Distributed Fault-Tolerance for Event Detection Using Heterogeneous Wireless Sensor Networks

ElMoustapha Ould-Ahmed-Vall and George F. Riley and Bonnie S. Heck
School of Electrical and Computer Engineering, Georgia Institute of Technology
Atlanta, GA 30332-0250
{eouldahm,riley,bonnie.heck}@ece.gatech.edu

***Abstract*— Distributed event detection using wireless sensor networks has received growing interest in recent years. In such applications, a large number of inexpensive and unreliable sensor nodes are distributed in a geographical region to make firm and accurate local decisions about the presence or absence of specific events based on their sensor readings. However, sensor readings can be unreliable, due to either noise in the sensor readings or hardware failures in the devices, and may cause nodes to make erroneous local decisions. We present a general fault-tolerant event detection scheme that allows nodes to detect erroneous local decisions based on the local decisions reported by their neighbors. This detection scheme does not assume homogeneity of sensor nodes and can handle cases where nodes have different accuracy levels. We prove analytically that the derived fault-tolerant estimator is optimal under the maximum a posteriori (MAP) criterion. An equivalent weighted voting scheme is also derived. Further, we describe two new error models that take into account the neighbor distance and the geographical distributions of the two decision quorums. These models are particularly suitable for detection applications where the event under consideration is highly localized. Our fault-tolerant estimator is simulated using a network of 1024 nodes deployed randomly in a square region and assigned random probability of failures.**

I. Introduction

A sensor network consists of a set of sensing elements powered by batteries and collaborating to perform sensing tasks in a given environment. It may contain one or more sink nodes (base stations) to collect sensed data and relay it to a central processing and storage system. These networks have potential for use in many military and civilian applications [1].

One particular application that has received a growing amount of attention in the recent years is event detection [2], [3], [4], [5], [6]. In such an application, nodes are tasked to determine whether a particular event of interest is occurring in their sensing range. Such an event could be, for example, a volcanic eruption at specific site [5], or the presence of a specific target [6]. An event could be detected from a high value of the sensor reading, for example. Each sensor node first determines if its sensor reading indicates the presence of an event before sending this information to its neighbors or to a sink node. However, in case of failure the sensor can produce a false positive or a false negative. That is, a high reading indicating an event occurred when it did not or a low reading indicating the absence of event when one did occur.

Event detection is commonly performed using a large number of unreliable low-cost sensor nodes. These nodes can each have a very high probability of errors (misses and false-positives). It is, therefore, important to develop fault-tolerant mechanisms that can detect detection faults and take appropriate actions. A possible solution is to provide a high degree of redundancy to compensate for faulty nodes. However, the cost sensitivity and energy limitation of sensor networks make such an approach impractical [7].

In this environment, collaboration between neighboring nodes can be used to increase the reliability of the detection decisions. This is valid if we assume that failures at neighboring nodes are not correlated, yet there is a spatial correlation between occurrences of detected events at local nodes. In other words, a failure at node n is independent from failures at any of its neighbors. On the other hand, the presence of the detected event (e.g., a chemical agent) at node n is highly correlated with the situation at its neighbors.

Here, we address fault-tolerance in the context of distributed binary detection. A node n is trying to decide whether or not a specific event is present within its coverage range. A binary variable is used to code this decision, with a value of 1 when an event is detected, and a value of 0 otherwise. In its decision scheme, the node uses the sensed data obtained by its local sensor as well as the decisions at its neighboring nodes, assuming spatial correlations.

Distributed fault-tolerance for event detection using the assumption of spatial correlation was first considered in [8]. The algorithm in [8] assumes that all nodes in the network have the same detection error probability and that this rate is known prior to the deployment. These are unrealistic assumptions. In fact, a node can become faulty with time either because of a lower energy level or because of aging or unsuitable environmental or operating conditions, thereby increasing its error probability. We can also have a heterogeneous sensor network with nodes that have different operational capabilities and accuracy levels. Moreover, the proposed algorithm in [8] is not well suited for highly localized events where the event region is very small. In fact, all nodes within a node communication range are given identical weights in the decision scheme regardless of their distances .

The work in [8] has been followed by two other publications dealing with the same problem of event region detection. In reference [9], the authors provide comments on the original paper and correct some of the mistakes in the theoretic analysis section. In [10], the authors extend the model in [8] to account for the fact that sensor errors have two different sources. An error could be noise-related or coming from a sensor fault. They also discuss the choice of the appropriate neighborhood size. However, they assume again that neighboring nodes of n at any distance have the same accuracy as estimators of the real situation at n. In such a case, the failure probability of the distributed decision scheme can be reduced by increasing the neighborhood size. Again, this is an unrealistic assumption and will introduce a large number of new errors in the case of a highly localized event. In reference [10] as in [8], it is assumed that nodes all have the same probability of failure and that this probability is known prior to the deployment.

We propose a new approach that considers the case where nodes can have different failure probability levels. This allows us to handle various types of failures including noise-related failures, biased measurement, drift over time, stuck-at failures, calibration-related failures, environment-related failures, etc. Our approach can be used as a general distributed fault-tolerance mechanism for any application where nodes may have different accuracy levels. These differences can result from different locations, heterogeneous operating conditions (different sensors, different hardware conditions), different deployment times, etc.

We also consider two new distributed error models that take into account the location and relative position of sensor nodes. The first model takes into account the fact that nodes that are closer to each other have a higher spatial correlation than nodes that are farther apart. The second model accounts for the importance of the relative geographical distributions of the two voting quorums (the two subsets of neighbors deciding the presence of an event or its absence, respectively). In this model, if a node has 50% of its neighbors reporting the same decision (e.g., '1': event detected) there is a difference in terms of the value of such a decision depending on whether these neighbors are geographically distributed around the node or are all on the same side. This comes from the observation that an event detected in all sides of a node is more likely to be present at the node itself than an event that was detected by nodes only on one side of the node.

The remainder of the paper is as follows. Section II formulates the fault-tolerance problem and describes our approach to solve this problem. Section III presents a distance-based error model. Section IV presents an error model that accounts for the relative geographical distribution of the decision quorums. Section V presents some simulation results. Section VI concludes the paper.

II. The Distributed Fault-Tolerance Solution with Different Probabilities of Failure

In this section we describe our solution used to provide fault-tolerance for distributed event detection while taking into account the possibility of nodes having different accuracy levels. The accuracy levels can differ for several reasons:

1) Nodes may have heterogeneous sensors with different quality levels. This leads to different probability of failures (miss probability and false alarm probability) at different nodes
2) Some nodes may suffer degradations during the deployment process. For example, in the case of a forest where the sensor network is deployed by dropping nodes from the air, nodes may suffer different impact effects degrading the quality of the sensor readings. This can result in different detection failure probabilities between neighboring nodes
3) Failure probabilities may be a function of the node distance from the detected object. In this case, if a node n is implementing fault-tolerance using the correlation between its local decision and those of its neighbors, it is possible that the neighbors closer to n give a more accurate estimate (lower probability of failure) of the real situation at n than would the neighbors located farther from n
4) Failure probabilities may be function of the sensor age. In this case, the sensor performance degrades over time and the sensor may become biased or suffer a gradual drift. The sensor can also remain stuck at the same value independently of the event reality. If using a reconfiguration mechanism, such as the one in [11], nodes that have been active for different periods of time will have different failure probability levels
5) Nodes accuracy may be affected differently in the presence of changing environment conditions such temperature, rain, snow, etc.

In this paper, we assume that a node n has a way of learning, through estimation, the different failure probabilities at its neighboring nodes. We have proposed a simple method that allows nodes to learn their probability of failure in [12]. Future work will address this issue. Further, our solution does not assume a specific probability distribution of the faulty sensor readings such as Gaussian as is assumed in [8], [10]. We also do not assume that the accuracy level of a specific node remains constant over time or that all nodes have the same accuracy levels. Relaxing these assumptions makes our solution more realistic and enables it to handle all sources of failures as long as nodes in a specific region are not all faulty.

A. Problem Formulation

We consider a sensor network composed of N nodes distributed over a detection field. Each node has a sensor and is tasked to detect the presence of a specific event. This decision is made using the node's sensor reading compared to a fixed threshold. For simplification, we consider that the presence of an event corresponds to a high sensor reading, while a low reading indicates its absence. An error occurs when a high reading is reported in the absence of an event (false positive) or when a low reading is obtained even though an event is occurring (detection miss). Errors could be due to noisy measurements or a faulty sensor [10]. Here, we treat errors as a single group regardless of the error origin.

Consider that the mean value of the sensor reading in the presence of an event is m_e, while in the absence of event the mean value is m_n. A reasonable threshold value is given by

$$th = \frac{m_e + m_n}{2} \quad (1)$$

We define the following three binary variables, similar to the ones in [8].

- $T_n(t)$: indicates the real situation at the node n and time t (presence or not of an event)
- $S_n(t)$: indicates the situation as obtained from the sensor reading of node n. It could be wrong in the case of failure
- $R_n(t)$: gives an estimate of the real value of $T_n(t)$ using the $S(t)$ values of the node and its neighbors

The probability of detection error p_n, at node n, is given by:

$$p_n = P(S_n(t) = a | T_n(t) \neq a) \quad (2)$$

For example, if we assume a Gaussian error term (e.g., noise-related error) with a mean of 0 and variance of σ_n^2 at node n, then the probability of detection error is given by:

$$p_n = Q(\frac{m_e - m_n}{2\sigma_n}) \quad (3)$$

where Q is the tail probability of the Gaussian distribution.

The problem at hand is to define an estimator of the real situation at node n that minimizes the detection error probability. This estimator takes into account the local decision obtained from the node sensor reading as well as the local decisions of the neighboring nodes. We consider a sensor network, $Ne = \{1, 2, 3, ..., N-1, N\}$ containing N sensor nodes. The nodes taken into account by a node $n \in Ne$ in its decision mechanism are all nodes within a fixed range, r. This fault-tolerance range, r, should be fixed so as to minimize the probability of error while keeping the communication energy cost low and taking into account the expected size of the event region. Below, we give a formal definition of this neighborhood.

Definition 1. *We define the fault-tolerance neighborhood (FTN_n) as the set of nodes that a node $n \in Ne$ takes into account in its fault-tolerance decision mechanism. If we consider a fault-tolerance range of r, this neighborhood is given by: $FTN_n = \{ne \in Ne : d(n, ne) \leq r\}$, where $d(n, ne)$ is the Euclidean distance between the nodes n and ne. This set contains the node n itself.*

Below, we define the decision vector taken into account by a node n in its fault-tolerant mechanism. We also define the probability of distributed detection error.

Definition 2. *The estimation fault-tolerance vector (FTV_n) is defined as the vector containing all the $S_n(t)$ of n and the $S_j(t)$ of all its fault-tolerance neighbors. FTV_n contains K elements, where K is the number of elements in FTN_n. We have that $FTV_n^k(t) = S_m(t)$, where m is the k^th element of FTN_n. This set contains, in particular the value $S_n(t)$ since $n \in FTN_n$.*

Definition 3. *The probability of estimator detection error at node n and time t is defined as:*

$$Pe_n(t) = P(R_n(t) \neq T_n(t)|T_n(t), FTV_n) \tag{4}$$

Using these definitions, the problem at hand consists of finding an estimation function that takes as an input the vector FTV_n and gives as an output R_n that minimizes the probability of error Pe_n.

B. Optimal Estimator for Fault-Tolerant Distributed Detection

To develop an optimal estimation function, we use the likelihood test ratio (LRT) [13]; that is, to choose $R_n(t) = j$ where $j \in \{0, 1\}$ that maximizes the probability $P(T_n(t) = j|FTV_n(t))$.

Below, we define the power set containing all possible sets composed of any subset of the neighbors of node n and the node n itself.

Definition 4. *We define P_n as the set of all possible FTV_n vectors. P_n can be represented by the power set of the set FTN_n, where a value of 1 in the k^{th} position of a vector $v \in P_n$ indicates the presence of the corresponding node (the k^{th} node in FTN_n) in the subset.*

Define the parameter $p(t)$ as the probability of the true situation being the presence of an event at time t.

$$\begin{aligned} P(T_n(t) = 1) &= p(t) \\ P(T_n(t) = 0) &= 1 - p(t) \end{aligned} \tag{5}$$

Next, we define the following two functions on the set P_n.

Definition 5. *The following two functions F_n^0 and F_n^1 are defined as follows:*

$$\begin{aligned} &F_n^j : P_n \to \mathbb{R}^+, j \in \{0, 1\} \\ &F_n^j(v) = P(T_n(t) = j) \prod_{k|v(k)=j} \frac{1 - p_{FTN_n(k)}}{p_{FTN_n(k)}} \end{aligned} \tag{6}$$

where $v \in P_n$ is the current value of the FTV_n vector, and $v(k)$ and $FTN_n(k)$ give the k^{th} elements of the vectors v and FTN_n, respectively. Note that $FTN_n(k)$ corresponds to the k^{th} node in the FTN_n set consisting n and its neighbors and $p_{FTN_n(k)}$ is the local detection error probability of this node defined in equation 2.

The function F_0 represents the product of the elements $\frac{1-p_k}{p_k}$ for all nodes reporting a local decision of 0 in the set containing n and its neighbors multiplied by the probability of non-occurrence of the event. The function F_1 gives the same product for nodes reporting a local decision of 1. We can now define the optimal estimator that minimizes the probability of detection error. This estimator is given in the following definition.

Definition 6. *We define the following fault-tolerant estimator for distributed detection (FTEDD) as an estimator that declares $R_n(t) = 0$ if and only if $F_0(FTV_n(t)) < F_1(FTV_n(t))$. The estimator declares $R_n(t) = 1$, otherwise.*

The optimality of this estimator is proven in the next theorem.

Theorem 1. *The FTEDD estimator is optimal with respect to the maximum a posteriori (MAP) criterion.*

Proof. For the two possible hypotheses, $T_n(t) = 0$ and $T_n(t) = 1$, the conditional probability given $FTV_n(t)$ can be obtained using the Bayes' rule. The two posterior probabilities are given by:

$$P(T_n(t) = j|FTV_n(t)) = \frac{P(FTV_n(t)|T_n(t) = j)P(T_n(t) = j)}{P(FTV_n(t))} \tag{7}$$

where $j \in \{0, 1\}$. The value of $P(FTV_n(t))$, the probability of occurrence of the current value of $FTV_n(t)$ is given by:

$$P(FTV_n(t)) = \sum_{j=0}^{1} P(FTV_n(t)|T_n(t) = j)P(T_n(t) = j) \tag{8}$$

We have that the local detection decision for a node $k \in FTN_n$ is correct with a probability of $1 - p_k$. Using this information and the fact that the events of errors in the local decisions are independent, it is clear that when the real situation is $T_n(t) = j$, the nodes reporting a local decision of $S_k(t) = j$ are correct, while the others are faulty. This gives the following conditional probabilities:

$$\begin{aligned} P(FTV_n(t)|T_n(t) = 0) &= \prod_{k|k \in FTN_n} p_k^{S_k(t)}(1 - p_k)^{1-S_k(t)} \\ P(FTV_n(t)|T_n(t) = 1) &= \prod_{k|k \in FTN_n} p_k^{1-S_k(t)}(1 - p_k)^{S_k(t)} \end{aligned} \tag{9}$$

These equations multiply the correctness probability $(1 - p_k)$ for nodes reporting the correct value (j) by the error probability for nodes reporting the opposite value.

We can now compute the posteriori probabilities as follows:

$$\begin{aligned} P(T_n(t) = 1|FTV_n(t)) &= \frac{P(FTV_n(t)|T_n(t) = 1)P(T_n(t) = 1)}{P(FTV_n(t))} \\ &= \frac{p(t) \prod_{k|k \in FTN_n} p_k^{1-S_k(t)}(1 - p_k)^{S_k(t)}}{P(FTV_n(t))} \end{aligned} \tag{10}$$

and:

$$\begin{aligned} P(T_n(t) = 0|FTV_n(t)) &= \frac{P(FTV_n(t)|T_n(t) = 0)P(T_n(t) = 0)}{P(FTV_n(t))} \\ &= \frac{(1 - p(t)) \prod_{k|k \in FTN_n} p_k^{S_k(t)}(1 - p_k)^{1-S_k(t)}}{P(FTV_n(t))} \end{aligned} \tag{11}$$

We can now compute the likelihood ratio as:

$$\begin{aligned} \gamma &= \frac{P(T_n(t) = 0|FTV_n(t))}{P(T_n(t) = 1|FTV_n(t))} \\ \gamma &= \frac{(1 - p(t)) \prod_{k|k \in FTN_n} p_k^{S_k(t)}(1 - p_k)^{1-S_k(t)}}{p(t) \prod_{k|k \in FTN_n} p_k^{1-S_k(t)}(1 - p_k)^{S_k(t)}} \end{aligned} \tag{12}$$

It can be easily seen that:

$$\gamma = \frac{1 - p(t)}{p(t)} \prod_{k|k \in FTN_n} (\frac{p_k}{1 - p_k})^{S_k} (\frac{1 - p_k}{p_k})^{1-S_k(t)} \tag{13}$$

This equation can be re-written as:

$$\begin{aligned} \gamma &= \frac{1 - p(t)}{p(t)} \prod_{k|k \in FTN_n} \frac{(\frac{1-p_k}{p_k})^{1-S_k(t)}}{(\frac{1-p_k}{p_k})^{S_k(t)}} \\ \gamma &= \frac{(1 - p(t)) \prod_{k|k \in FTN_n} (\frac{1-p_k}{p_k})^{1-S_k(t)}}{p(t) \prod_{k|k \in FTN_n} (\frac{1-p_k}{p_k})^{S_k(t)}} \end{aligned} \tag{14}$$

Since $1-S_k(t)=0$ when $S_k(t)=1$, the numerator in the previous equation can be written as follows:

$$
\begin{aligned}
&(1-p(t))\prod_{k|k\in FTN_n}(\frac{1-p_k}{p_k})^{1-S_k(t)}\\
&=(1-p(t))\prod_{k|v(k)=0}\frac{1-p_{FTN_n(k)}}{p_{FTN_n(k)}}=F_0(v)
\end{aligned}
\tag{15}
$$

where $v=FTV_n(t)$. Similarly, the denominator corresponds to $F_1(v)$. We can, therefore, re-write γ as:

$$\gamma=\frac{F_0(FTV_n(t))}{F_1(FTV_n(t))} \tag{16}$$

The estimator minimizes the error if it estimates $R_n(t)=0$ when $\gamma>1$ [13], which is equivalent to deciding based on the maximum a posteriori (MAP) criterion. This corresponds to the case of $F_0(FTV_n(t))>F_1(FTV_n(t))$. This completes the proof of the optimality of the FTEDD estimator. □

The following corollaries give the detection error probability of the FTEDD estimator and determine whether it is biased or not.

Corollary 1. *The probability of detection error at node n and time t is given by:*

$$
\begin{aligned}
Pe_n(t)=p_n(t)&\sum_{v\in\Omega_0}\prod_{k|k\in FTN_n}p_k^{1-S_k(t)}(1-p_k)^{S_k(t)}\\
+(1-p_n(t))&\sum_{v\in\Omega_1}\prod_{k|k\in FTN_n}p_k^{S_k(t)}(1-p_k)^{1-S_k(t)}
\end{aligned}
\tag{17}
$$

where $\Omega_0=\{v\in P_n: F_0(v)>F_1(v)\}$ and $\Omega_1=\{v\in P_n: F_1(v)>F_0(v)\}$.

Proof. An error occurs when the estimator decides a value $R_n(t)$ that is different from the real value $T_n(t)$. The probability of detection error is therefore given by:

$$
\begin{aligned}
Pe_n(t)&=P(R_n(t)\neq T_n(t)|T_n(t),FTV_n(t))\\
&=P(T_n(t)=1)P(R_n(t)=0|T_n(t)=1,FTV_n(t))\\
&+P(T_n(t)=0)P(R_n(t)=1|T_n(t)=0,FTV_n(t))
\end{aligned}
\tag{18}
$$

If $v=FTV_n(t)$, the conditional probabilities are computed as follows:

$$
\begin{aligned}
P(R_n(t)=0|T_n(t)=1,v)&=P(F_0(v)>F_1(v)|T_n(t)=1,v)\\
&=\sum_{v\in\Omega_0}\prod_{k|k\in FTN_n}p_k^{1-S_k(t)}(1-p_k)^{S_k(t)}
\end{aligned}
\tag{19}
$$

In a similar way, we have:

$$
\begin{aligned}
P(R_n(t)=1|T_n(t)=0,v)&=P(F_1(v)>F_0(v)|T_n(t)=0,v)\\
&=\sum_{v\in\Omega_1}\prod_{k|k\in FTN_n}p_k^{S_k(t)}(1-p_k)^{1-S_k(t)}
\end{aligned}
\tag{20}
$$

And since $P(T_n(t)=1)=p_n(t)$ and $P(T_n(t)=0)=1-p_n(t)$, we obtain the desired result. □

Corollary 2. *Suppose that for all nodes $k\in Ne$, the probability of local detection error $0<p_k<\frac{1}{2}$, then the FTEDD estimator is biased. However, the estimator is asymptotically unbiased.*

Proof. This corollary comes from the observation that because $0<p_k<\frac{1}{2}$, we will always have a non-null probability of error: $Pe_n(t)=P(R_n(t)\neq T_n(t)|T_n(t),FTV_n(t))>0$ for a finite number of neighbors. The expected value of the estimator decision is given by:

$$
\begin{aligned}
E(R_n(t)|T_n(t),FTV_n(t))&=T_n(t)P(R_n(t)=T_n(t)|T_n(t),FTV_n(t))\\
&+(1-T_n(t))P(R_n(t)\neq T_n(t)|T_n(t),FTV_n(t))
\end{aligned}
\tag{21}
$$

Since $P(R_n(t)\neq T_n(t)|T_n(t),FTV_n(t))>0$, we have that $E(R_n(t)|T_n(t),FTV_n(t))\neq T_n(t)$. The estimator is, therefore, biased. However, as the number of elements in the FTN_n increases, the probability of error gets closer to 0 since $p_k<\frac{1}{2}$ for all nodes in the network. In this case, $E(R_n(t)|T_n(t),FTV_n(t))$ approaches the real situation, $T_n(t)$, and the estimator becomes unbiased. □

The next theorem allows the expression of the FTEDD estimator as a weighted voting scheme [14] and provides the corresponding node weights. This is in contrast with the majority and $k-out-of-n$ schemes used in [8].

Theorem 2. *When the presence or absence of an event are equally likely ($p_n(t)=\frac{1}{2}$), the FTEDD estimator is equivalent to a weighted voting scheme of the nodes in FTN_n. A node $k\in FTN_n$ has the following weight:*

$$w_k=\ln\frac{1-p_k}{p_k},\forall k\in FTN_n \tag{22}$$

Proof. We have that when $1-p_n(t)=p_n(t)=\frac{1}{2}$, the likelihood ratio is given by:

$$\gamma=\prod_{k|k\in FTN_n}(\frac{p_k}{1-p_k})^{S_k}(\frac{1-p_k}{p_k})^{1-S_k(t)} \tag{23}$$

Since $\frac{1-p_k}{p_k}=e^{w_k}$, we can write γ in the following form:

$$
\begin{aligned}
\gamma&=\prod_{k|k\in FTN_n}(\frac{1}{e^{w_k}})^{S_k}(e^{w_k})^{1-S_k(t)}=\prod_{k|k\in FTN_n}(e^{w_k})^{1-2S_k(t)}\\
&=\prod_{k|k\in FTN_n}e^{w_k(1-2S_k(t))}=e^{(\sum_{k|k\in FTN_n}w_k(1-2S_k(t)))}
\end{aligned}
\tag{24}
$$

It is clear that $\gamma>1$ is equivalent to $\sum_{k|k\in FTN_n}w_k(1-2S_k(t))>0$. This sum can be written in the following way by replacing the values of $1-2S_k(t)$ with 1 or -1 depending on the values of S_n^k:

$$
\begin{aligned}
&\sum_{k|k\in FTN_n}w_k(1-2S_k(t))\\
&=\sum_{k\in FTN_n|S_k(t)=0}w_k-\sum_{k\in FTN_n|S_k(t)=1}w_k
\end{aligned}
\tag{25}
$$

So $\gamma>1$ is equivalent to $\sum_{k\in FTN_n|S_k(t)=0}w_k>\sum_{k\in FTN_n|S_k(t)=1}w_k$. This corresponds to a weighted majority vote in favor of the hypothesis of $T_n(t)=0$. This completes the proof of the theorem. □

III. Distance-Based Error Model for Fault-Tolerant Distributed Detection

In this section, we present a new fault-tolerant event detection scheme that uses a distance-based error model. This scheme allows us to account for the fact that the evidences coming from two different neighbors of n do not necessarily have the same importance when used to estimate the real situation at node n. In fact, the correlation between the real situation at node n and the situation at a node $n_1\in FTN_n$ is higher than the correlation with the situation n_2 when $d(n,n_1)<d(n,n_2)$. Here, $d(n,n_i)$ is the Euclidean distance between the two nodes.

We use a model inspired by the distance-based signal model in [15]. Consider a node n_k at distance d from an event site. If the true sensor reading of a node collocated at the event site is e_0, then the average sensor reading at n_k can be modeled as:

$$e = \frac{be_0}{d^a} \tag{26}$$

where the parameters a and b represent the attenuation factors and are function of the event propagation characteristics, size of the event region and the deployment terrain properties. Example values are $b = 1$ and $a = 2$ in the absence of obstacles. However, the values of a and b depend on terrain characteristics and propagation properties [15].

To take the neighbor distance into account, we define a new weighted voting model that gives a weight factor to each neighbor that is a function of its relative distance to n compared to other neighbors.

Definition 7. *We define the distance weight wd_k as the weight given to the node $k \in FTN_n$ as follows:*

$$wd_k = 1 + \frac{d(n,k)}{\sum_{m|m \in FTN_n} d(n,m)} \tag{27}$$

And the node weight in the voting scheme is given by:

$$wn_k = wd_k w_k \tag{28}$$

where w_k represents the original node weight, defined previously in theorem 2.

This node is not ideal, since it does not give to each neighbor k a weight corresponding to the exact probability of detection error when using a node $k \in FTN_n$ to estimate the real situation at node n. The computation of this probability requires the assumption that the sensor readings follow a specific probability distribution model, which is not assumed here for the purpose of generality. For example if we assume a Gaussian error term of mean 0 and variance σ_n^2, then the probability of detection error when using a node $k \in FTN_n$ to estimate the real situation at node n is given in the next proposition.

Proposition 1. *Assuming a Gaussian error term, the probability of error when an event occurring at node n is detected by a node $k \in FTN_n$ is given by:*

$$p_k^n = Q(\frac{bm_e}{\sigma_k(d(n,k))^a} - Q^{-1}(p_k)) \tag{29}$$

where m_e is the mean value in the presence of event and $d(n,k)$ is the distance between the two nodes.

Proof. This proposition comes from the assumption that the node k uses the threshold defined in equation 1. In this case an error occurs when an event occurs and is not detected or an event is detected when no event did occur. These two probabilities are equal. Using the mean sensed value in presence of event m_e instead of e_0 in equation 26. The detection error is therefore:

$$\begin{aligned} p_k^n &= Q(\frac{\frac{bm_e}{d(n,k)^a} - th}{\sigma_k}) = Q(\frac{\frac{bm_e}{d(n,k)^a} - (\frac{m_e - m_n}{2})}{\sigma_k}) \\ &= Q(\frac{\frac{bm_e}{d(n,k)^a}}{\sigma_k} - \frac{m_e - m_n}{2\sigma_k}) \end{aligned} \tag{30}$$

And since $\frac{m_e - m_n}{2\sigma_k} = Q^{-1}(p_k)$, we obtain the result in the proposition. □

We note that by using this Gaussian error model, different nodes have different perceived error probabilities for a node n depending on their distances from n. These probabilities are normally different from the local probability at n. In this scheme, the error probabilities p_k^n can be used instead of the values of p_k by node n to compute each neighbor weight for the decision scheme in definitions 5 and 6 and theorem 2.

IV. Group-Based Error Model for Fault-Tolerant Distributed Detection

In this section, we develop an error model that takes into account in the fault-tolerance mechanism the geographical distribution of the group of neighbors reporting a specific detection decision. To illustrate this idea, we consider the following examples. We assume, for simplification, that all nodes have the same probability of error p. In the first example, nodes 1, 2, 3 in Figure 1 report a detection decision of 1. In the second example, nodes 2, 4, 6 report the same decision. The idea here is that even though the same number of neighbors of n reported a decision of 1 in the two examples, the second decision is more reliable. This is because if all nodes reporting a decision of 1 are on one side of n it is conceivable that these nodes are at the border of the event region. In this case, the node n is outside of the event region and no event should be detected. On the other hand, if nodes from different sides of n report a decision of 1, it is very likely that an event is also present at n. This is specially true in the case of a convex event region.

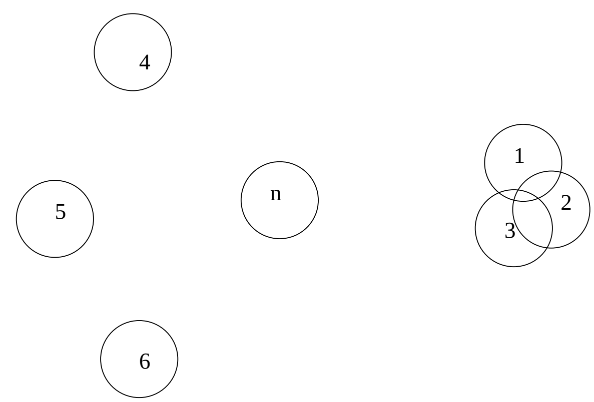

Fig. 1. Group-based error model

To take the geographical distribution into account, we define a new weighted voting model that gives a weight factor to each neighbor that is a function of the geographical distribution of the decision group to which they belong. There are two decision groups $G_0 = \{k \in FTN_n \setminus n : S_k(t) = 0\}$ and $G_1 = \{k \in FTN_n \setminus n : S_k(t) = 1\}$.

Definition 8. *We define the weight wg_j as the weight given to the decision group G_j with $j \in \{0,1\}$. This group weight is given by:*

$$wg_j = \frac{r - d(n, m_j)}{r} \tag{31}$$

where m_j is the geographical centroid of nodes in G_j, $d(n,m_j)$ is the distance between n and m_j and r is the fault-tolerance range.

This weight is higher when m_j is closer to n. In fact, the centroid of the group is closer to n when nodes are distributed around n than when the nodes are on the same side while having the same distances from n.

We can now compute the node weight that is a function of both its individual local detection error and its group weight. The new node weight of a neighbor $k \in G_j$ is given by:

$$wn_k = wg_j w_k \tag{32}$$

where w_k represents the original node weight, defined previously in theorem 2.

Using these new weights, we can now compute the probability of detection error when using a node $k \in FTN_n \setminus n$ to estimate the real situation at node n. This probability is given in the following proposition.

Proposition 2. *The probability of error when an event occurring at node n is detected by a node $k \in FTN_n$ is given by:*

$$p_k^n = \frac{1}{1 + e^{wn_k}} \tag{33}$$

Proof. This error probability is obtained by inverting the relationship in theorem 2 and using the new wn_k instead of w_k. □

We can now use these new probabilities of detection errors in the decision scheme instead of the original ones in definitions 5 and 6.

Note that this group-based error model can be used in combination with the distance-based one presented in the last section. In such a case, the detection error probability values are used to compute the original neighbor weights (w_k) to obtain new weights. These distance-based weights are then used to compute the node weight in this model.

V. Simulation Results

In this section, we present a set of simulation results that are intended to demonstrate some of the capability of our fault-tolerant estimator (FTEDD). In particular, the estimator is compared to the approach proposed in [8]. The simulations were conducted using the Georgia Tech Sensor Network Simulator (*GTSNetS*) [16], [12].

We simulated a sensor network of 1024 nodes randomly deployed (uniform distribution) in a region of 680 meters by 680 meters. The communication range was set to 23 meters. The parameter r defining the fault-tolerance range was set such that each interior node has 4 neighbors that are used in the distributed decision mechanism. One source (sensed object) was placed at the lower left corner of the region of interest. The sensing range was set to 93 meters. All simulation results were obtained by averaging over 1000 runs.

To reduce the size of the exchanged messages, we run an initial neighbors discovery phase prior to the execution of the fault-tolerance algorithm. This avoids having to send node locations along with every sensor reading message, which greatly reduces the sensor message size. This helps to reduce the energy cost of the algorithm, but requires the existence of an identification mechanism [17]. Every node communicates only with nodes in its fault-tolerance range. In this specific simulation, this range is less than the communication range. Nodes, therefore, broadcast their messages to the neighboring nodes and there is no need for any routing protocol.

In this simulation scenario, we are assuming a Gaussian error term. However, our decision scheme does not require this assumption. In fact, nodes are not required to know the specific distribution of the error term prior to the deployment or to have the same distribution. The error is assumed to have a mean of 0. We simulate the case of nodes having different error probability levels by assigning to a node k a random standard deviation σ_k. This standard deviation varies uniformly in a range of plus or minus a fixed percentage of the average standard deviation σ. This fixed percentage is referred to as the variation percentage in the rest of this section. The error level corresponding to the average σ is referred to as the nominal error probability in the rest of this section. The error probability p_k is then computed using the expression in equation 3. With $m_e = 32$ and $m_n = 0$. The tail function was approximated using the erfc (error complementary) function using the following expression.

$$Q(x) = \frac{1}{2} erfc(\frac{x}{\sqrt{2}}) \tag{34}$$

The following metrics, defined in [8] are used to compare our algorithm the algorithm proposed in [8] which uses a majority voting scheme.

- Number of errors corrected: number of original sensor errors detected and corrected by the algorithm
- Number of errors uncorrected: number of original sensor errors undetected and uncorrected by the algorithm
- Number of errors introduced by the solution: number of new errors introduced by the algorithm
- Reduction in errors: overall reduction in number of errors, taking into account the original errors and the ones introduced by the algorithm

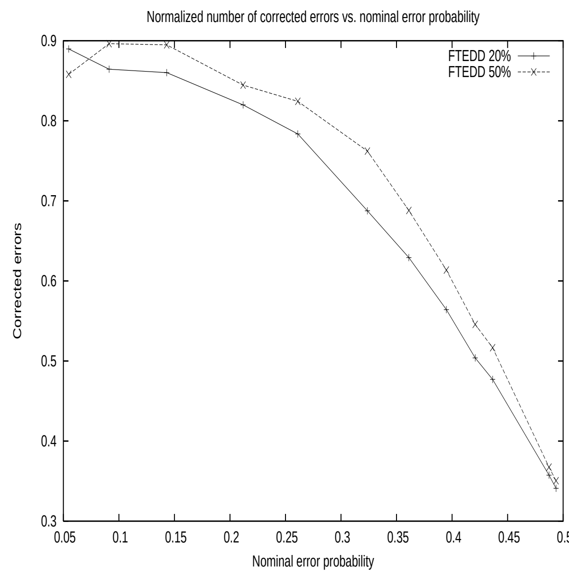

Fig. 2. Normalized number of corrected errors vs. nominal error probability for 20% and 50% percentage variations

The effects of two parameters on these metrics are studied. These parameters are the nominal local detection error probability levels and the variation percentage. In Figure 2, the normalized number of original errors detected and corrected using the FTEDD estimator is plotted as a function of the nominal error rate for two different values of the variation percentage. This graph shows that the estimator corrects a large percentage of errors caused by the inaccuracy of the local node decision scheme. The estimator corrects more than 85% of the local errors for an error level as high as 15%. The plot shows also that the FTEDD estimator maintains a high level of performance as the heterogeneity of the sensor nodes, represented by the variation in standard deviation, increases. In fact, the percentage of corrected errors remains relatively unchanged when the variation goes from 20% to 50%. The slightly better performance obtained for a percentage variation of 50% comes from the fact we vary the standard deviation σ and not the error probability itself. As the variation increases, the average error probability decreases due to the nature of erfc function. In fact, as this probability increases the variation in σ translates into less significant variation around the nominal probability of error. This is due to the decreasing rate of increase of p as a function of σ given in equation 34.

The normalized number of uncorrected errors can be readily computed from the normalized number of corrected errors, since the two sum up to 1.

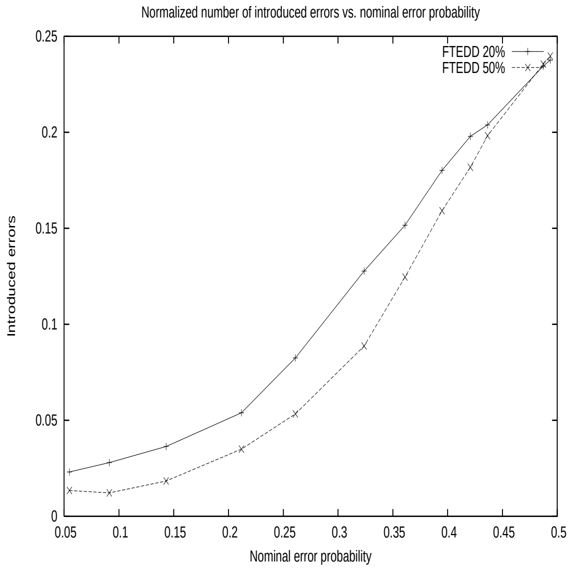

Fig. 3. Normalized number of introduced errors vs. nominal error probability for 20% and 50% percentage variations

Figure 3 gives the number of errors introduced by the use of the FTEDD estimator as a function of the nominal probability. This normalized percentage is computed as the number of introduced errors divided by the number of errors present when the estimator is not used. As the figure shows, the number of introduced errors remains relatively small, e.g., less than 5% for a nominal error probability of up to 15%. Again, the performance of the estimator remains very stable with respect to the level of variation. In particular, the normalized number of introduced errors does not change much between the cases 20% and 50% variation.

The normalized reduction in errors shows that the estimator reduces greatly the level of errors. At a nominal error level of 15%, the estimator reduces the average number of decision error by more than 80% as shown in Figure 4. Again, there is a slightly better performance when the percentage variation increases from 50% to 20%.

Our simulations demonstrate that the FTEDD estimator performs better than the majority voting scheme of [8]. The two estimators give similar results of the normalized number of corrected errors. However, FTEDD introduces fewer new errors than does the majority voting scheme as shown in Figure 5. The difference is even greater when we increase the level of variation around the nominal probability level. For a variation of 50%, for example, the number of errors introduced by the majority voting scheme is more than three times the number introduced by FTEDD for an error probability level of up to 15% as shown in Figure 6.

VI. Conclusion

We presented an optimal fault-tolerant estimator for distributed detection in sensor networks with sensor nodes of different accuracy levels. This estimator is proven to be equivalent to a weighted voting

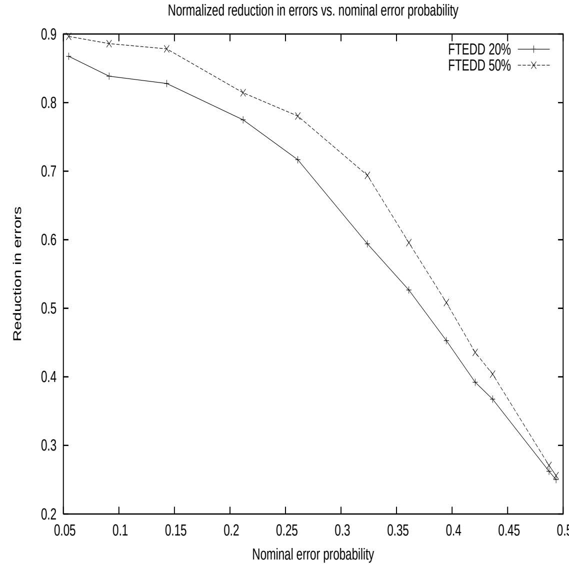

Fig. 4. Normalized reduction in errors vs. nominal error probability for 20% and 50% percentage variations

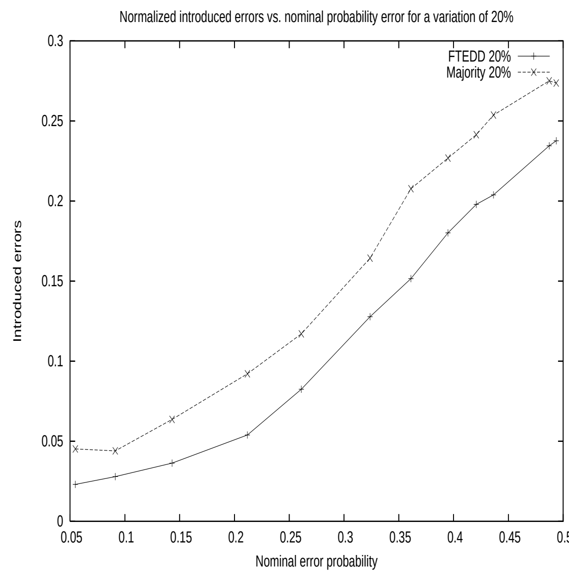

Fig. 5. Normalized introduced errors vs. nominal probability error for a variation of 20%

scheme. We also provided two new error models that account for the node distance and the geographical quorum distribution in the distributed detection decision scheme.

In addition to the theoretical analysis, the proposed fault-tolerance event detection scheme was tested and gave good performance under various simulation settings. It was found, for example, that this scheme can detect and correct more than 85% of original detection errors, while introducing only less than 5% of new errors.

Acknowledgment

This work is supported in part by NSF under contract numbers ANI-9977544, ANI-0136969, ANI-0240477, ECS-0225417, CCR 0209179.

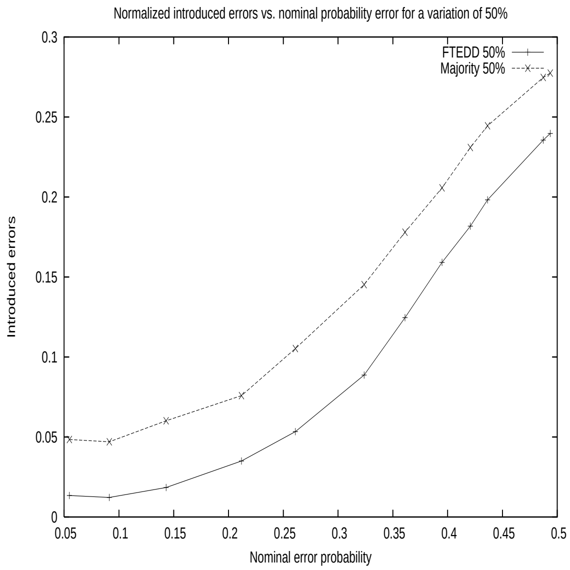

Fig. 6. Normalized introduced errors vs. nominal probability error for a variation of 50%

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 38 , Issue 4*, 2002.

[2] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors: Part i-fundamentals," *Proceedings of the IEEE, Vol.85, No.1*, 1997.

[3] R. Blum, S. Kassam, and H.V.Poor, "Distributed detection with multiple sensors: Part ii-advanced topics," *Proceedings of the IEEE*, January 1997.

[4] J. Chamberland and V. V. Veeravalli, "Distributed detection in sensor networks," *IEEE on Signal Processing, Vol.51, No.2*, 2003.

[5] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *Proceedings of European Workshop on Sensor Networks (EWSN'05)*, 2005.

[6] R. R. Brooks, C. Griffin, and D. Friedlander, "Self-organized distributed sensor network entity tracking," *International Journal of High Performance Computing Applications, Vol. 16, No. 3*, 2002.

[7] F. Koushanfar, M. Potkonjak, and A. Sanjiovanni-Vincentelli, "Fault tolerance techniques for wireless ad hoc sensor networks," in *Proc of IEEE Sensors*, 2002.

[8] 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*, March 2004.

[9] Q. Chen, K.-Y. Lam, and P. Fan, "Comments on "distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks"," *IEEE Transactions on Computers, Vol.54, No.9*, September 2005.

[10] X. Luo, M. Dong, and Y. Huang, "On distributed fault-tolerant detection in wireless sensor networks," *IEEE Transactions on Neural Networks, to appear*, 2005.

[11] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, 2003.

[12] E. Ould-Ahmed-Vall, B. S. Heck, and G. F. Riley, "Simulation of large-scale networked control systems using GTSNetS," in *Springer's Lecture Notes in Control and Information Sciences, Vol. 331, to appear*, P. J. Antsaklis and P. T. , Eds. Springer, 2006.

[13] P. K. Varshney, *Distributed Detection and Data Fusion*. Springer, 1996.

[14] D. K. Gifford, "Weighted voting for replicated data," in *Proceedings of the Seventh ACM Symposium on Operating Systems Principles*, 1979.

[15] G. Xing, C. Lu, R. Pless, and J. A. O'Sullivan, "Co-grid: An efficient coverage maintenance protocol for distributed sensor networks," in *Proceedings of the International Symposium on Information Processing in Sensor Networks (IPSN'04)*, 2004.

[16] E. Ould-Ahmed-Vall, G. F. Riley, B. S. Heck, and D. Reddy, "Simulation of large-scale sensor networks using GTSNetS," in *Proceedings of Eleventh International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'05)*, 2005.

[17] E. Ould-Ahmed-Vall, D. M. Blough, G. F. Riley, and B. S. Heck, "Distributed unique global id assignment for sensor networks," in *Proceedings of 2nd IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2005.