2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Fault Tolerant Event Localization in Sensor Networks using Binary Data

Michalis P. Michaelides
Dept. of Electrical and Computer Engineering
University of Cyprus
Nicosia, Cyprus
michalism@ucy.ac.cy

Christos G. Panayiotou
Dept. of Electrical and Computer Engineering
University of Cyprus
Nicosia, Cyprus
christosp@ucy.ac.cy

Abstract— **This paper investigates the use of wireless sensor networks for estimating the location of an event that emits a signal which propagates over a large region. In this context, we assume that the sensors make binary observations and report the event if the measured signal at their location is above a threshold; otherwise they remain silent. Based on the sensor binary beliefs we use 4 different estimators to localize the event: CE (Centroid Estimator), ML (Maximum Likelihood), SNAP (Subtract on Negative Add on Positive) and AP (Add on Positive). The main contribution of this paper is the fault tolerance analysis of the proposed estimators. Furthermore, the analysis shows that SNAP is the most fault tolerant of all estimators considered.**

I. Introduction

Localization of the event position has been extensively studied in the last 20 years using arrays of sensors for radar, sonar and acoustic target tracking applications (see [1], [2], [3] and references therein). A variety of techniques have been proposed to solve the localization problem that can be classified into 3 main categories: 1. DOA (Direction of Arrival) 2. TDOA (Time Difference of Arrival) 3. Energy-based. More recently, the use of a WSN (Wireless Sensor Network) has been proposed to deal with a number of environmental monitoring and tracking applications including acoustic source localization, toxic source identification, early detection of fires and so on [4], [5], [6]. This new promising technology also comes with unique challenges and constraints that have not been adequately addressed by the existing methods: energy efficiency, network latency and fault tolerance (see Collaborative Signal Information Processing in [7]).

Sensor nodes are expected to be low-cost, simple devices with limited resources (processing capabilities, memory, and power). In this context, we decided to only consider estimators that use binary data to localize the event. Binary is the "easiest" problem a sensor node can solve by simply comparing its measurement to a predefined threshold. Binary decisions are also less sensitive to calibration mismatches and varying sensor sensitivities. Moreover, using binary observations limits the bandwidth usage and conserves energy; only single-bit information needs to be transmitted from the sensor nodes with positive observations for estimating the event location.

In this paper we specifically address *Fault Tolerance*. The simple nature of sensor nodes makes them extremely vulnerable to faults. These faults can occur for a variety of reasons: noise, energy depletion, environmental harsh conditions of operation, software problems. These have been reported in real experiments and can result in erroneous, unexpected behavior (Byzantine faults). For example, the authors in [8] report situations where a node would constantly detect "false events" when its program board was overheated, or scenarios in which nodes were programmed undetectably in an incorrect manner which yielded Byzantine behavior. Another source of faults is the network unreliability that can result in a high percentage of dropped packets. These can be attributed to the uncertain nature of the communication medium, as well as collisions due to the dense deployment of the sensor networks. Robustness to all kinds of faults is essential for any estimation algorithm, so it can tolerate a number of misbehaving nodes and a percentage of dropped packets in the network. Fault tolerance has been studied for event detection using sensor networks (see [9]). It has been shown that a majority voting rule for detection provides the desired robust behavior to a large percentage of faults. For estimation, however, this problem has not received the necessary attention.

The main contribution of this paper is the fault tolerance analysis of 4 different estimators that only use binary data from the sensor nodes to localize the event: CE (Centroid Estimator), ML (Maximum Likelihood), SNAP (Subtract on Negative Add on Positive) and AP (Add on Positive). Specifically, we show that SNAP (and its variant AP) demonstrates the desired robust behavior in the presence of a large percentage of faults.

The paper is organized as follows. First, in Section II, we present the model we have adopted and the underlying assumptions. Then, in Section III, we provide the details of the 4 different estimation algorithms. Section IV shows the fault tolerance analysis of the proposed estimators. Section V presents several simulation results. Finally, this paper concludes with Section VI.

This work is partly supported by the Cyprus Research Promotion Foundation under contract ΠΛΗΡΟ/0505/10.

II. Model

For the sensor network that estimates the position of an event we are going to make the following assumptions:

1) A set of N sensor nodes is uniformly spread over a rectangular field of area A. The nodes are assumed stationary. The position of each node is denoted by (x_n, y_n), $n = 1, \cdots, N$ and it is assumed that it is known through the use of a combination of GPS and localization algorithms.
2) A single source of the event is located according to a uniform distribution at a position (x_s, y_s) inside A.
3) The source emits a continuous signal that propagates uniformly in all directions and there are no environmental changes throughout the propagation.
4) The event has been correctly detected by the network. We assume that the sensor nodes have been programmed with a common threshold T and they become alarmed when their measurement exceeds this threshold.
5) Only the alarmed sensors send a packet to the base station (sink); the rest remain silent.
6) The network does not drop packets.

Assumptions 1, 2 and 5 are quite common and reasonable for sensor networks. Assumption 3 defines a propagation model that may be accurate for sources that emit sound or electromagnetic waves, but it is not very accurate for problems where an actual substance is released in the environment (for example in problems of environmental pollution). Regarding Assumption 4, readers are referred to [10] or [11], where the detection problem in the context of WSN is addressed. Finally, Assumption 6 may be a little restrictive in the context of "unreliable" WSN. This assumption is relaxed in Section V, where we show that with the exception of ML, all other algorithms considered exhibit robust behavior even when the network drops a significant number of packets (e.g., due to noise or collisions).

For this paper, we assume that the measured signal at the source is c and as we move away from the source, the signal is attenuated inversely proportional to the distance from the source raised to some power $\alpha \in \mathbb{R}^+$ which depends on the environment. As a result, the t-th sample measurement of any sensor n located at (x_n, y_n) is given by

$$z_{n,t} = s_n + w_{n,t}, \tag{1}$$

for $n = 1, \cdots, N$, $t = 1, \cdots, M$, where

$$s_n = \frac{c}{1 + r_n^\alpha}, \tag{2}$$

while r_n is the radial distance from the source, i.e.,

$$r_n = \sqrt{(x_n - x_s)^2 + (y_n - y_s)^2} \tag{3}$$

For the purposes of this paper $w_{n,t}$ is additive white Gaussian noise, i.e. $w_{n,t} \sim N(0, \sigma_w^2)$ for $n = 1, \cdots, N$ and $t = 1, \cdots, M$.

Given the threshold T for any t, we define:

- *Alarmed Sensor*: Any sensor with $z_{n,t} \geq T$ (positive observation).
- *Non-Alarmed Sensor*: Any sensor with $z_{n,t} < T$ (negative observation).

Next, we define the *Region of Influence* (ROI), as the area around the source location inside which a sensor node will be alarmed with high probability. For the model of (2) the ROI is a disc centered at the source location (see Fig. 1).

Finally, to define the *Fault Model* that we adopt in this paper we assume that the sensor nodes that exhibit erroneous behavior are randomly chosen and their original belief is simply reversed as shown in the example of Fig. 1. When applying the fault model, some sensor nodes that fall outside the ROI become alarmed as a result of a fault and are shown as *false positives*. Similarly, sensors that fall inside the ROI become non-alarmed as a result of a fault and are shown as *false negatives*.

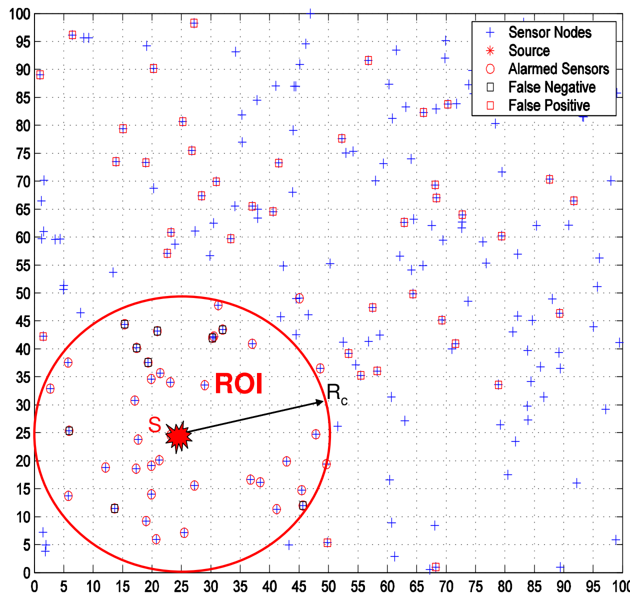


Fig. 1. A field with 200 randomly placed sensor nodes and a source placed at position (25,25). Alarmed sensors are indicated on the plot with red circles inside the disc around the source (ROI). 50 of the sensor nodes exhibit faulty behavior and are indicated on the plot as false positives (red squares outside disc) and false negatives (black squares inside disc).

III. Binary Estimators

We investigate 4 different estimators for localizing the event using only binary data: CE, ML, SNAP and AP. Below we provide the details of each of these algorithms.

A. Centroid Estimator (CE)

The centroid of a finite set of points can be computed as the arithmetic mean of each coordinate of the points. Let (x_n, y_n), $n = 1, \cdots, P$ $(P \leq N)$ denote the positions of all alarmed sensor nodes. Then, the event location estimated by CE is the centroid of these positions:

$$\hat{\theta}_{CE} = [\hat{x}_s, \hat{y}_s] = \left[\frac{1}{P} \sum_{n=1}^{P} x_n, \frac{1}{P} \sum_{n=1}^{P} y_n \right] \tag{4}$$

B. Maximum Likelihood

We define the indicator function for $n = 1, \cdots, N$ and $t = 1, \cdots, M$:

$$I_{n,t} = \begin{cases} 0, & \text{if } z_{n,t} < T \\ 1, & \text{if } z_{n,t} \geq T \end{cases} \tag{5}$$

thus, the sensor data can be represented as $\mathbf{I} = \{I_{n,t} : n = 1, \cdots, N, t = 1, \cdots, M\}$. The goal is to estimate the source location $\theta = [x_s, y_s]$ using the collected data $\mathbf{I}$.

The Maximum Likelihood Estimator has the form:

$$\hat{\theta}_{ML} = \max_{\theta} \log p(\mathbf{I} \mid \theta) \tag{6}$$

where the log-likelihood function is given by:

$$\log p(\mathbf{I} \mid \theta) = \sum_{n=1}^{N} \sum_{t=1}^{M} I_{n,t} \times \log \left[Q\left(\frac{T - s_n(\theta)}{\sigma_w} \right) \right] + (1 - I_{n,t}) \times \log \left[1 - Q\left(\frac{T - s_n(\theta)}{\sigma_w} \right) \right] \tag{7}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{t^2}{2}} dt$ is the complementary distribution function of the standard Gaussian distribution. Also, $s_n(\theta)$ is the signal that would have been measured by sensor n if the source was at location θ and there was no noise (given by (2)). The derivation details for the case where $\sigma_w^2 = 1, \forall n, \forall t$ can be found in [12]. Assuming an arbitrary σ_w^2 is a natural extension of this work.

For the purposes of this paper we use a discrete version of the algorithm described above. Specifically, we divide the area in $G \times G$ equal size grid-cells and evaluate the log-likelihood function G^2 times using (7) assuming the source is located at the center of each cell (i, j). The maximum of the resulting matrix points to the event location. From now on, for the rest of the paper we will refer to this discrete version of the algorithm as simply ML (Maximum Likelihood).

C. SNAP (Subtract on Negative Add on Positive)

The SNAP algorithm can be described in 4 steps: 1. Grid Formation 2. Region of Coverage 3. Likelihood Matrix 4. Maximization. Next, we provide the main idea of the algorithm by explaining the various steps:

1) Grid Formation: The entire area is divided into a grid. The number of cells is a tradeoff between estimation accuracy and complexity. Each sensor node is associated with a cell (i, j) based on its position (depending on the resolution a cell may contain multiple sensors or no sensors at all).

2) Region of Coverage (ROC): For each sensor node we define a neighborhood of cells around the sensor node location that we call the Region of Coverage. Inside the cells of its ROC, each sensor node outputs a value based on its binary observation (+1 on positive observation and -1 on negative). For the model in (2), it can be shown that setting $ROC \equiv ROI$ produces the best estimation results when using SNAP. For the purposes of this paper, the ROC of a sensor n is approximated by a square region around the center of the cell (i, j) as shown in Fig. 2.

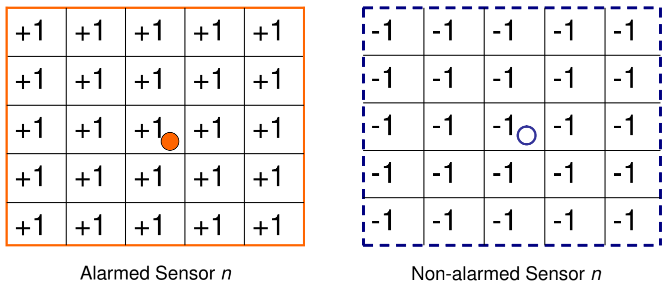


Fig. 2. Region of Coverage (ROC)

3) Likelihood Matrix $\mathcal{L}$: For each cell of the area grid we calculate the likelihood of a source occurring in the particular cell by summing the ROC of the respective sensor nodes. In other words, we *add on positive observations and subtract on negative*. Thus, the elements of the likelihood matrix are obtained by

$$L(i,j) = \sum_{n=1}^{N} \sum_{t=1}^{M} b_{n,t}(i,j), \text{ for } i,j = 1, \cdots, G, \tag{8}$$

where,

$$b_{n,t}(i,j) = \begin{cases} +1, & \text{if } z_{n,t} \geq T \text{ AND } (i,j) \in ROC_n \\ -1, & \text{if } z_{n,t} < T \text{ AND } (i,j) \in ROC_n \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

and ROC_n represents the region of coverage of sensor node n. Fig. 3 shows an example of the resulting likelihood matrix after adding and subtracting the contributions of 8 sensors using SNAP.

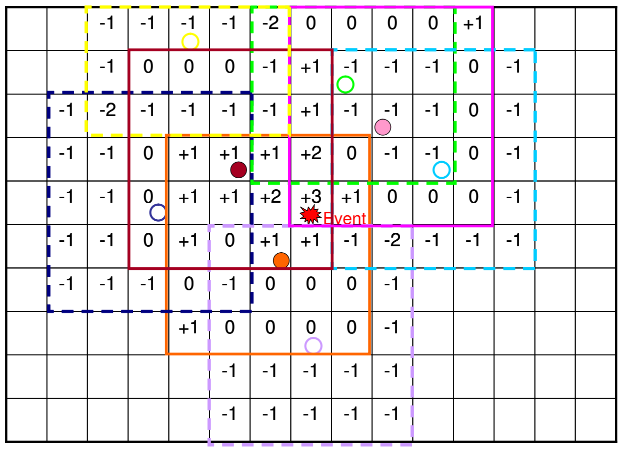


Fig. 3. $\mathcal{L}$ resulting from SNAP with 8 sensor nodes, 3 of which are alarmed and are shown in solid color. The event is correctly localized in the grid cell with the maximum value +3.

4) Maximization: The maximum of this likelihood matrix points to the estimated event location. If more than one elements of the $\mathcal{L}$ matrix have the same maximum value, the estimated event position is the centroid of the corresponding cell centers.

D. Add on Positive (AP)

This algorithm is similar to SNAP in the sense that we define a "likelihood matrix" but in this matrix we only add $+1$ contributions from the alarmed sensors inside the ROI of the source. The purpose of this algorithm is to demonstrate the value of using the negative information from non-alarmed sensors when using SNAP.

IV. Fault Tolerance

In this section we investigate the behavior of the proposed estimators in the presence of faults. We start with a test case.

A. Test case

Consider the 1-D scenario displayed in Fig. 4 where the line is divided into 20 equal cells. The event is located at position ($x_s = 9.5$) and we try to estimate its position using 4 sensor nodes which are located as shown in Fig. 4(a). Two of the sensor nodes fall inside the source ROI (which for this example it is assumed that it spans 9 cells) and are alarmed ($x_n = 5.5, 13.5$), the other two fall outside and are non-alarmed ($x_n = 4.5, 14.5$). According to the SNAP algorithm described in Section III-C, the alarmed sensor nodes provide $+1$ contribution in the cells inside their ROC (it is assumed that ROC=ROI), while the non-alarmed sensors provide -1 contribution. The sum of the contributions in each cell i gives the likelihood $L(i)$ of the source occurring in that cell and is plotted over the corresponding cells in Fig. 4(a) above the sensor locations. The maximum of this likelihood plot occurs in the cell where the source is located. In fact, in the absence of faults all 4 estimation algorithms correctly estimate the event position.

Now consider a fifth sensor node which is faulty and behaves according to the fault model described in Section II (i.e., it is non-alarmed when positioned inside the source ROI and alarmed everywhere else). Fig. 4(b) shows the estimated source location using the 4 different algorithms, as we vary the position of the faulty sensor node. From the plot, it is evident that only SNAP displays a fault tolerant behavior for *all* positions of the faulty sensor node; from Fig. 4(a) it is clear that the maximum of $L(i)$ cannot change no matter where we place the faulty sensor node. ML fails to estimate the source location when the faulty sensor node is close to the event (false negative), while both CE and AP fail to estimate the event location when the faulty sensor node is far away (false positive). Next, we present some further insight in the operation of each algorithm in the presence of a fault.

B. CE

CE as seen by the previous example, is especially sensitive to the presence of false positives that can appear at random locations *away* from the source because of noise or malfunction due to overheating. Since the algorithm essentially treats all alarmed sensor nodes with equal weight, as seen by (4), it is especially sensitive to false positives that occur far away from the true event location. These can result in large errors when calculating the centroid of the alarmed sensor nodes' positions.

C. ML

ML is extremely sensitive to false negatives. Even a single faulty sensor node inside the ROI of the source can completely throw off the estimation results. This is a direct result of the construction of the likelihood matrix of ML using (7). Note that for source positions θ close to the sensor, the term $Q\left(\frac{T-s_n(\theta)}{\sigma_w}\right) \to 1$ and as a result $\log\left[1 - Q\left(\frac{T-s_n(\theta)}{\sigma_w}\right)\right] \to -\infty$. If for some reason (e.g., due to a fault), a sensor fails to detect an event that is very close to it (false negative), then $(1 - I_{n,t}) = 1$ and as result the faulty sensor has a *very* large negative contribution to the likelihood function at all points near itself which (by assumption) is also the point where the source is located. On the other hand, a healthy sensor near the source contributes only $I_{n,t} \times \log Q\left(\frac{T-s_n(\theta)}{\sigma_w}\right) \to 0$ which is not sufficiently large to counteract the negative contribution of the faulty sensor. In fact, since the negative contribution of the faulty sensor is unbounded, even several well behaving sensors cannot correct the error.

D. SNAP

The fault tolerant behavior of SNAP results from 2 main observations: First, to construct the likelihood function at a source location θ it uses only "local" information in the sense that it uses only the data from the sensors that are inside the ROI of θ. Therefore, false positives away from the source location have no influence on the estimation results. Second, and most important, it bounds the "damage" that a faulty sensor can cause to the likelihood function by allowing a sensor to subtract *at most* one from the corresponding cells in the likelihood matrix. Furthermore, unlike ML, a *single* healthy sensor close to the source can correct the error in the likelihood function caused by the faulty sensor.

E. AP

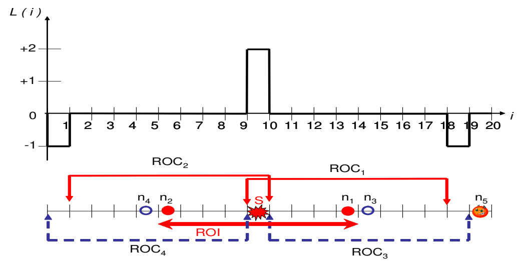
As a variant of SNAP, AP inherits the fault tolerant behavior of SNAP, however, because it "ignores" the information provided by non-alarmed sensors its overall performance (in terms of estimation accuracy) is inferior to the performance of SNAP. At this point it is worth pointing out the results of Fig. 4(b) are a little "unfair" for AP since the one dimensional example has only 4 well behaving sensors. As indicated in the simulation results, for a more realistic two-dimensional case, AP outperforms both the ML and CE for a fairly large range of faulty sensors.

V. Results

For all subsequent experiments we use a square 100×100 sensor field with 200 sensor nodes where the sensor readings are given by:

$$z_{n,t} = \frac{c}{r_n^2} + w_{n,t} \tag{10}$$

for $n = 1, \cdots, N$, $t = 1, \cdots, M$. Furthermore, we assume $w_{n,t}$ to be white Gaussian noise $N(0, 1)$, $M = 1$, $c = 3000$, and $T = 5$. Finally, the RMS error reported is the average over 500 Monte-Carlo simulations. For all experiments we use Matlab.


(a)

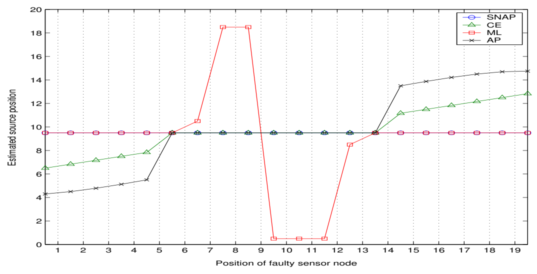

(b)

Fig. 4. 1-D example with 5 sensor nodes, one of which is faulty. The source is correctly estimated using SNAP for all positions of the faulty sensor node. ML fails to estimate the source position when the faulty sensor node is close to the source (false negative). CE fails to estimate the source position when faulty sensor is far away (false positive).

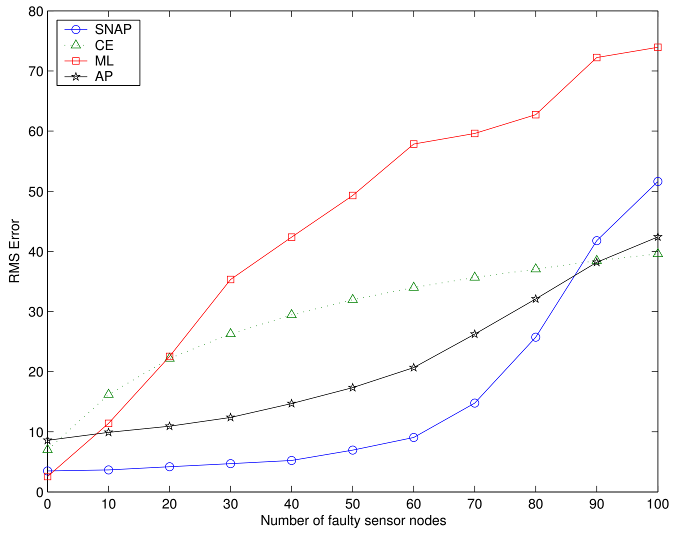


Fig. 5. Estimator performance vs. number of faulty sensor nodes

A. Fault Tolerance

For the fault tolerance analysis we use the fault model in Section II. The faulty sensor nodes are chosen randomly at the beginning of each experiment and their original beliefs are simply reversed (see Fig. 1). Fig. 5 displays the estimation error as a function of the number of faulty sensors in the field. In the absence of faults, both ML and SNAP display the best performance compared to the other estimators (CE and AP). The same cannot be stated, however, as we start introducing faulty sensor nodes. In fact both CE and ML are very sensitive to sensor faults and lose accuracy continuously as the number of faults increase. This is especially evident for ML. SNAP however, as it can be observed from these plots, displays a fault tolerant behavior and loses very little in accuracy even when 50 out of the 200 sensor nodes exhibit erroneous behavior. In fact, its fault tolerance improves when we increase the percentage of alarmed sensor nodes in the field. The intelligent construction of the likelihood function makes individual sensor faults unimportant in estimating the correct result. Finally, AP displays similar fault tolerant behavior to SNAP but the estimation error is larger than SNAP to begin with (even in the absence of faults). Note however, that for a range of faulty sensors from 10 to 80, AP exhibits better performance than the ML and CE.

B. Dropped Packets

In this section we relax the assumption that the network does not drop any packets and investigate the performance of the four algorithms if packet drops are allowed. Recall that for the network we investigate, alarmed sensors send a packet to the sink while non-alarmed sensors remain silent. Thus, if the sink does not receive a packet from a node, it assumes that the node is in the non-alarmed state. Therefore, to investigate the effect of dropped packets we change the fault model to allow *only* alarmed sensors to randomly flip their state (false negatives) with probability P_d which corresponds to the probability of dropping a packet.

In Fig. 6 we investigate the performance of the estimators in the presence of false negatives. ML, as expected from the analysis in Section IV, loses accuracy immediately in the presence of false negatives. SNAP is the best estimator for values of $P_d \leq 0.3$. For higher percentages of dropped packets, CE becomes the best option. This is due to the increasing number of false negatives that “exterminate” the small number of correctly alarmed sensor nodes left in the field when using SNAP. AP does not have this problem, so it displays a similar robust behavior to false negatives as CE. For the CE and AP, even one correctly alarmed sensor can localize the source at its own location since both algorithms completely neglect the non-alarmed sensor nodes in computing the event location.

C. Board Overheating

For symmetry, in this section we also investigate the effect of the board overheating which, as reported in [8], caused the nodes to report false events. In Fig. 7 we investigate the performance of the estimators in the presence of false positives. We simulate the presence of false positives by varying the probability P_o that a non-alarmed sensor node produces a positive observation. CE displays the worst performance in the presence of false positives; this is expected from the analysis in Section IV. SNAP on the other hand, displays the most robust behavior in the presence of these overheating faults. The importance of the non-alarmed sensor nodes for SNAP in correcting false positives, is also evident by looking at the large difference in performance between SNAP and AP. ML steadily loses performance, and for values of $P_o \geq 0.4$ it becomes even worse than AP.

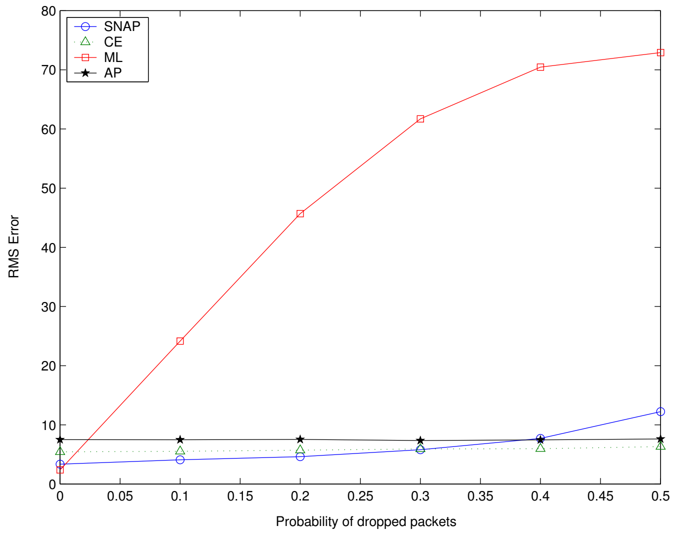


Fig. 6. Estimator performance vs. probability of dropped packets P_d

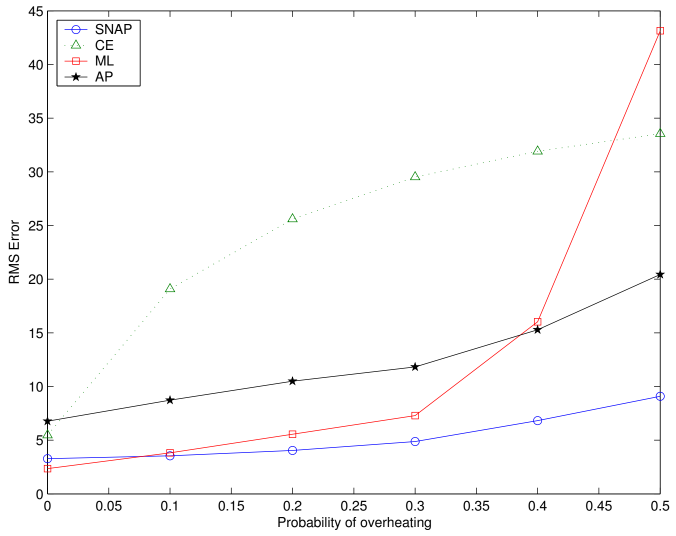


Fig. 7. Estimator performance vs. probability of overheating P_o

VI. Conclusions

This paper investigates the fault tolerance of 4 different estimators that can be applied for localizing an event in a sensor network given only binary data from the sensor nodes: CE (Centroid Estimator), ML (Maximum Likelihood), SNAP (Subtract on Negative Add on Positive) and AP (Add on Positive). Out of the four, SNAP has a superior performance in terms of estimation accuracy in the presence of faults.

References

[1] J. Chen, R. Hudson, and K. Yao, "Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1843–1854, Aug. 2002.

[2] X. Sheng and Y.-H. Hu, "Energy based acoustic source localization," in *Information Processing in Sensor Networks (IPSN), Second International Workshop, Palo Alto, CA, USA*, April 2003, pp. 286–300.

[3] D. Li, K. Wong, Y. Hu, and A. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 3, pp. 17–29, Mar. 2002.

[4] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. San Francisco, CA: Morgan Kaufmann, 2004.

[5] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, pp. 102–114, Aug. 2002.

[6] C. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[7] S. Kumar, F. Zhao, and D. Shepherd, "Collaborative signal and information processing in microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, pp. 13–14, Mar. 2002.

[8] A. Arora et al., "A line in the sand: a wireless sensor network for target detection, classification and tracking," *Computer Networks*, vol. 46, pp. 605–634, 2004.

[9] B. Krishnamachari and S. Iyengar, "Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241–250, March 2004.

[10] M. P. Michaelides and C. G. Panayiotou, "Event detection using sensor networks," in *45th IEEE Conference on Decision and Control*, Dec. 2006, pp. 6784–6789.

[11] R. Niu, P. Varshney, M. Moore, and D. Klamer, "Decision fusion in a wireless sensor network with a large number of sensors," in *7th IEEE International Conference on Information Fusion (ICIF'04)*, Stockholm, Sweden, June 2004.

[12] R. Niu and P. Varshney, "Target location estimation in wireless sensor networks using binary data," in *38th Annual Conference on Information Sciences and Systems*, Princeton, NJ, March 2004.