Fast event detection in mission-critical surveillance with wireless video sensor networks

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

***Abstract*—Wireless Video Sensor Network (WVSN) can be used for efficient event detection such as intrusion detection or disaster relief systems. These applications have a high level of criticality and can not be deployed with the current state of technology. In this paper, we show how a dynamic criticality management scheme can provide fast event detection for mission-critical surveillance applications. Based on a criticality model that uses modified Bezier curves to determine for each sensor node the corresponding frame capture rate, simulation results show that the network lifetime can be increased, that the stealth time is kept low and that sentry nodes succeed in detecting intrusions.**

***Index Terms*—Sensor networks, video surveillance, coverage, event detection, mission-critical applications**

I. Introduction

This paper focuses on Wireless Video Sensor Networks (WVSN) where sensor nodes are equipped with miniaturized video cameras. We consider WVSN for mission-critical surveillance applications where sensors can be thrown in mass when needed for intrusion detection or disaster relief applications. Surveillance applications have very specific needs due to their inherently critical nature associated to security [1], [2], [3], [4]. Early surveillance applications involving WSN have been applied to critical infrastructures such as production systems or oil/water pipeline systems [5], [6]. 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. On one hand, it is desirable that most sensor nodes move to a so-called *hibernate* mode in the absence of events in order to save energy. On the other hand, it is also highly desirable that some sensor nodes still keep a relatively high capture rate in order to act as sentry nodes in the surveillance system to better detect intrusions/events and to alert other active nodes to move to an *alerted* mode. With video sensors the higher the capture rate is, 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. Therefore, a common approach is to define a subset of the deployed nodes to be active while the other nodes can sleep. In [7], [8] the idea we developed is that when a node has several covers, it can increase its frame capture rate to act as a sentry node because if it runs out of energy it can be replaced by one of its covers. Then, depending on the applications's criticality, the frame capture rate of those nodes with large number of cover sets can varied: a low criticality level indicates that the application does not require a high video frame capture rate while a high criticality level does. [8] also proposed to apply a risk-based approach for scheduling sensor nodes: 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 [9], the authors introduce so-called differentiated services by dynamically modify the time duration for a node to work during each round. The authors in [10] propose to probabilistically support flexible QoS [10] without over-provisioning resources. 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 [9].

Based on the criticality models developed in [8], this article presents the performance of dynamic criticality management for fast event detection in mission-critical application. Therefore the main issue that is addressed in this paper is to demonstrate that our dynamic risk-based approach for scheduling sensor nodes can provide increases network lifetime and reduced intrusion detection time. The rest of the paper is organized as follows. Section II quickly presents the dynamic criticality management models. We then present the main contribution of this paper that focuses on fast event detection in section III. We conclude in section IV.

II. Criticality-based scheduling of randomly deployed nodes with cover sets

Our framework for enabling fast event detection in mission-critical applications operates in 3 phases. In the first phase each sensor broadcasts its position. Only one message per sensor node is required and we assume GPS facilities. In the second phase each node v constructs its set of cover sets $Co(v)$. Interested readers can refer to [11] for more details on fast cover set construction techniques. The third phase is the scheduling phase where each node decides to be active or in sleep mode. Phases 1 and 2 occur only once at the beginning of the network lifetime, unless mobility is provided. As said previously, the frame capture rate is an important parameter that defines the surveillance quality. In [8], 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 applications. This risk level can oscillate from a concave to a convex shape as illustrated in Figure 1 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 1 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 1 box B), most projections of x values are gathered close to the max frame capture rate.

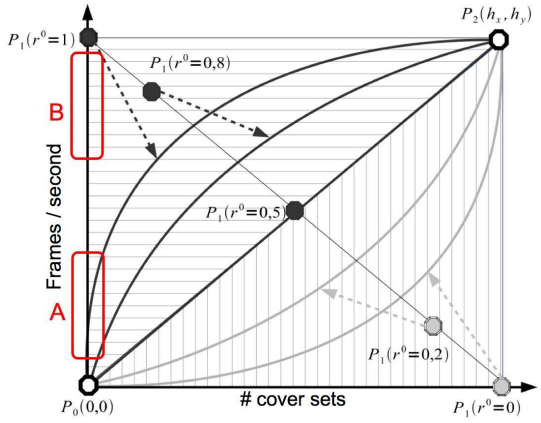

Fig. 1. The Behavior curve functions

[8] proposes 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 criticality) or concave (low criticality) 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 illustrated in Figure 1, 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. 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
0	.01	.02	.05	0.1	.17	.26	.38	.54	.75	1.1	1.5	3
.2	.07	.15	.25	.37	.51	.67	.86	1.1	1.4	1.7	2.2	3
.4	.17	.35	.55	.75	.97	1.2	1.4	1.7	2.0	2.3	2.6	3
.6	.36	.69	1.0	1.3	1.5	1.8	2.0	2.2	2.4	2.6	2.8	3
.8	.75	1.2	1.6	1.9	2.1	2.3	2.5	2.6	2.7	2.8	2.9	3
1	1.5	1.9	2.2	2.4	2.6	2.7	2.8	2.9	2.9	2.9	2	3

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

III. FAST EVENT DETECTION WITH DYNAMIC CRITICALITY MANAGEMENT

We used the OMNET++ discrete event simulator (http://www.onmetpp.org) to randomly deploy 150 sensor nodes in a $75m * 75m$ area. Sensors have an 36^o AoV. Each sensor node captures with a given number of frames per second (between 0.01fps and 3fps) according to the model defined in figure 1. Nodes with 12 or more cover sets will capture at the maximum speed. Simulation ends when there are no active nodes anymore. Figure 2 shows the percentage

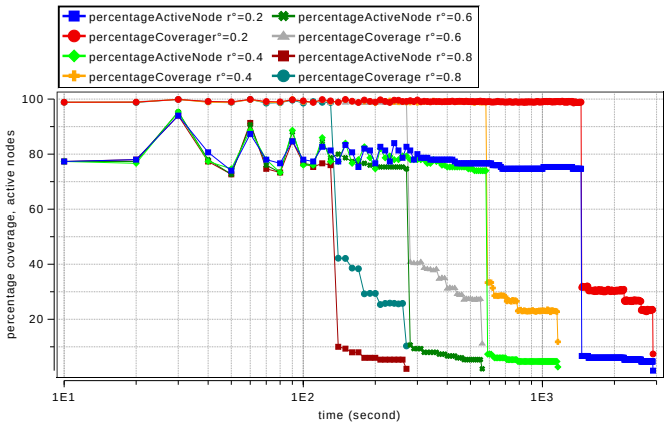

Fig. 2. Percentage of coverage and actives nodes as r^0 is varied.

of coverage and the percentage of active nodes for 4 levels of criticality: $r^0 = 0.2$, 0.4, 0.6 and 0.8. The corresponding capture rates are those shown in table I. The x-axis is in logarithmic scale. We define the full area coverage as the region covered initially by the whole network (i.e when all the deployed nodes are active). Nodes with high capture rate will use more battery power until they run out of battery. In this case, the cover sets they belong to will not be a valid cover set anymore for the other nodes. 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.

In order to show the benefit of the adaptive behavior, we computed the mean capture rate for each of the simulations of figure 2 and then used that value as a fixed capture rate for all the sensor node in the simulation model. $r^0 = 0.2$ gives a mean capture rate of 0.32fps, $r^0 = 0.4$ gives 0.56fps, $r^0 = 0.6$ gives 0.83fps and $r^0 = 0.8$ gives 1.18fps. The results of the fixed frame capture rate are illustrated in figure 3.

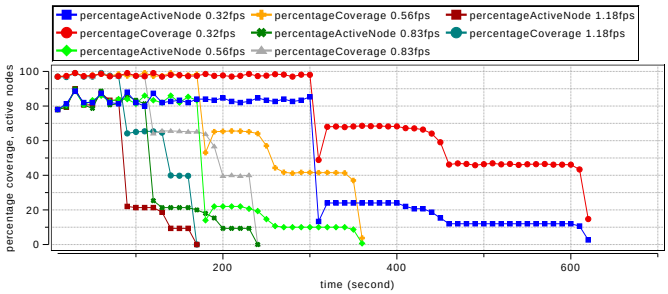

Fig. 3. Percentage of coverage and actives nodes with fixed capture rate.

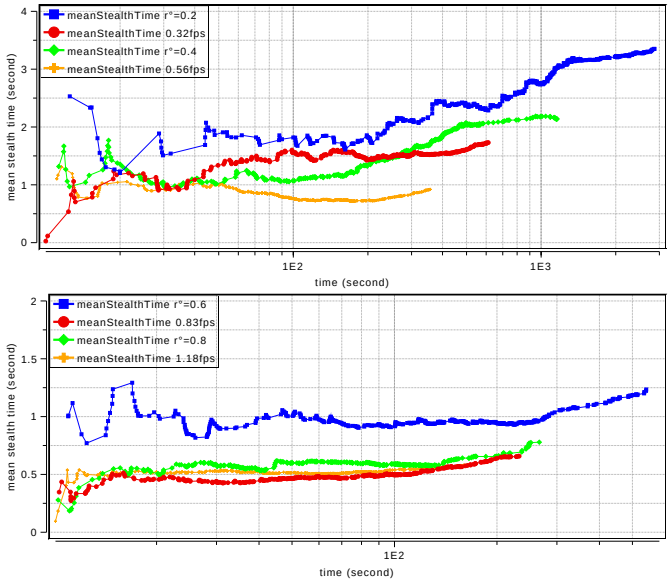

Fig. 4. Mean stealth time. Top: $r^0 = 0.2$, $fps = 0.32$, $r^0 = 0.4$, $fps = 0.56$. Bottom: $r^0 = 0.6$, $fps = 0.83$, $r^0 = 0.8$, $fps = 1.18$.

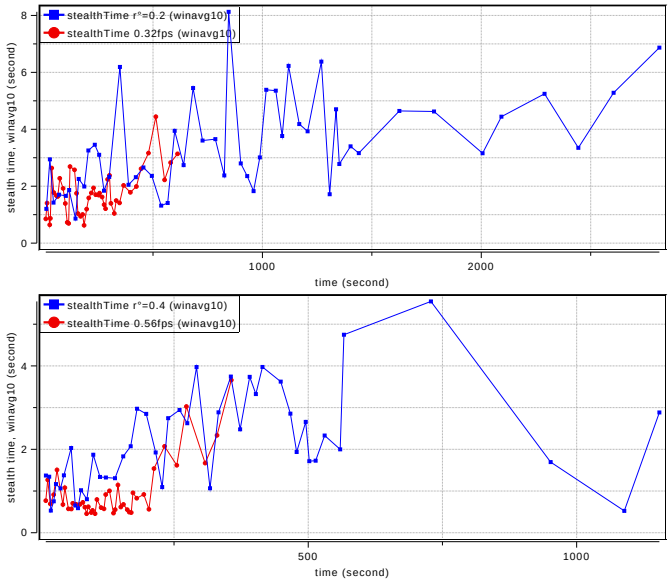

Fig. 5. Stealth time, winavg with 10-sample batch. $r^0 = 0.2$, $fps = 0.32$ (top). $r^0 = 0.4$, $fps = 0.56$.

We can see that using the adaptive frame rate is very efficient as the network lifetime is $2900s$ for $r^0 = 0.2$ while the 0.32fps fixed capture rate last only $620s$. However, in order to evaluate the quality of surveillance we show in figure 4(top) the mean stealth time when $r^0 = 0.2$, $fps = 0.32$, $r^0 = 0.4$ and $fps = 0.56$, and in figure 4(bottom) the case when $r^0 = 0.6$, $fps = 0.83$, $r^0 = 0.8$ and $fps = 1.18$. 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 mean stealth time computed. Then a new intrusion appears at another random position. This process is repeated until the simulation ends. Figures 5 and 6 plot the stealth time using a window average filter of 10 samples.

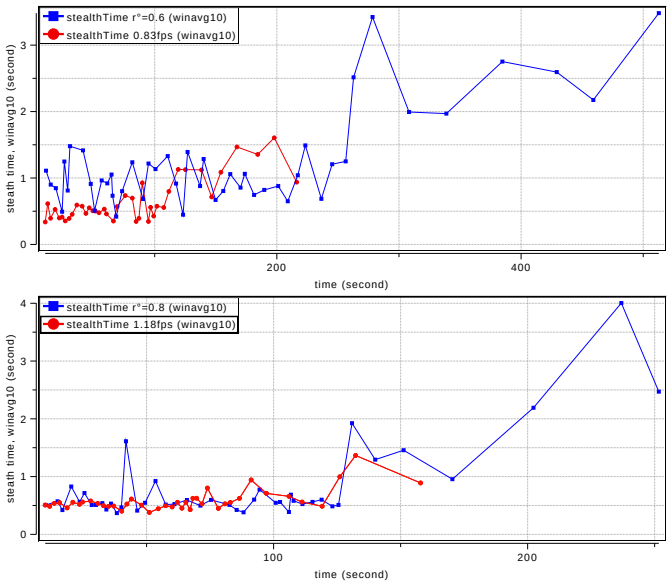

Fig. 6. Stealth time, winavg with 10-sample batch. $r^0 = 0.6$, $fps = 0.83$ (top). $r^0 = 0.8$, $fps = 1.18$.

For the particular case of disambiguation, we introduce a 8m.4m rectangle at random positions in the field. The rectangle has 8 significant points as depicted in figure 7 and moves at the velocity of 5m/s in a scan line mobility model (left to right). Each time a sensor node covers at least 1 significant point or when the rectangle reaches the right boundary of the field, it appears at another random position. This process starts at time $t = 10s$ and is repeated until the simulation ends. The purpose is to determine how many significant points are covered by the initial sensor v and how many can be covered by using one of v's cover set. For instance, figure 7 shows a scenario where v's FoV covers 3 points, the left cover set ($\{v_3, v_1, v_4\}$) covers 5 points while the right cover set ($\{v_3, v_2, v_4\}$) covers 6 points.

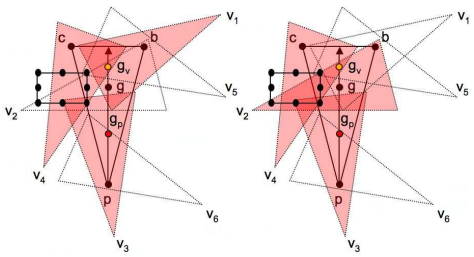

Fig. 7. Rectangle with 8 significant points. v and 2 different cover sets.

In the simulations, each time a sensor v covers at least 1 significant point of the intrusion rectangle, it determines how many significant points are covered by each of its cover sets. The minimum and the maximum number of significant points

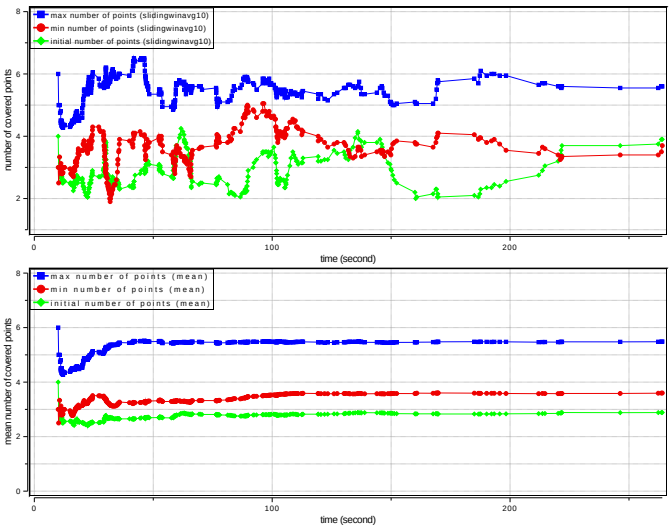

Fig. 8. Number of covered points of an intrusion rectangle. Sliding winavg of 20-sample batch (top), mean (bottom).

covered by v's cover sets are recorded along with the number of significant points v was able to cover initially. Figure 8 shows these results. The top part shows the values using a sliding window averaging filter with a batch window of 20 samples. The bottom part shows the evolution of the mean value. We can see that node's cover sets always succeed in identifying more significant points.

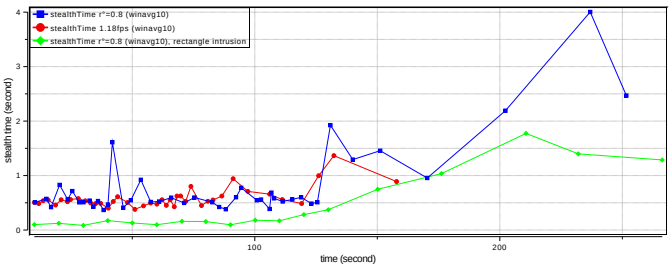

Fig. 9. Stealth time, winavg with 10 samples batch. $r^0 = 0.8$, $fps = 1.18$ and $r^0 = 0.8$ with rectangle intrusion.

Figure 9 shows that with the rectangle intrusion, that could represent a group of intruders instead of a single intruder, the stealth time is greatly reduced.

With a criticality level of $r^0 = 0.8$, the surveillance quality of very high while the network lifetime is nearly two times longer than the fixed capture rate scenario. Figure 10 shows that our sentry node selection strategy succeeds in enabling fast detection of intruders in the field. The left part of the figure shows the sensors' position and their respective number of cover sets. Those nodes with a high number of cover sets will capture faster according to table I. The right part of the figure shows the number of intrusions detected by each node. The bigger the dot, the higher the number of detected intrusions by that node is. We can clearly see that there is a strong relation between nodes with high number of cover sets and those that have been able to detect the intrusions.

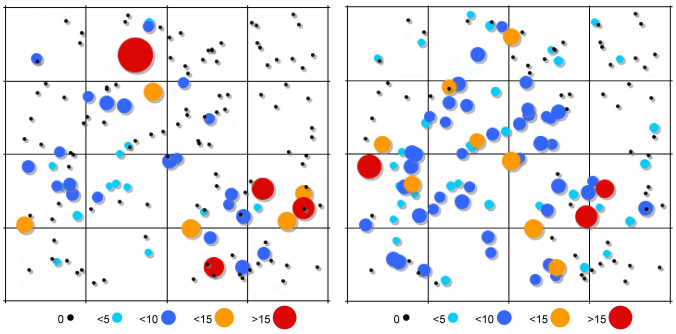

Fig. 10. Node's cover set size and node's detected intrusion number. 150 nodes, $r^0 = 0.8$.

IV. Conclusion

This paper presented the performance results of a dynamic criticality management model that enable fast event detection for mission-critical surveillance applications with video sensor networks. We show that our approach saves energy and improves the network lifetime while providing small stealth time in case of an intrusion detection system. 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.

Acknowledgment

This work is partially supported by the Pyrénées-Atlantique Council and by the PHC Tassili project 09MDU784.

References

[1] 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 *ACM MobiSys*, 2004.

[2] 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.

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

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

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

[6] 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.

[7] 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.

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

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

[10] Y. Zhu and L. M. Ni, "Probabilistic approach to provisioning guaranteed qos for distributed event detection," in *ACM MSWIM*, 2007.

[11] 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.