Risk Management in Intrusion Detection Applications with Wireless Video Sensor Networks

Abdallah Makhoul
University of Franche-Comté
LIFC Laboratory
Email: abdallah.makhoul@iut-bm.univ-fcomte.fr

Rachid Saadi
INRIA
Rocquencourt
Email: rachid.saadi@inria.fr

Congduc Pham
University of Pau
LIUPPA Laboratory
Email: congduc.pham@univ-pau.fr

***Abstract*—In randomly deployed wireless video sensor networks for surveillance applications, the scheduling of sensor nodes can be seen from the risk perspective: different parts of the area of interest may have different risk levels according to the pattern of observed events such as the number of detected intrusions. In this paper, we propose a multiple-level activity model that uses behavior functions to define application classes and allows for adaptive scheduling based on the application criticality and on the availability of multiple cover sets per video sensor node. The paper then describes how an adaptive scheduling model can be defined in order to dynamically schedule video nodes by varying the capture rate according to nodes' environment. Simulation results are presented to validate the performance of the proposed approach.**

I. Introduction

Instead of using traditional vision systems built essentially from fixed video cameras it is possible to deploy autonomous and small wireless video sensor nodes (WVSN) [1] to add a much higher level of flexibility, therefore extending the range of surveillance applications that could be considered. More interestingly, this scenario can support dynamic deployment scenario even in so-called object and obstacle-rich environments or hard-to-access areas. Such wireless video sensor nodes can in addition be thrown in mass to constitute a large scale surveillance infrastructure. In these scenarios, hundreds or thousands of video nodes of low capacity (resolution, processing and storage) of a same or similar type can be deployed in an area of interest.

Surveillance applications have very specific needs due to their inherently critical nature associated to security [2], [3], [4]. For video-based surveillance systems, the quality of the captured images or the capture rate must be in adequacy with the application's objectives: it is unnecessary to send an image with a high bandwidth if it is not sufficiently enlightened to allow detection and/or identification of intruders. We can add to these problems the fact that such networks for surveillance strongly lose its interest if the relevant scenes could not be received leading to a bad interpretation of an event which can have disastrous effects.

One other issue of prime importance is related to energy considerations since the scarcity of energy does have a direct impact on coverage as it is not possible to have all the sensor nodes in activity at the same time. In high density randomly deployed sensor networks, 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 sleep. Many contributions have been made in the last few years and the authors in [5], [6], [7], to name a few, have proposed interesting energy-efficient approaches that aim at providing the highest detection quality. However, it is also desirable to have differentiated surveillance services for various target areas with different degrees of security requirements [8] or to be able to probabilistically support flexible QoS [4] without over-provisioning resources. In surveillance applications such as an intrusion detection system having multiple levels of activity is essential because such systems must operate on a long term basis as no one knows when such intrusions could occur. In addition some surveillance applications may focus on barrier coverage rather than blanket coverage. In this case, sensor nodes at the border of the area of interest should be more active while interior nodes can decrease their activity.

In this paper that specifically addresses directional video sensor nodes, we take into account the application criticality and propose a model that deterministically defines multiple levels of activity corresponding to how many samples (images) are captured per unit of time. In order to take into account the energy considerations, the capture rate of each node further depends on the redundancy level found for this node. For this purpose, we use a multiple cover set approach to manage sensor's Field of View (FoV) redundancies [9]. The approach is based on a distributed algorithm that helps each node to organize its neighbors into non-disjoint subsets, each of which being a cover set that overlaps its FoV. Then, based on neighbors activity, a node decides to be active or in sleep mode, without compromising the coverage of its own FoV. For video sensor networks efficient scheduling and the ability to provide multiple levels of activity is even more important than in traditional sensing systems (e.g. temperature, pressure,...) as capturing and transmitting images are much more energy-consuming.

The paper is then organized as follows. Section II presents the application scenario we wan to address in this paper with the proposed dynamic risk model. Section III presents the adaptive scheduling model and the proposed dynamic risk model that uses behavior functions modeled by modified

Bezier curves for providing the necessary flexibility to the end applications. Simulation results are presented in section IV before the conclusions.

II. Randomly deployed video sensors for surveillance

Surveillance and monitoring applications are domains where wireless sensor networks can provide attractive features. In this type of application, wireless sensor nodes can run in two different modes: one mode we call infrastructure mode and the other we call the open mode. Each mode has an impact on the associated algorithms and the possible scenario. The infrastructure mode refers to a surveillance mode where the area under surveillance is known and in which one can identify accurately predefined area to monitor. It is typically the case when a building is under surveillance and where parking lots and entrances are typical areas that have been identified as important (there are strong semantic associated to each area, for instance, the entrance is usually the access/weak point of the building). In this mode, each sensor node knows what it is monitoring and it is much easier to handle coverage, data redundancy and node's scheduling.

In the open mode, there is no well-defined area and no strong semantic associated to a particular area : all the covered zone is potentially important and generally it is a much larger geographical area than in the infrastructure mode. Depending on the deployment scenario that has been discussed before, some applications are typical of the open mode : intrusion detection, observation, exploration. However, random deployment appears to be the privileged deployment method of the open mode and it is the type of deployment we are considering in this paper. When the sensor nodes are deployed to form a barrier to detect intrusion, the network is said to provide barrier coverage [10], [11]. Kumar et al. in [11] proposes an optimal sleep-wake up algorithm for barrier coverage that greatly improved the network lifetime while considering the case when sensor nodes have distinct lifetime. In the work presented in this paper, not only barrier coverage applications are considered but more generally intrusion detection in any point of the area of interest and provision for facilitating the tracking process. Also, we are considering video surveillance applications with video sensor nodes.

For wireless video sensor nodes the frame capture process and transmission to a sink is a huge energy-consuming task. On the other hand, the higher the capture rate is, the better relevant events could be detected and identified. Therefore if we consider a video surveillance application for intrusion detection, video nodes should capture at a high rate due to the criticality of this type of application. There are other types of non-critical surveillance or monitoring applications where the capture rate does not need to be set to the maximum rate. However, even in the case of an intrusion detection application, it is not realistic to consider that video nodes should always capture at their maximum rate when in active mode because network lifetime is almost as important as coverage in such applications. In general, it is desirable to be able to adjust the capture rate according to the application's requirements. In our approach we want to express the application criticality by a single r^0 variable which can take values between 0 and a defined 1 representing the low and the high criticality level respectively. Low level criticality indicates that the application does not require a high video frame capture rate while a high level criticality does. Then, according to the application's requirements, r^0 could be initialized accordingly into all sensors nodes prior to deployment.

Then one other way to see the scheduling problem in such critical surveillance applications is from the risk perspective: different parts of the area of interest may have different risk levels according to the pattern of observed events such as the number of detected intrusions. In [8], the authors introduce so-called differentiated services by dynamically modify the time duration for a node to work during each round. As we directly linked the application 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 [8].

Figure 1 shows the scenario of a random deployment of video sensor nodes for a video surveillance application that we address in this paper. We want that most of sensor nodes move to a so-called *hibernate* mode in the absence of intrusions: the risk level should decrease to $r^0 = 0$ 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 even when $r^0 = 0$ in order to act as sentry nodes in the surveillance system (figure 1a) to better detect intrusions and to alert, on intrusions, all active nodes to increase their risk level r^0 to the maximum value, therefore moving to an *alerted* mode (figure 1b). This maximum value is called the criticality threshold and is noted R^0. It can depend on the application requirements in term of criticality, which in turn may depend on the environment the sensor network is intended to work in, and 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 by decreasing its risk level r^0 to 0 in order to save energy, see figure 1c. 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 1c) stays at r^0 close to the maximum value.

In order to completely implement the dynamic risk level management scenario depicted previously by figure 1, it is necessary to define:

- how (fast) a sensor node decreases its risk level (r^0) to move to *hibernate* mode when there are no intrusions.
- how many nodes are set to *alerted* mode upon intrusion detection by a sensor node i: this can vary from all nodes at only 1-hop of sensor node i to all nodes at k-hops of sensor node i. If k is large, all sensor nodes could be set in *alerted* mode.

These issues are out of the scope of this paper and we leave them to future works. The rest of the paper will present

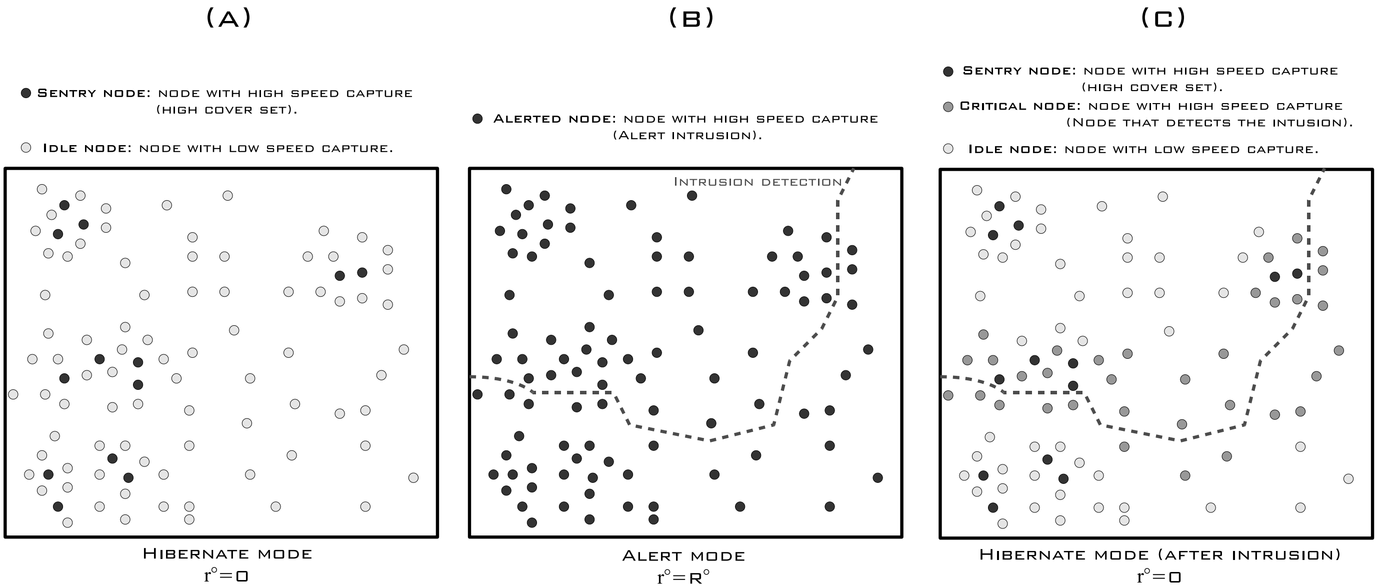

Fig. 1. Evolution of the video network nodes

our dynamic risk management model in order to support the scenario described above.

III. Dynamic risk level management model

A naïve approach would consist in fixing the frame capture rate of all video nodes to a given rate. As illustrated in figure 2, we show how the video nodes capture speed can be regulated proportionally to the critical level R^0. For instance, a high criticality level pushes video nodes to capture at near the maximum frame rate capability.

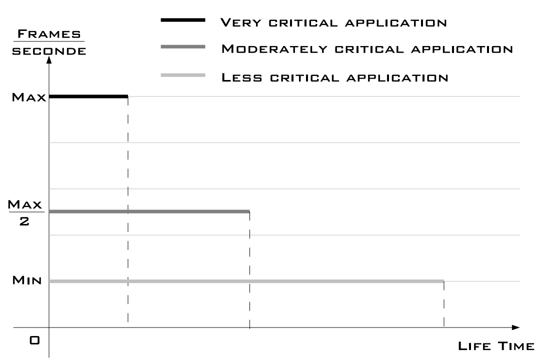

Fig. 2. Naïve approach.

However, this simple approach presents some drawbacks. In fact, (i) setting video nodes to work at full capacity provides very good capture quality but the network lifetime is very short, (ii) although setting the nodes at low capacity saves energy and extends the network lifetime, it provides poor surveillance quality, (iii) choosing a moderate frame capture rate could balance between capture quality and network lifetime but at the same time sensors cannot be fully exploited if it is necessary.

To fully exploit the video node capabilities we propose that a video node captures frames at a rate that is defined by the size of its cover set. The idea 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 covers. In addition, the frame capture rate variation should also depend on the criticality level of the application as discussed previously. In what follows we will define different application classes which will determine a node's frame capture rate.

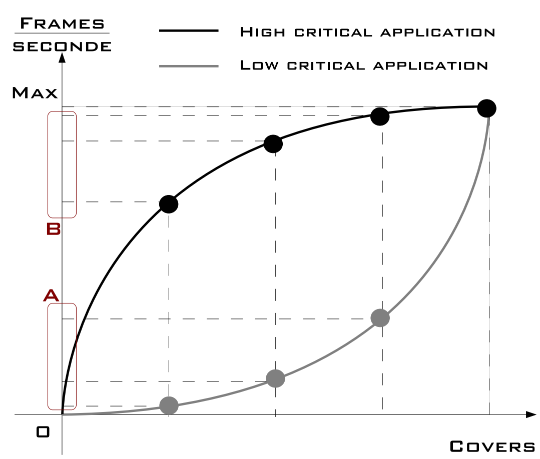

Fig. 3. Dynamic approach.

A. Application classes

We can classify applications into different categories based on their criticality level. In our approach we define two classes of applications: high and low criticality applications. This criticality level can oscillate from a concave to a convex shape as illustrated in Figure 3. The reason we chose concave and convex shapes will be described later on when we will define the 2 application classes. In figure 3 the values on the x and y axis are defined as follows:

- values on the x axis are positive integers representing the cardinality of the cover set $|Co|$. Integer values lie between 1 and max, where max is fixed according to the network topology. A video sensor node that is the only node capable of covering its FoV has a cover size of 1 on the x axis.
- the y axis gives the corresponding frame capture rate based on the cardinality of the cover set expressed on the x axis and the application criticality level (R^0).

We now present the contrast between applications that exhibit high and low criticality level:

- **Class 1 "low criticality"**, $0 \le R^0 < 0.5$: this class of applications does not need high frame capture rate. This characteristic can be represented by a concave curve. As illustrated in figure 3 (box A), most projections of x values are gathered close to zero (i.e. the majority of the sensors will preserve their energy by capturing slowly). However, the concave shape allows nodes with large number of cover sets to keep capturing at a high rate to become a sentry node in the network.
- **Class 2 "high criticality"**, $0.5 \le R^0 \le 1$: This class of applications needs high frame capture rate. This characteristic can be represented by a convex curve. As illustrated in figure 3 (box B), most projections of x values are gathered close to the max frame capture rate (i.e. the majority of nodes capture at a high rate). However, for those nodes with very few or no cover sets, the energy considerations take over and they remain at a relatively slow capture rate.

In the same way, the *alerted* and the *hibernate* modes are defined as follows with the risk level r^0 :

- **Hibernate mode**, $0 \le r^0 < \frac{R^0}{2}$: this mode decreases the frame capture speed of the entire network and is represented by a concave shape where most nodes capture very slowly except for the sentry nodes.
- **Alerted mode** $\frac{R^0}{2} \le r^0 \le R^0$: this mode increases the network frame capture speed and is represented by a convex ($R^0 > 0.5$) or a concave ($R^0 < 0.5$) curve close to the R^0 curve.

Note that there is no threshold that defines whether a node with large number of cover sets should act as sentry nodes or not. The only difference between active nodes relies on their capture frame rate. It is however possible to say that, by definition, a sentry node is a node that captures at a rate above 4 fps for instance.

B. The behavior function

Given the desired behavior described above, we want to define a mathematical function which will allow each node i to link its frame capture speed to the network deployment (i.e. cover set cardinality $|Co_i|$) and the observed events (i.e risk level r^0). We call this function BV (BehaVior) function which must be an increasing function regardless of the value of r^0: nodes with a large number of cover sets can afford to capture faster because they can be easily replaced. By contrast, a node with a small number of cover sets should preserves its energy because it can hardly be replaced. Furthermore, when the risk level increases the capture speed must also increase to take into account the higher criticality level even for nodes with small number of cover sets.

We propose to use a Bezier curve to model the BV function. The bezier curve is a parametric form to draw a smooth curve. It is fulfilled through some points $P_0, P_1 ... P_n$, starting at P_0 going towards $P_1 ... P_{n-1}$ and terminating at P_n. The advantage of using Bezier curves is that with only three points we can easily define a ready-to-use convex (high criticality) or concave (low criticality) curve: P_0, P_1, and P_2 (Quadratic Bezier curve) which are defined by:

$$B(t) = (1-t)^2 * P_0 + 2t(1-t) * P_1 + t^2 * P_2.$$

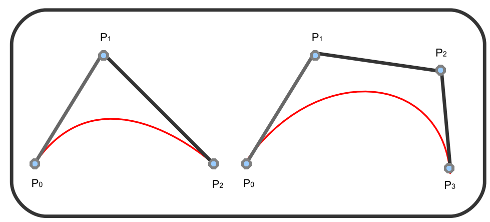

Fig. 4. The Bezier curve

$P_0(0,0)$ 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 illustrated in Figure 5, 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. The BV function describes the application criticality. It takes $|Co|$ as input on the x axis and returns the corresponding "frame capture rate" on the y axis. To apply the BV function with the Bezier curve, we modify this latter to obtain y as a function of x, instead of taking a temporal variable t as input to compute x and y. Based on the Bezier curve, the BV function curve can be drawn through the three points $P_0(0,0)$, $P_1(b_x, b_y)$ and $P_2(h_x, h_y)$ using the Bezier curve as follows:

$$\begin{array}{rcl} BV : [0, h_x] & \longrightarrow & [0, h_y] \\ X & \longrightarrow & Y \end{array}$$

$$BV_{P_1,P_2}(X) = \left\{ \begin{array}{ll} \frac{(h_y - 2b_y)}{4b_x^2} X^2 + \frac{b_y}{b_x} X & if\ (h_x - 2b_x = 0) \\ (h_y - 2b_y)(\propto (X))^2 + 2b_y \propto (X), & if\ (h_x - 2b_x \neq 0) \end{array} \right.$$

$$Where\ \propto (X) = \frac{-b_x + \sqrt{{b_x}^2 - 2b_x * X + h_x * X}}{h_x - 2b_x} \wedge \left\{ \begin{array}{l} 0 \le b_x \le h_x \\ 0 \le X \le h_x \\ h_x > 0 \end{array} \right.$$

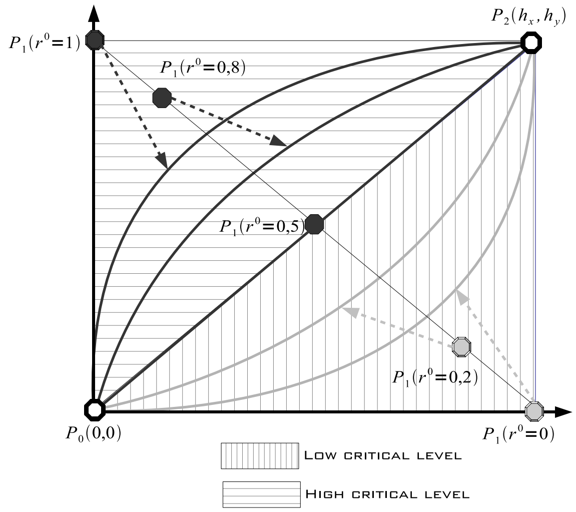

Fig. 5. The Behavior curve functions

C. The risk level r^0

As discussed above, the risk level r^0 of an node is given into the interval $[0, R^0]$. According to this level, we define the risk function called Rk which operates on the behavior point P_1 to control the BV function curvature.

According to the position of point P_1 the Bezier curve will morph between a convex and a concave form. As illustrated in figure 5 the first and the last points delimit the curve frame. This frame is a rectangle and is defined by the source point $P_0(0, 0)$ and the threshold point $P_2(h_x, h_y)$. The middle point $P_1(b_x, b_y)$ defines the risk level. We assume that this point can move through the second diagonal of the defined rectangle $b_x = \frac{-h_y}{h_x} * b_y + h_y$.

We define the Rk function as follows, such that varying r^0, the dynamic risk level, between 0 and R^0 gives updated positions for P_1:

$$\begin{array}{rcl} Rk : [0, R^0] & \longrightarrow & [0, h_x] * [0, h_y] \\ r^0 & \longrightarrow & (b_x, b_y) \end{array}$$

$$Rk(r^0) = \begin{cases} b_x = -h_x \times r^0 + h_x \\ b_y = h_y \times r^0 \end{cases}$$

Level r^0 is represented by the position of point P_1. If $r^0 = 0$ P_1 will have the coordinate $(h_x, 0)$. If $r^0 = 1$ P_1 will have the coordinate $(0, h_y)$. Table I shows the corresponding capture rate for some relevant value of r^0. The cover set cardinality $|Co(v)| \in [1, 6]$ and the maximum frame capture rate is 6 fps.

r^0 \ $\|Co(v)\|$	1	2	3	4	5	6
0.0	0.05	0.20	0.51	1.07	2.10	6.00
0.2	0.30	0.73	1.34	2.20	3.52	6.00
0.5	1.00	2.00	3.00	4.00	5.00	6.00
0.8	2.48	3.80	4.66	5.27	5.70	6.00
1.0	3.90	4.93	5.49	5.80	5.95	6.00

TABLE I
CAPTURE RATE IN FPS

IV. PREMLIMINARY SIMULATION RESULTS

To evaluate our approach we conducted a series of simulations based on the discrete event simulator OMNet++ [13]. For these set of experiments, we randomly deploy 150 sensor nodes in a $75m * 75m$ area. Nodes have equal communication and sensing ranges of $30m$ and $25m$ respectively, an offset angle α of $\pi/6$, a battery life of 100 units, random position P and random direction $\overrightarrow{V}$. A simulation starts by a neighborhood discovery then, round by round each node decides to be active or not. Each sensor node captures with a given number of frames per second (between 0 fps and 6 fps). In these simulations we observed the following initial statistics: the maximum number of cover sets observed is 10 (1 node), there is about 10% of nodes with no cover set, 40% of nodes have 6 or more cover sets. The battery capacity is decreased by 1 unit per captured frame (initial battery capacity is set to 100 units). The performance evaluation was realized with 2 fixed frame capture rate scenarios (3 fps and 6 fps) and 3 levels of application criticality: $r^0 = 0$ (low criticality), $r^0 = 0.5$ and $r^0 = 1$ (high criticality) where a node's frame capture rate depends on the size of its cover set. Nodes with 6 or more cover sets will capture at the maximum speed. Simulation ends as soon as the subset of nodes with power left is disconnected (when all active nodes have no neighbors anymore). We run simulations 15 times to reduce the impact of randomness.

Figure 6 shows the average frame capture rate of all active nodes per round which is representative of the surveillance application quality during the network lifetime.

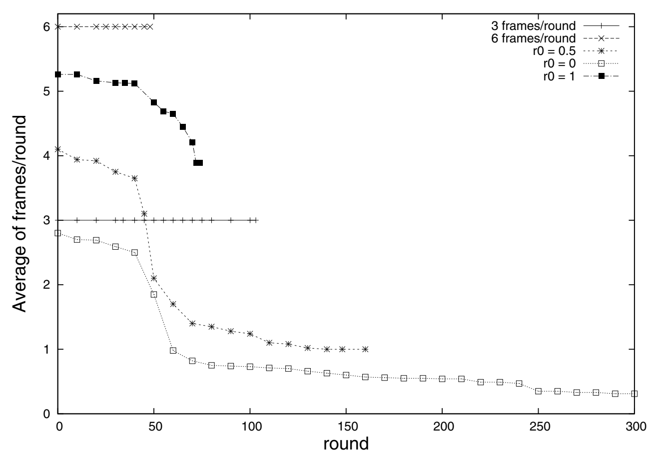

Fig. 6. Average frame capture rate per round.

As the application criticality r^0 is varied the frame capture rate of each sensor node that depends on the size of its cover set is modified according to the behavior function. Figure 7 shows the percentage of coverage while varying the frame capture rate. We define the full area coverage as the region covered initially by the whole network (i.e when all the deployed nodes are active). This area represents the union of all FoV areas of the deployed nodes. Now, according to the multiple cover sets approach described in [9], and used in our simulations, not all nodes are active and 100% coverage can be achieved with less than 40% of nodes. Nodes with high capture rate will use more battery power until they run out of battery after some rounds. In this case, the cover sets they belong to will not be a valid cover set anymore for some nodes.

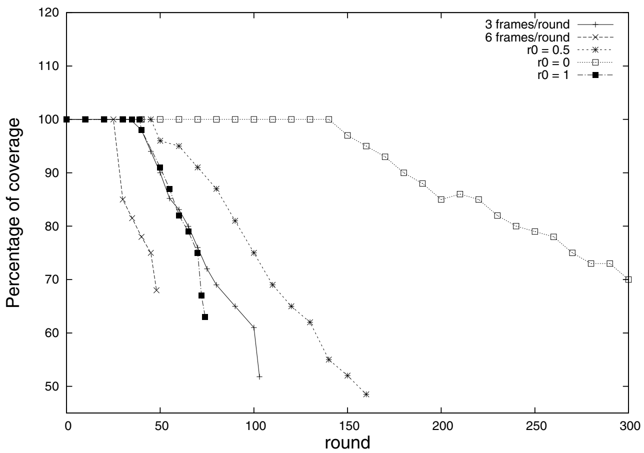

Fig. 7. Percentage of coverage while varying the frame capture rate.

We can see in figure 6 that the high criticality scenario gives a mean frame capture rate of 4.63 fps which should draw in figure 7 a curve that lies between the 6 fps and the 3 fps curves. However, figure 7 shows a high criticality curve very close to the 3 fps curve. These results, although preliminary, show that besides providing a model for translating a subjective criticality level into a quantitative parameter of the surveillance system, our proposed approach for visual sensor nodes can also optimize the resource usage by dynamically adjusting the provided service level. Figure 8 shows in more details the 4.63 fps (fixed rate version) and the $r^0 = 1$ curves.

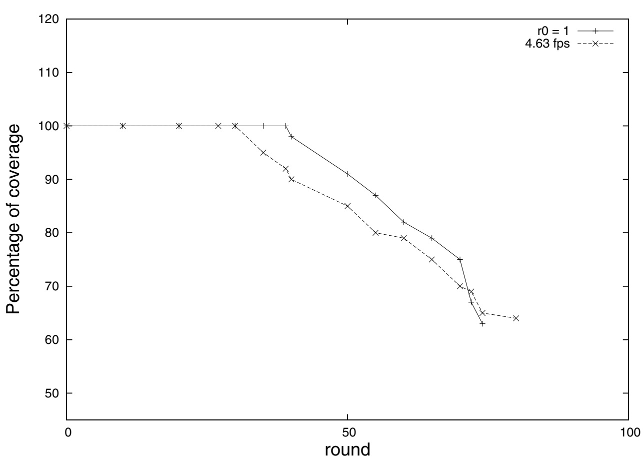

Fig. 8. Average frame capture rate per round.

V. Conclusion

This paper addresses the problem of scheduling randomly deployed video sensor nodes for critical surveillance applications. Based on a coverage model that handle FoV redundancies by providing multiple covers per sensor node, we proposed a multiple levels activity model that uses behavior functions modeled by modified Bezier curves to define application classes and allow for adaptive scheduling. Besides providing a model for translating a subjective criticality level into a quantitative parameter of the surveillance system, our proposed approach for video sensor nodes can also optimize the resource usage by dynamically adjusting the provided service level. We also present how our adaptive scheduling model can take into account the risk level in intrusion detection systems in order to automatically and dynamically adjust the video sensor node's frame capture rate according to the environment stimulus. Future works will investigate in more details how such risk levels could be dynamically managed in the deployed video sensor network.

Acknowledgment

This work is partially supported by the ANR JCJC TCAP project of the French National Research Agency and by the Pyrénées-Atlantique Regional Council.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci., "Wireless sensor networks: a survey." *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[2] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-efficient surveillance system using wireless sensor networks," in *MobiSys*, 2003, pp. 270–283.

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

[5] E. H. Gupta, S. R. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query," in *ACM MobiHoc*, 2003.

[6] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Connected sensor cover: Self-organization of sensor networks for efficient query," in *ACM SenSys*, 2003.

[7] S. Shakkottai, R. Srikant, and N. B. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," in *IEEE INFOCOM*, 2003.

[8] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *SenSys*, 2003, pp. 51–62.

[9] A. Makhoul and C. Pham, "Dynamic scheduling of cover-sets in randomly deployed wireless video sensor networks for surveillance applications," in *TR, University of Pau, France*, 2009.

[10] S. Kumar, T. H. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *MobiCom'05*, 2005, pp. 284–298.

[11] S. Kumar, T.-H. Lai, M. E. Posner, and P. Sinha, "Optimal sleep-wakeup algorithms for barriers of wireless sensors." in *BROADNETS*. IEEE, 2007, pp. 327–336.

[12] R. Saadi, O. Hasan, J. Pierson, and L. Brunie, "Establishing trust beliefs based on a uniform disposition to trust," *IEEE international Conference on Signal Image and Technologie Based Systems*, 2007.

[13] OMNeT++, "http://www.omnetpp.org/."