Dynamic risk-based scheduling and mobility of sensors for surveillance system

Congduc Pham
LIUPPA laboratory, Pau University, France
Congduc.Pham@univ-pau.fr

Moufida Maimour
CRAN laboratory, Nancy University, CNRS, France
Moufida.Maimour@cran.uhp-nancy.fr

Khadidja Fellah
Oran Es-Sénia University, Oran, Algeria
fellahkhadidja@yahoo.fr

Bouabdellah Kechar
Oran Es-Sénia University, Oran, Algeria
kechar.bouabdellah@univ-oran.dz

Hafid Haffaf
Oran Es-Sénia University, Oran, Algeria
haffaf_hafid@yahoo.fr

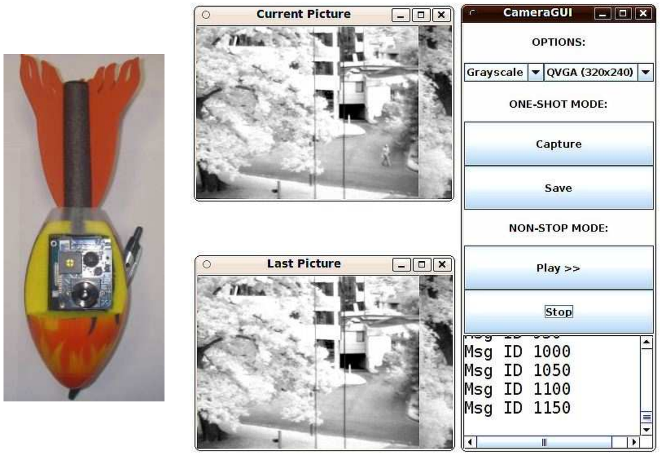
***Abstract*—Intrusion detection surveillance applications with wireless video sensor networks are those applications which require low energy consumption to increase network lifetime while at the time require a high level of quality of service. This paper show by simulation how a dynamic risk management scheme based on Bezier curves that takes into account the application's criticality can provide fast event detection for mission-critical surveillance applications while increasing at the same the network lifetime. We will also present some preliminary results on how to further increase network lifetime when some video sensor nodes have mobility feature. The motivation behind node's mobility is that video applications are characterized by their large amount of data so that a extra mobility cost can be justified as the video duration increases.**

***Index Terms*—Sensor networks, video surveillance, risk-based scheduling, reinforcement.**

I. Introduction

This article focuses on Wireless Video Sensor Networks (WVSN) mission-critical surveillance applications where sensors can be thrown in mass when needed for intrusion detection or disaster relief applications. The first concern in randomly deployed video sensors is that they will not land upside-down with the embedded camera turned towards the ground. This can actually be easily avoided for fixed nodes by fitting the video sensor in a rocket-shaped case which will always touch ground in the right way as illustrated by figure 1(left) (the figure shows an iMote2 with an IMB400 multimedia board [1]). Figure 1(right) shows a simple video surveillance application developed for the iMote2 with the IMB400 multimedia board that continuously takes pictures and displays both the current picture and the last picture.

The next thing to consider is that surveillance applications have very specific needs due to their inherently critical nature associated to security [2], [3], [4], [5], [6]. Early surveillance applications involving WSN have been applied to critical infrastructures such as production systems or oil/water pipeline systems [7], [8]. There have also been some propositions for intrusion detection applications but most of these early studies focused on coverage and energy optimizations without explicitly having the application's criticality in the control loop which is the main concern in our work. When adding the criticality concerns, one must introduce accountability aspects or more generally the so-called Quality of Service requirements to take into account the application's criticality so that an appropriate level of service can be defined. For instance, with video sensors, the higher the capture rate, the better relevant events could be detected and identified. However, even in the case of very mission-critical applications, it is not realistic to consider that video nodes should always capture at their maximum rate when in active mode.



```
Msg ID 1000
Msg ID 1050
Msg ID 1100
Msg ID 1150
```

Fig. 1. A proof-of-concept of a video sensor in a beach rocket toy (left). A simple video surveillance application with the iMote2 and IMB400 multimedia board (right).

Provided that the sensor node density is sufficiently high, randomly deployed sensor nodes can be redundant (nodes that monitor the same region) leading to overlaps among the monitored areas. Therefore, a common approach is to define a subset of the deployed nodes to be active while the other nodes can go to sleep. One obvious way of saving energy is to say that nodes that can be put in sleep mode are typically those whose sensing area are covered by others. The notion of cover set has therefore been introduced to define the redundancy level of a sensor [9]. However, in mission-critical applications where some sentry nodes are needed to increase responsiveness, nodes that possess a high redundancy

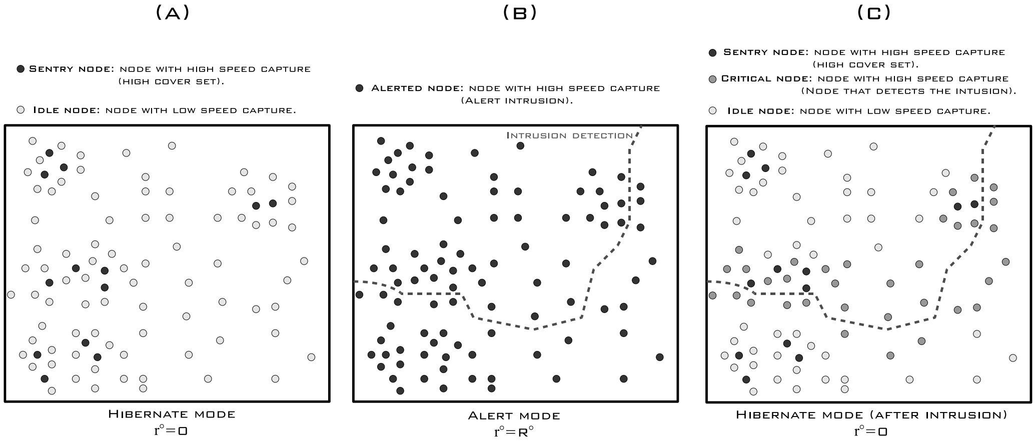

(A)
(B)
(C)
● Sentry node: node with high speed capture (high cover set).
○ Idle node: node with low speed capture.
● Alerted node: node with high speed capture (Alert intrusion).
● Sentry node: node with high speed capture (high cover set).
● Critical node: node with high speed capture (Node that detects the intusion).
○ Idle node: node with low speed capture.
Intrusion detection
Hibernate mode
r°=0
Alert mode
r°=R°
Hibernate mode (after intrusion)
r°=0

Fig. 2. Evolution of the video network nodes

level could rather be more active than other nodes with less redundancy level. In [10] the idea we developed is that when a node has several covers, it can increase its frame capture rate because if it runs out of energy it can be replaced by one of its cover sets. Then, depending on the application's criticality, the frame capture rate of those nodes with large number of cover sets can vary. It has been shown that this scheduling method outperforms a statically assigned frame capture rate approach.

Based on the criticality model of [10], this article focuses on the risk-based scheduling method for providing fast event detection. One new contribution is to investigate whether dynamic risk management with the risk level changing during the network lifetime can preserve network lifetime. Then we will present some preliminary results on how to further increase network lifetime when some video sensor nodes have mobility feature (small robots for instance). The motivation behind node's mobility is that video applications are characterized by their large amount of data so that a extra mobility cost can be justified as the video duration increases. The mobility optimization is mainly done at the initialization phase but can also be done when major topology changes occur to preserve connectivity and coverage. The main target in this paper are intrusion detection systems but the methodology can be extended to other surveillance applications such as environmental. The paper is then organized as follows: Section II presents the surveillance scenario and quickly reviews our coverage model for building sensor's cover sets. In section III we briefly present the dynamic risk management model. Section IV introduces the mobility possibilities. Section V then present the simulation results for both fast event detection and increased network lifetime in case of mobility. We conclude in section VI.

II. Intrusion detection with video sensor nodes

A. Intrusion detection scenario

One way to see the scheduling problem in critical surveillance applications is from the risk perspective: different parts of the area of interest may have different risk levels, noted r^0, according to the pattern of observed events such as the number of detected intrusions. In [11], the authors introduce so-called differentiated services by dynamically modifying the time duration for a node to work during each round. As we directly linked the application's criticality to the frame capture rate of a video sensor node, we want to impact on quality (number of frames) rather than on whole coverage as in [11]. Moreover, figure 2 shows the surveillance scenario we want to address in this paper where most of sensor nodes are in a so-called *hibernate* mode in the absence of intrusions: the risk level should be close to 0 and the sensor nodes should decrease their capture rate. However, it is also highly desirable that some sensor nodes still keep a relatively high capture rate to act as sentry nodes in the surveillance system (figure 2a). These nodes will be able to quickly detect intrusions and to alert, on intrusions, all active nodes so that they increase their risk level r^0 to a maximum value R^0, therefore moving to an *alerted* mode (figure 2b). R^0 can depend on the application's requirements in term of criticality, which in turn may depend on the environment the sensor network is intended to work in, and can be set in sensor nodes prior to deployment. In this scenario, after some time, an alerted node which does not detect more intrusions, should slowly go back to *hibernate* mode again (figure 2c). In this figure, we can also see that an alerted sensor node which does detect an intrusion (all sensor nodes close to the intruder's trajectory – dash line – in figure 2c) should stay with r^0 close to the maximum value.

B. Video sensor nodes and cover sets

The first step for defining coverage capabilities, is to define the video sensing model. We consider a commonly used 2-D model of a video sensor node where the FoV is defined as a triangle (*pbc*) denoted by a 4-tuple $v(P, d, \overrightarrow{V}, \alpha)$. Here P is the position of v, d is the distance pv (depth of view, DoV), $\overrightarrow{V}$ is the vector representing the line of sight of the camera's FoV which determines the sensing direction, and 2α is the angle of view (AoV). The left side of figure 3 illustrates the FoV of a video sensor node in this model. The AoV is

60^o and distance bc is the linear FoV. Some wireless sensor platforms can therefore have a video camera board. This is case for the iMote2 from Crossbow [1] where the IMB400 multimedia board's camera has an AoV of about 20^o. Figure 1(right) also showed a picture taken with this board.

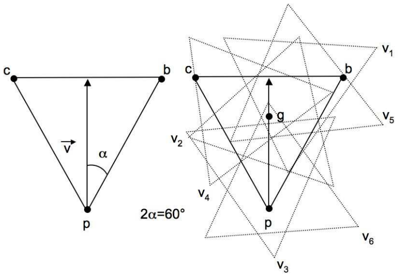


Fig. 3. Video sensing and coverage model

Random deployment of sensor leads to a high level of redundancy. We define a cover set $Co_i(v)$ of a video node v as a subset of video nodes such that: $\bigcup_{v' \in Co_i(v)} (v'$'s FoV area) covers v's FoV area. $Co(v)$ is then the set of all the cover sets of node v. Determining whether a sensor's FoV is completely covered or not by a subset of neighbor sensors is a time consuming task which is usually too resource-consuming for autonomous sensors. The basic approach presented in [12] is to use significant points of a sensor's FoV to quickly determine cover sets that may not completely cover sensor v's FoV but a high percentage of it. In the literature, most of existing omni-directional sensing coverage works try to construct disjoint sets of active nodes [13], [14], [15], [16]. In our case, we have the possibility that two or more covers have some video nodes in common. Hence, selecting one cover also reduces the life time of the sensor it has in common with another cover. In figure 3(right) we have $Co(v) = \{\{v\}, \{v_1, v_4, v_6\}, \{v_4, v_5, v_6\}\}$. [17] showed that this method has very good accuracy in terms of percentage of coverage and extensions have been proposed to handle heterogeneous AoV and very small AoV.

III. Risk-based scheduling of randomly deployed nodes with cover sets

A. Risk-based scheduling with Bezier curves

Once every sensor has broadcasted its position P and line of sight $\overrightarrow{V}$ to its neighbors and have then constructed all possible covers ($Co(v)$) that satisfy its local coverage objective (e.g. covering its FoV area), the scheduling of nodes can begin. The cover sets are sorted by increasing cardinality order. If needed, sensors could also estimate the percentage of coverage of each cover set by using a random sampling.

At startup, every node is active and waits to receive status packets from its neighbors. When a video node v receives the status of a neighbor v_x it adds v_x to the set of active neighbors and tests whether there is one cover set $Co_i(v)$ in $Co(v)$ that is included in the set of active neighbors. Every node orders their cover sets according to their cardinality, and gives priority to the covers with minimum cardinality. If a $Co_i(v)$ is found, v goes in sleep mode after sending its decision to its neighbors. In the case where no $Co_i(v)$ is satisfied, node v decides to remain in active mode and diffuses its decision.

As said previously, the frame capture rate is an important parameter that defines the surveillance quality. In [10], we proposed to link a sensor's frame capture rate to the size of its cover set. In our approach we define two classes of application: high and low risk applications. This risk level can oscillate from a concave to a convex shape as illustrated in Figure 4 with the following interesting properties:

- **Class 1 ”low risk”**, does not need high frame capture rate. This characteristic can be represented by a concave curve (figure 4 box A), most projections of x values are gathered close to 0.
- **Class 2 ”high risk”**, needs high frame capture rate. This characteristic can be represented by a convex curve (figure 4 box B), most projections of x values are gathered close to the max frame capture rate.

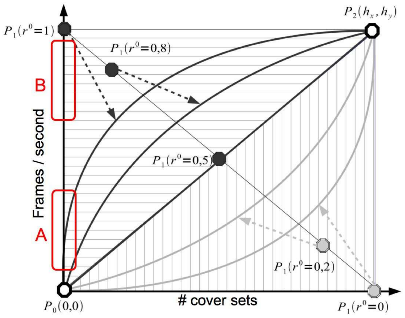


Fig. 4. The Behavior curve functions

We proposed in [10] to use a Bezier curve to model the 2 application classes. The advantage of using Bezier curves is that with only three points we can easily define a ready-to-use convex (high risk) or concave (low risk) curve: P_0, P_1, and P_2. $P_0(0, 0)$ is the origin point, $P_1(b_x, b_y)$ is the behavior point and $P_2(h_x, h_y)$ is the threshold point where h_x is the highest cover cardinality and h_y is the maximum frame capture rate determined by the sensor node hardware capabilities.

As also illustrated in Figure 4, by moving the behavior point P_1 inside the rectangle defined by P_0 and P_2, we are able to adjust the curvature of the Bezier curve, therefore adjusting the risk level r^0 introduced in the introduction of this paper. According to the position of point P_1 the Bezier curve will morph between a convex and a concave form. Interested readers can refer to [10] for more details on the modified Bezier curves definitions. Table I shows the corresponding capture rate for some relevant values of r^0. The cover set cardinality $|Co(v)| \in [1, 12]$ and the maximum frame capture rate is set to 3fps.

r^0	1	2	3	4	5	6	7	8	9	10	11	12
.1	.03	.08	.14	.22	.32	.45	.61	.81	1.1	1.4	1.9	3
.3	.11	.24	.38	.54	.72	.92	1.1	1.4	1.7	2.1	2.5	3
.6	.36	.69	1.0	1.3	1.5	1.8	2.0	2.2	2.4	2.6	2.8	3
.9	1.1	1.6	1.9	2.2	2.4	2.5	2.7	2.8	2.9	2.9	3	3

TABLE I
CAPTURE RATE IN FPS WHEN P2 IS AT (12,3).

B. Dynamic risk-based scheduling

Given the model described above, it is quite easy to vary the risk level during the network lifetime. The purpose is to only set the surveillance network in an alerted mode (high risk value) when needed, i.e. on intrusions. For instance, all nodes could start with a risk level $r^0 = 0.1$ and when a sensor node detects an intrusion, it would send an alert message to its neighbors and would increase its own risk level to $r^0 = 0.9$. Alerted nodes will then also increase their risk level to $r^0 = 0.9$. Alerted nodes will run at a high risk level for an alerted period before going back to $r^0 = 0.1$.

IV. INTRODUCING SENSOR MOBILITY

When some areas of the field are not or become uncovered, the mission of the entire network may be affected especially when the uncovered area is security critical. Connectivity, for its part, allows the different sensors to be able to reach each other as well as the sink (central controller or a gateway). Lack of connectivity could create unconnected sets in the network leading to some sensors to be unable to reach the sink.

Due to connectivity and coverage issues, nodes have to be placed carefully when deployed in the network field according to the target application. Good coverage and strong connectivity can be achieved through careful planning of node densities and fields of view so the network topology can be defined before startup [18][19]. However, this is impossible to achieve in randomly deployed networks. Moreover, a sensor network is dynamic by nature since sensors stop working when they exhaust their on-board energy supply. In a dynamic, hostile or hard-to-access environment, there is a need to be able to dynamically redeploy the network such that the application's requirements in terms of coverage and connectivity continue to be met while saving energy. This is what we call On-demand repositioning. In [20] for instance, sensor's ability to move is used to distribute them as evenly as possible in the region so coverage is achieved within the shortest time duration and with minimal overhead. A survey on node placement in WSN can be found in [21].

In this paper, we explore the possibility of having locomotion capabilities at some sensors so they are able to move [**?**]. The aim of this work is to save the overall communication energy in a video session by allowing mobile nodes to move. Even if mobility cost may be higher than communication, moves can be justified by preserving coverage and connectivity in the network. Moreover, moves are generally performed only once, at the beginning of a session, so video applications characterized by their large amount of data can have a small mobility cost as the video duration increases. Our approach is based on linear programming where we extended the work of [22] so both coverage and connectivity are considered. Additionally, our formulation fits the case of heterogeneous networks where video and scalar nodes coexist. Nodes may have different types of energy supplies (traditional batteries, solar or wind energy, etc.). Energy levels at nodes can be considered in the model so the network lifetime is increased.

A. Network Model

During the position broadcast phase described previously, support for mobility optimization can be provided by extending the neighbors broadcast to allow the sink(s) to get the position of all mobile nodes in the field which will be represented by a two-dimensional grid ($N_1 \times N_2$). Therefore all sensor nodes positions are then assumed to be known and are given by a boolean matrix P:

$$p_{i,j} = \begin{cases} 1 & \text{if there is a sensor at position } (i,j) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $0 \leq i \leq N_1 - 1$ and $0 \leq j \leq N_2 - 1$. The network can be heterogeneous to contain video and scalar sensors with different energies and processing powers. $c_{i,j,i',j'}$ is the amount of energy needed to transmit a 1-bit message by a sensor located at (i,j) and to be received by the one located at (i',j') and can be estimated using [23]:

$$c_{i,j,i',j'} = \alpha_{i,j}(2 \times E_{elec} + \epsilon_{amp} \times d^2_{i,j,i',j'}) \quad (2)$$

where $d_{i,j,i',j'}$ is the distance between the two sensors located at (i,j) and (i',j') positions, E_{elec} is the dissipated energy by the radio to run the transmitter or the receiver circuitry and ϵ_{amp} is the required energy by the transmit amplifier. We introduced a parameter $\alpha_{i,j}$, $0 \leq \alpha_{i,j} \leq 1$, defined on a per sensor basis in order to individually consider the energy capacities of each sensor node. For instance, a mobile node with solar cells can be assigned an $\alpha_{i,j}$ close to 0 and a node with a low energy level at a given time (possibly with ubiquitous energy) can be assigned an $\alpha_{i,j}$ close to 1. Sensors in the network can have different energy capacities. They can operate on batteries or even use energy extracted from the environment, such as solar energy or vibrations. This does not mean that the energy could become infinite [24] since harvesting energy can not be possible all the time and could be insufficient to provide sensors mobility for instance.

In our network model, some nodes have locomotion capabilities so they are able to move. Their positions can be known thanks to the mobility matrix $B(N_1 \times N_2)$:

$$b_{i,j} = \begin{cases} 1 & \text{if the sensor at location } (i,j) \text{ is mobile} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

To move from point (i,j) to (i',j') in the sensor field, the required energy is noted $m_{i,j,i',j'}$ and assumed to drain much more energy compared to communication cost per bit for the same distances, that is,

$$\forall i,j,i',j' : m_{i,j,i',j'}/c_{i,j,i',j'} = \rho > 1$$

In order to cover a given region or to avoid obstacles, a video sensor with locomotion facility may move mainly as a response to a sink request. However, a video sensor is assumed to stay at its location for the whole session when it begins capturing/transmitting images. Since there is a big amount of data to be transmitted in a video session and assuming that the transportation path is provided from the network layer, a relatively long schedule of messages send/receive can be obtained. We note by L, the number of messages to be transmitted. S and R are the transmission and reception matrices respectively before move where $s_{i,j,l} = 1$ if node at position (i,j) (before moving) sends the l^{th} message to another node and $r_{i,j,l} = 1$ if node at position (i,j) (before moving) receives the l^{th} message from another node. Each sensor node has a radio communication range r_c which is fixed and can not be varied during the video session.

Finally, we assume that each sensor node is able to sense within a disk of constant radius r_s and introduce the notion of *coverage degree*. Noted d_c, it is the number of redundant sensors that cover a given area. For video sensors, we aim to obtain a *soft* video coverage as opposed to *hard* coverage. a video sensor is able to move when there is another node to replace it even if it is not a video sensor and can not insure the same service degree (rich video capture). Nevertheless, it can contribute in covering the sensor field by sensing other physical (scalar) phenomenon such as movement detection. In a hard video coverage however, a video sensor moves only if there is another video sensor that it is able to replace it in the coverage of a given zone.

Notations and different parameters and variables used in this paper are listed in tables II and III.

B. Problem Formulation

In this section, we present our formulation to the problem of minimizing energy through mobility while preserving connectivity and coverage in our relatively heterogeneous network as described in the previous section. The problem can be formulated as an integer linear program (ILP) as follows:

minimize

$$E = \sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}\sum_{i'=0}^{N_1-1}\sum_{j'=0}^{N_2-1} \delta_{i,j,i',j'} \times m_{i,j,i',j'} + \sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}\sum_{i'=0}^{N_1-1}\sum_{j'=0}^{N_2-1}\sum_{l=1}^{L} sr_{i,j,i',j',l} \times k \times c_{i,j,i',j'} \qquad (4)$$

subject to

$$\forall i \in 0..N_1-1, \forall j \in 0..N_2-1, \sum_{i'=0}^{N_1-1}\sum_{j'=0}^{N_2-1} \delta_{i,j,i',j'} = p_{i,j} \qquad (5)$$

$$\forall i' \in 0..N_1-1, \forall j' \in 0..N_2-1, \sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1} \delta_{i,j,i',j'} \leq 1 \qquad (6)$$

TABLE II
NOTATIONS: PARAMETERS

N	number of sensor nodes.
$N_1 \times N_2$	sensor field dimensions.
P	matrix position: $p_{i,j} = 1$ if there is a node at (i,j).
$d_{i,j,i',j'}$	the distance between sensors located at (i,j) and (i',j').
B	mobility matrix: $b_{i,j} = 1$ if node at (i,j) is able to move.
k	number of bits per message.
L	number of messages to send.
S	transmission matrix before move, $s_{i,j,l} = 1$ if node at (i,j) (before moving) sends the l^{th} message, $1 \leq l \leq L$.
R	reception matrix before move: $r_{i,j,l} = 1$ if node at (i,j) (before moving) receives the l^{th} message, $1 \leq l \leq L$.
$\alpha_{i,j}$	weight given to node located at (i,j).
ρ	ratio of mobility to communication per bit cost: $\rho > 1$
C	communication energy matrix: $c_{i,j,i',j'}$ is the required energy to send a 1-bit message by a sensor located at (i,j) and to be received by the another one located at (i',j').
M	mobility energy matrix: $m_{i,j,i',j'}$ is the required energy to move from point (i,j) to (i',j').
r_c	communication radio range of the different sensors.
r_s	sensing (coverage) radius of each sensor.
d_c	required degree of coverage.

TABLE III
NOTATIONS: VARIABLES

$\dot{S}$	sending matrix after move: $\dot{s}_{i,j,l} = 1$ if node at (i,j) (after a move) sends the l^{th} message to any other node, $(1 \leq l \leq L)$.
$\dot{R}$	reception matrix after move: $\dot{r}p_{i,j,l} = 1$ if node at (i,j) (after a move) receives the l^{th} message from any other node, $(1 \leq l \leq L)$.
Δ	movement matrix: $\delta_{i,j,i',j'} = 1$ if node at (i,j) moves to optimal location (i',j').
SR	send/receive matrix after move: $sr_{i,j,i',j',l} = 1$ if (after move) node (i,j) takes part in the communication of message number l and sends it to a node located at (i',j'), $1 \leq l \leq L$.

$$\forall i' \in 0..N_1-1, \forall j' \in 0..N_2-1, \forall l \in 1..L,$$
$$\dot{r}_{i',j',l} = \sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1} \delta_{i,j,i',j'} \times r_{i,j,l} \qquad (7)$$

$$\forall i' \in 0..N_1-1, \forall j' \in 0..N_2-1, \forall l \in 1..L,$$
$$\dot{s}_{i',j',l} = \sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1} \delta_{i,j,i',j'} \times s_{i,j,l} \qquad (8)$$

$$\forall i \in 0..N_1-1, \forall j \in 0..N_2-1,$$
$$(p_{i,j} = 1) \wedge (b_{i,j} = 0) \Rightarrow \delta_{i,j,i,j} = 1 \qquad (9)$$

$$\forall i \in 0..N_1-1, \forall j \in 0..N_2-1, \forall l \in 1..L,$$
$$\sum_{i'=0}^{N_1-1}\sum_{j'=0}^{N_2-1} sr_{i,j,i',j',l} = \dot{s}_{i,j,l} \text{ with } d_{i,j,i',j'} \le r_c \tag{10}$$

$$\forall i \in 0..N_1-1, \forall j \in 0..N_2-1, \forall l \in 1..L,$$
$$\sum_{i'=0}^{N_1-1}\sum_{j'=0}^{N_2-1} sr_{i',j',i,j,l} = \dot{r}_{i,j,l} \text{ with } d_{i,j,i',j'} \le r_c \tag{11}$$

$$\forall i \in 0..N_1-1, \forall j \in 0..N_2-1,$$
$$\sum_{i''=0}^{N_1-1}\sum_{j''=0}^{N_2-1}\sum_{i'=0}^{N_1-1}\sum_{j'=0}^{N_2-1} \delta_{i'',j'',i',j'} \ge d_c \tag{12}$$

with $(i' \ge i - r_s) \wedge (i' \le i + r_s) \wedge (j' \ge j - r_s) \wedge (j' \le j + r_s) \wedge ((i \ne x) \vee (j \ne y))$ where (x, y) is the sink coordinates.

where E is the overall consumed energy including both communication and movement cost and k is the number of bits per transmitted packet. The different joined constraints are explained below:

(5): a node can move to any non-occupied place and a move can only take place from an occupied position in the network.

(6): any move to a non-occupied position is performed by only one node; otherwise this latter stays in its initial position.

(7) and (8) give expressions of $\dot{S}$ and $\dot{R}$, the emission and reception matrices respectively after move.

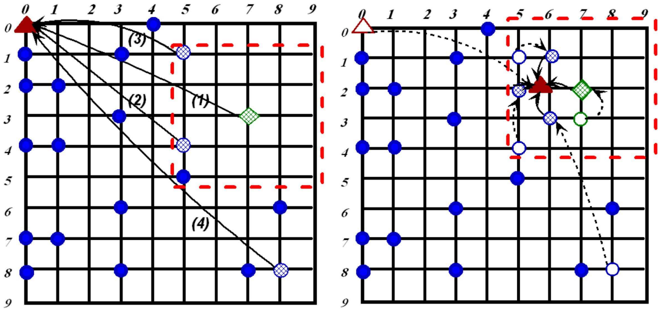
(9): a non mobile node located at (i, j) (i.e. $b_{i,j} = 0$) stays at its initial position.

(10) and (11) are the connectivity constraints. A message m is sent by one node and received by only one node (unicast communication). Moreover two nodes can not communicate unless they are in each other radio range. The distance between the two nodes (after move) is less or equal to the communication radio range [22].

(12) is the coverage constraint. Each position in the field is covered by at least d_c nodes to satisfy the required coverage degree. A node moves to position (i', j') from another one (i'', j'') or it stays at its initial position i.e. $i' = i''$ and $j' = j''$. Position (i, j) must be in the zone covered by the sensor located at (i', j').

Illustrative Example: we consider a field 10×10 where 20 sensor nodes are deployed as depicted by Figure 5(a) with 4 sources (at $(3, 7)$, $(4, 5)$, $(1, 5)$ and $(8, 8)$ willing to transmit one message each to the sink. Taking $r_s = 2$, each sensor node covers in addition to its own position, the 24 neighboring ones: the node located at $(3, 7)$ covers the square area within the dotted boundary as shown in Figure 5. In this sensor field, positions $(0, 8)$ and $(0, 9)$ are not covered. We assume that the communication radio range $r_s = 4$ and that communications are only possible in single hop (there is no underlying routing protocol). All sources can not reach the sink in one hop.

After applying our optimization program, all source nodes as well as the sink move as shown by dashed arrows in Figure 5(b). In this way, the required connectivity is satisfied so all the sources can achieve the sink in one hop. Additionally, the node located at $(3, 7)$ moves to position $(2, 7)$ so the problem of coverage is solved (points $(0, 8)$ and $(0, 9)$ become covered). The consumed energy is also reduced (for one message with 1024 bits, $401mJ$ is consumed instead of $403mJ$).



(a)

(b)

Fig. 5. Illustrative Example: coverage and connectivity constraints

V. Simulation results

A. Risk-based scheduling

For all the risk-based scheduling simulations we used the OMNET++ discrete event simulator (http://www.omnetpp.org). 150 sensor nodes are randomly deployed in a $75m * 75m$ area. Sensors have an 36^o AoV, a DoV of 25m and communication range of 30m. Each sensor node captures with a given number of frames per second (between 0.01fps and 3fps) according to the model defined in figure 4. Nodes with 12 or more cover sets will capture at the maximum speed. Simulation ends when there are no active nodes anymore.

1) Static risk-based scheduling: We ran simulations for 9 levels of risk, from $r^0 = 0.1$ to $r^0 = 0.9$. The corresponding capture rates are those shown in table I. With 150 nodes, the percentage of coverage of the initial area is very close to 100% of the initial area. Nodes with high capture rate will use more battery power until they run out of battery (initial battery level is 100 units, 1 captured image consumes 1 unit) but according to the scheduling model nodes with high capture rate are also those with large number of cover sets. Note that it is the number of valid cover sets that defines the capture rate and not the number of cover sets found at the beginning of the cover sets construction procedure. Figure 6 shows the mean stealth time (MST) when r^0 is varied from $r^0 = 0.1$ to $r^0 = 0.9$ by 0.1 increments. The stealth time is the time during which an intruder can travel in the field without being seen. The first intrusion starts at time 10s at a random position in the field. The scan line mobility model is then used with a constant velocity of 5m/s to make the intruder moving to the right part of the field. When the intruder is seen for the first time by a sensor, the stealth time is recorded and the MST computed. Then a new intrusion appears at another random position. This process is repeated until the simulation ends. In figure 6, the higher the risk level, the smaller the MST and the network lifetime.

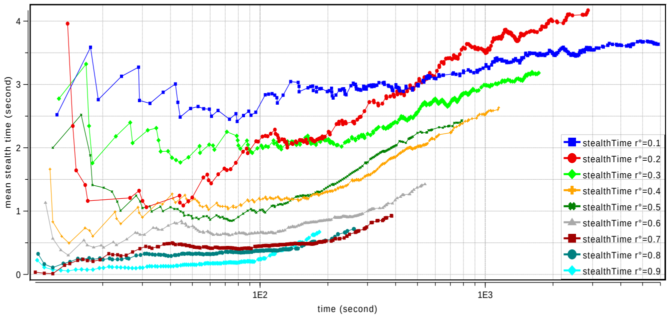


Fig. 6. Mean stealth time. Static scheduling: $r^0 = 0.1, .., 0.9$.

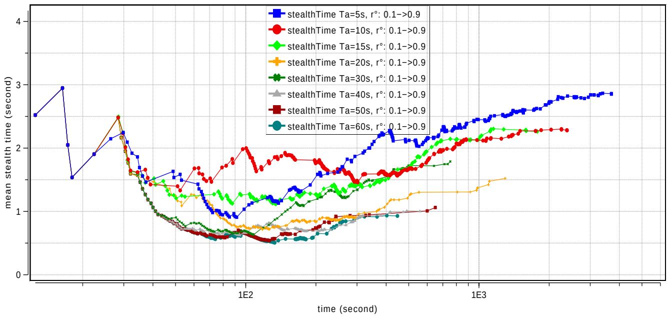


Fig. 7. Mean stealth time. Dynamic scheduling: r^0 increases from 0.1 to 0.9 on intrusion.

2) Dynamic risk-based scheduling: With the same network topology than the previous simulations, we set the initial risk level of all nodes to $r^0 = 0.1$. When a sensor node detects an intrusion, it sends an alert message to its neighbors and increases its risk level to $r^0 = 0.9$. Alerted nodes will then also increase their risk level to $r^0 = 0.9$. Both the node that detects the intrusion and the alerted nodes will run at a high risk level for an alerted period, noted T_a, before going back to $r^0 = 0.1$. Nodes may be alerted several times but an already alerted nodes will not increase its T_a value any further in this simple scenario. Figure 7 shows the MST when T_a is varied from 5s to 60s. When T_a is high, the stealth time is small but the network lifetime decreases dramatically. However, we can also see that this simple dynamic scenario already succeeds in reducing the MST while increasing the network lifetime when compared to the static scheduling that provides the same level of service. For instance the $T_a = 20s$ case that gives a MST very close to 1.5s (below 1s most of the time) lasts for 1300s while an equivalent level of service needs a risk level of $r^0 = 0.6$ which only lasts for 540s.

B. Sensor mobility

In order to get some insight into the benefit of mobility to save energy in a WVSN, our formulated problem was coded using AMPL (A Mathematical Programming Language) [25] and solved using the CPLEX solver [26] on NEOS server [27].

We consider the case of a grid of dimension 10×10 where 40 nodes among which a given ratio is assumed to be mobile, are randomly placed. The sink is located at position $(0, 0)$ and depending on the experiment, one to seven sources are randomly chosen in the field. Paths from each source to the sink are generated using MFR (Most Forward within Radius) [28]. Each source is assumed to capture and transmit a 10-second video sequence (*Hall Monitor* [29]). Data packets are assumed to have 1024 bits of payload. Information about paths, amount of data to be transmitted and the size of packets allow us to generate the corresponding communication schedule required as an input of our ILP. For the energy model, we put in equation (2), $E_{elec} = 50nJ/bit$ and $\epsilon_{amp} = 0.1nJ/bit/m^2$.

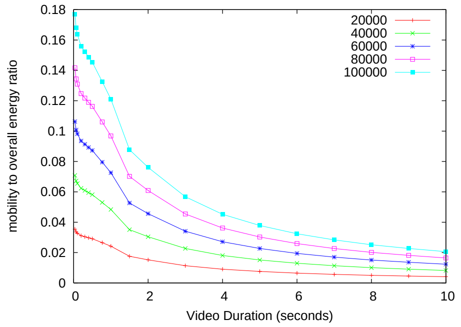


Fig. 8. Mobility to overall energy ratio for different values of ρ. One source, 20% of nodes are mobile

Figure 8 plots the mobility to the overall consumed energy ratio as a function of video duration for different values of ρ. In this scenario, 20% nodes have locomotion facilities and only one source is transmitting. The overall consumed energy includes energy required by nodes to move to their optimal positions and the consumed energy due to transmitting/receiving data packets. We can see that if we increase ρ (till 100,000) so the mobility cost is much higher than the communication one and even for a small video duration (0.1 second for instance), mobility cost is at most about 18% of the overall consumed energy. It is also to notice that the share of mobility in overall energy consumption decreases with session duration. In fact, the longer the video session, the larger the amount of data to deliver. As a result, the communication cost increases compared the mobility one where moves are performed only once at the beginning of a session.

In order to assess the gain obtained thanks to nodes mobility, we plot curves of Figure 9 showing the amount of energy (in Joules) saved when applying our optimization problem as a function of video duration for different densities of mobile nodes in the field. We can see that the amount of saved energy is higher for larger number of mobile nodes. Furthermore, when the video session duration increases, saved energy is also increased. This confirms results obtained and observed in Figure 8. The amount of energy saved allows for augmenting the lifetime of the entire network.

Finally, we varied the number of transmitting sources from 1 to 7 and reported the amount of saved energy for different video streaming durations ranging from 1 to 5 minutes. Figure 10 plots this saved energy and shows, once again, that when increasing the video duration, the saved energy increases.

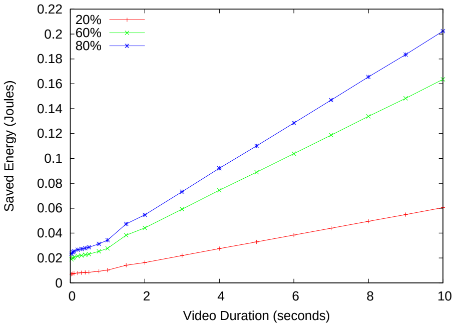


Fig. 9. Saved energy as a function of video duration for different densities. $\rho = 1000$

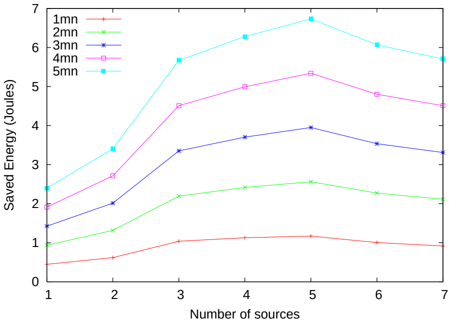


Fig. 10. Saved energy as a function of the number of sources. $\rho = 1000$

When augmenting the number of sources until 5 sources, we save more energy. However when the number of sources reaches 6, we get less energy saving. This is due to the fact that when increasing the number of sources, some nodes are likely to belong simultaneously to more than one path.

VI. Conclusions

This paper addresses video intrusion detection systems and is based upon a risk-based scheduling model that determines the video node's capture rate according to the node's number of cover sets and the risk level. We showed that using dynamic risk management can increase the network lifetime while maintaining a high level of service in the form of small stealth time. These results show that besides providing a model for translating a subjective risk level into a quantitative parameter, our approach for video sensor nodes also optimize the resource usage by dynamically adjusting the provided service level. Regarding mobility, we formulated the problem of optimal node placement in a WVSN so energy consumption is minimized under coverage and connectivity constraints using ILP. The results mainly showed that even if mobility cost is much higher than communication, the latter tends to be predominant in the overall consumed energy as the video session duration increases.

Acknowledgment

This work is partially supported by the FEDER POCTEFA EFA35/08 PIREGRID project, the Aquitaine-Aragon OMNI-DATA project and by the PHC Tassili project 09MDU784.

References

[1] Crossbow, "http://www.xbow.com," accessed 3/05/2010.

[2] T. He et al., "Energy-efficient surveillance system using wireless sensor networks," in *ACM MobiSys*, 2004.

[3] S. Oh, P. Chen, M. Manzo, and S. Sastry, "Instrumenting wireless sensor networks for real-time surveillance," in *Proc. of the International Conference on Robotics and Automation*, May 2006.

[4] O. Dousse, C. Tavoularis, and P. Thiran, "Delay of intrusion detection in wireless sensor networks," in *ACM MobiHoc*, 2006.

[5] Y. Zhu and L. M. Ni, "Probabilistic approach to provisioning guaranteed qos for distributed event detection," in *IEEE INFOCOM*, 2008.

[6] E. Freitas et al., "Evaluation of coordination strategies for heterogeneous sensor networks aiming at surveillance applications," in *IEEE Sensors*, 2009.

[7] L. N. I. Stoianov and S. Madden, "Pipenet: A wireless sensor network for pipeline monitoring." in *ACM IPSN*, 2007.

[8] S. C. M. Albano and R. D. Pietro, "A model with applications for data survivability in critical infrastructures," *Journal of Information Assurance and Security*, vol. 4, 2009.

[9] M. Cardei et al., "Energy-efficient target coverage in wireless sensor networks," *IEEE INFOCOM*, 2005.

[10] A. Makhoul, R. Saadi, and C. Pham, "Risk management in intrusion detection applications with wireless video sensor networks," in *IEEE WCNC*, 2010.

[11] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *ACM SenSys*, 2003.

[12] A. Makhoul and C. Pham, "Dynamic scheduling of cover-sets in randomly deployed wireless video sensor networks for surveillance applications," in *IFIP Wireless Days*, 2009.

[13] C. Wang et al., "Coverage based information retrieval for lifetime maximization in sensor networks," *In IEEE GlobeCom*, 2007.

[14] J. Bahi, A. Makhoul, and A. Mostefaoui, "Hilbert mobile beacon for localization and coverage in sensor networks," *Int. Jour. of Systems Science*, vol. 39, no. 11, pp. 1081–1094, 2008.

[15] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Localized sensor area coverage with low communication overhead," *IEEE Trans. on Mobile Computing*, vol. 7, no. 5, pp. 661–672, 2008.

[16] J. Bahi, A. Makhoul, and A. Mostefaoui, "Improving lifetime and coverage through a mobile beacon for high density sensor networks," *SENSORCOMM08*, pp. 335–341, august 2008.

[17] C. Pham and A. Makhoul, "Performance study of multiple cover-set strategies for mission-critical video surveillance with wireless video sensors," in *IEEE WiMOB*, 2010.

[18] M. Ishizuka and M. Aida, "Performance study of node placement in sensor networks," in *24th International Conference on Distributed Computing Systems Workshops - W7: EC (Icdcsw'04)*, 2004.

[19] A. Boukerche, X. Fei, and R. B. Araujo, "A coverage preserving and fault tolerant based scheme for irregular sensing range in wireless sensor networks," in *IEEE GLOBECOM*, 2006.

[20] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[21] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 6, no. 4, pp. 621–655, 2008.

[22] I. Kadayif, M. T. Kandemir, N. Vijaykrishnan, and M. J. Irwin, "An integer linear programming-based tool for wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 65, no. 3, pp. 247–260, 2005.

[23] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *HICSS*, 2000.

[24] D. E. Tiliute, "Battery management in wireless sensor networks," *Electronics and Electrical Engineering*, vol. 4, no. 76, pp. 9–12, 2007.

[25] K. B. W. Fourer R, Gay D M, *AMPL: A Modelling language for Mathematical Programming*. Duxury, 2002.

[26] "Cplex," www.cplex.com [accessed, August 2nd, 2011].

[27] "Neos," http://neos.mcs.anl.gov/neos/solvers/milp:scip/AMPL.html [accessed, August 2nd, 2011].

[28] X. Lin and I. Stojmenovic, "Location-based localized alternate, disjoint and multi-path routing algorithms for wireless networks," *J. Parallel Distrib. Comput.*, vol. 63, no. 1, pp. 22–32, 2003.

[29] "Evalvid," http://www.tkn.tu-berlin.de/research/evalvid/cif.html [accessed, August 2nd, 2011].