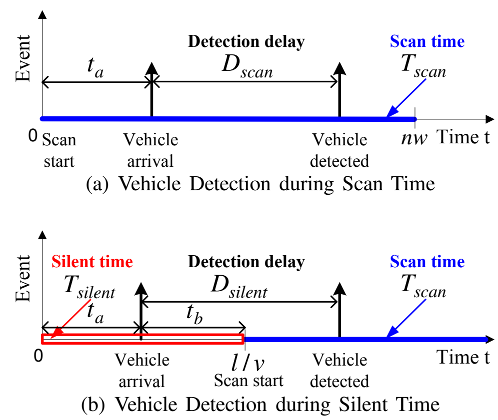
[28] P. Balister, B. Bollobás, A. Sarkar, and S. Kumar, “Reliable Density Estimates for Achieving Coverage and Connectivity in Thin Strips of Finite Length ,” in *MOBICOM*. ACM, Sep. 2007.

[29] C. Gui and P. Mohapatra, “Virtual Patrol: A New Power Conservation Design for Surveillance Using Sensor Networks,” in *IPSN*. IEEE, Apr. 2005.

Appendix

A. Average Detection Time in Virtual Scanning

In this section, we derive the Average Detection Time (ADT) for virtual scanning in a road segment. At first, for clarity, we assume vehicle speed is constant, the same as with the maximum speed v. Later, we relax this assumption; that is, vehicle speed is bounded variable speed.



(a) Vehicle Detection during Scan Time

(b) Vehicle Detection during Silent Time

Fig. 18. Vehicle Detection Cases in Virtual Scanning

Enter during Scan Time: Figure 18(a) shows a vehicle enters during the scan time T_{scan}. Since each node covers road segment of length l/n, the virtual scan wave moves along the road segment with the speed $v_{scan} = l/(nw)$. The relative speed between the scan wave and the vehicle is $l/(nw) + v$. Suppose a vehicle enters at t_a *after* the start of scan, the scan wave has already traveled $lt_a/(nw)$. Therefore it takes $(l - \frac{lt_a}{nw})/(\frac{l}{nw} + v)$ seconds before the scan wave reaches the vehicle, which is the detection delay D_{scan}. Integrated t_a over the interval $[0, nw]$, expected detection delay (denoted as $E[D_{scan}]$) during scan time is:

$$\begin{aligned} E[D_{scan}] &= \int_0^{nw} \frac{nwl - lt_a}{nwv + l}\frac{1}{nw}dt_a \\ &= \frac{nwl}{2(nwv+l)}. \end{aligned} \tag{5}$$

Enter during Silent Time: Figure 18(b) shows a vehicle enters during the silent time T_{silent}. Suppose a vehicle enters at t_a *after* the start of silent time. As shown in Figure 18(b), since it enters at t_b *before* the start of scan, the vehicle has already traveled $t_b v$. Therefore it takes $(l - t_b v)/(\frac{l}{nw} + v)$ seconds before the scan wave reaches the vehicle. For the detection delay, we also need to count the vehicle movement time t_b along with the previous detection delay after the start of the scan. Note that $t_b = l/v - t_a$. Thus the detection delay becomes $D_{silent} = \frac{l}{v} - t_a + (l - (\frac{l}{v} - t_a)v)/(\frac{l}{nw} + v)$. Integral t_a over the interval $[0, l/v]$, expected detection delay (denoted as

$E[D_{silent}]$) during the silent time is:

$$E[D_{silent}] = \int_0^{l/v} \frac{nwl - lt_a + l^2/v}{nwv+l} \frac{v}{l} dt_a = \frac{2nwl + l^2/v}{2(nwv+l)}. \quad (6)$$

Combined both scenarios, we can compute the expected ADT for the virtual scanning as follows:

$$E[D] = \frac{nw}{nw+l/v} E[D_{scan}] + \frac{l/v}{nw+l/v} E[D_{silent}] = \frac{l}{2v}. \quad (7)$$

ADT Computation for Bounded Variable Vehicle Speed: Now we relax the assumption that vehicle speed is constant, the same as with the maximum speed v_{max}. We assume that vehicle speed is bounded variable speed $v = [v_{min}, v_{max}]$ for $0 < v_{min} < v_{max}$. Since this relaxation causes the silent time to be changed as $T_{silent} = l/v_{max}$, the expected detection delay during the silent time becomes as follows:

$$E[D_{silent}] = \int_0^{l/v_{max}} \frac{nwl - lt_a + l^2/v_{max}}{nwv+l} \frac{v_{max}}{l} dt_a = \frac{2nwl + l^2/v_{max}}{2(nwv+l)}. \quad (8)$$

Since there exists no change in the detection delay during the scan time, the combined expected ADT is:

$$E[D] = \frac{nw}{nw+l/v_{max}} E[D_{scan}] + \frac{l/v_{max}}{nw+l/v_{max}} E[D_{silent}] = \frac{l(nwv_{max}+l)}{2v_{max}(nwv+l)}. \quad (9)$$

Clearly, Eq. 9 becomes the same one as with Eq. 7 for $v = v_{max}$.

Now we can compute the average detection time for bounded variable vehicle speed. Suppose that the vehicle speed is uniformly distributed in the range of $v = [v_{min}, v_{max}]$. We can compute the expected ADT for this setting as follows:

$$E[D] = \int_{v_{min}}^{v_{max}} \frac{l(nwv_{max}+l)}{2v_{max}(nwv+l)} \frac{1}{v_{max}-v_{min}} dv = \frac{l(nwv_{max}+l)}{2nwv_{max}(v_{max}-v_{min})} \log \frac{nwv_{max}+l}{nwv_{min}+l}. \quad (10)$$

To see the trend of ADT according to the average of bounded variable speed, we let $v = [\mu_v - \sigma_v, \mu_v + \sigma_v]$ where $\mu_v = 40$MPH and $\sigma_v = \{0, 5, ..., 30\}$MPH.

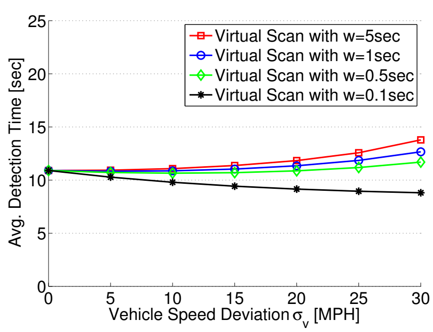


Fig. 19. The Impact of Vehicle Speed Deviation on Average Detection Time

Figure 19 illustrates the impact of vehicle speed deviation σ_v on the average detection time for four working times, $w = \{0.1, 0.5, 1, 5\}$sec. For the constant speed of $\sigma_v = 0$, all of the four cases have the average detection time of 10.9 sec. In the three cases except for $w = 0.1$ sec, the higher vehicle speed deviation, the longer average detection time; for $w = 0.1$ sec, the higher deviation leads to the shorter average detection time.