Fault-tolerant Event Boundary Detection in Wireless Sensor Networks

Kui Ren, Kai Zeng and Wenjing Lou
Department of ECE, Worcester Polytechnic Institute, MA 01609
{kren, kzeng, wjlou}@wpi.edu

***Abstract*— Event boundary detection is in and of itself a useful application in wireless sensor networks (WSNs). Typically, it includes the detection of a large-scale spatial phenomenon such as the transportation front line of a contamination or the diagnosis of network health. In this paper, we present FEBD, a fully distributed and light-weight Fault-tolerant Event Boundary Detection scheme. FEBD features an enhanced (nonparametric) statistical model that supports localized detection among neighboring nodes. To enhance detection accuracy, FEBD also introduces an error suppression technique prior to the determination of boundary nodes. The proposed scheme shows a much better detection accuracy and fault tolerance properties as compared to the previous models. The proposed FEBD is evaluated by extensive simulations, and presents very good detection accuracy, even when sensor fault probability is as high as 20%.**

I. Introduction

An important application of WSNs is to monitor, detect, and report the occurrences of events of interest [1], [2], [7], [8]. For some large-scale spatial phenomena, such as forest fire, environment temperature, and chemical spills, detecting the event boundary is sufficient or of more importance than collecting measurement over the entire event area. Due to the strict resource limitations (e.g., battery power, bandwidth) of sensor nodes and the nature of some events, it is not feasible to collect all sensor measurements and compute event boundaries in a centralized manner [3], [4]. A localized approach that allows in-network processing is therefore demanded. Sensor nodes are expected to collaborate with each other based on each own local view and provide a global picture for spatially distributed phenomena with greatly improved efficiency.

Recently, several localized boundary detection schemes have been proposed [5]–[10]. Nowak and Mitra [9] propose an edge estimation scheme when there exists a predefined hierarchical structure within the sensor network. Clouqueur, et. al. [5] seek algorithms to collaboratively detect the presence of a target in a region. Each sensor obtains the target energy (or local decision) from all other sensors in the region, drops extreme values if faulty sensors exist, computes the average, and then compares it with a pre-determined threshold for final decision. Krishnamachari and Iyengar [7] propose several localized threshold based decision schemes to detect both faulty sensors and event regions. The 0/1 decision predicates from the neighborhood[1] are collected and the number of neighbors with the same predicates are calculated. This number is used for the final decision based on a majority vote. The unique work that targets localized boundary detection in sensor networks is proposed by Chintalapudi and Govindan in [6]. All three schemes in [6] take as inputs the 0/1 decision predicates from neighboring sensors. The statistical approach computes the number of 0's and 1's in the neighborhood and a boundary sensor is detected if its neighbors contain a "similar" number of 0's and 1's. Here the "similarity" is defined based on a threshold whose value can be obtained based on a lookup table. Ding et. al. propose another localized statistical approach that takes as input not only 0/1 decision predicates but also numbers that abstract sensor readings or sensor behaviors [8]. Liao et. al. also propose a composite hypothesis test based approach for edge detection [11].

In this paper, we propose a Fault-tolerant Event Boundary Detection (FEBD) scheme, which detects the event boundaries in a localized manner with enhanced accuracy. In contrast to previous schemes, FEBD comes up with a novel error suppression technique. FEBD follows a decision fusion approach. Once an event of interest is detected, senor nodes first exchange their measurements (i.e., 0/1) among neighbors and each node suppresses its own (possibly) faulty measurements following a majority rule. A nonparametric statistical boundary detection model further operates on the error-adjusted measurement results, and identify boundary nodes. FEBD shows a much higher accuracy and better fault-tolerance as compared to previous schemes [6]–[8]. The performance of FEBD is justified by our extensive simulation study.

The remaining part of this paper is organized as follows. Section II describes the proposed FEBD. Section III then reports the simulation results on the performance of FEBD. Section IV concludes the paper.

II. FEBD: The Scheme

A. Network model

In FEBD, we adopt a refined event boundary definition based on those in [6], [8]. Consider a phenomenon (i.e., event $\mathcal{E}$) that spans some arbitrarily shaped sub-region of $\mathbb{S}$. Each sensor can, based on locally collected measurements, determine whether it belongs to the sub-region covered by the phenomenon or not. We assume that sensor nodes are uniformly distributed in the sensor field. Ideally, a boundary node, say S_i, is such a node that every closed disc centered at S_i contains both points in $\mathcal{E}$ and $\overline{\mathcal{E}}$ (that is, the boundary node should be right on the *real event boundary*, denoted as $\mathcal{B}_R$.), where $\mathcal{E}$ is the ground truth of the event covering sub-region

[1] We use "neighborhood" to indicate the one-hop neighbors of a node.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

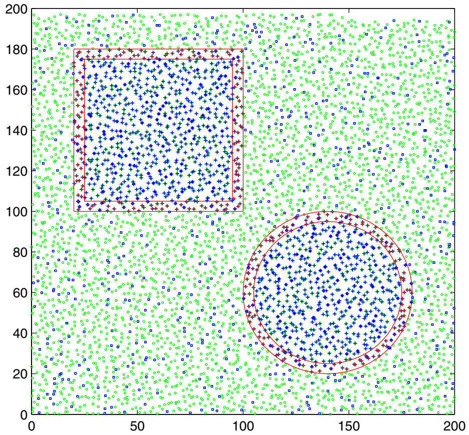

Fig. 1. An illustration of boundary definition $\mathbb{B}$ with $r = \frac{R}{2}$, where the event area is located inside the outer square and circle, and the boundary nodes are denoted in red '*', faulty nodes are denoted in blue '□' and normal nodes are denoted in green '○'.

in $\mathbb{S}$, and $\overline{\mathcal{E}}$ represents the remaining region, i.e., $\overline{\mathcal{E}} = \mathbb{S} - \mathcal{E}$. Hence, an event boundary, denoted as $\mathbb{B}$, when represented by sensor nodes, is simply a collection of such boundary nodes. However, due to the actual node density in practice, an event boundary found in this case constitutes only a very restrictive node set, which is far from enough to approximate/reveal $\mathcal{B}_R$ [6]. For this reason, the notion of *boundary width* r is introduced with its value $0 < r < R$ in FEBD, where R is the communication radius of sensor nodes. In FEBD, we define a sensor node, S_i, as a boundary node,

$$\mathbb{B} = \bigcup_i S_i,\ \forall\ i : |S_i \perp \mathcal{B}_R| \leq r, and\ S_i \in \mathcal{E},$$

where $|S_i \perp \mathcal{B}_R|$ denotes the distance between S_i and $\mathcal{B}_R$. This definition is illustrated in Fig. 1.

Naturally, the design goal of FEBD is then to identify as many nodes as possible in $\mathbb{B}$, in the presence of node random faults. (Here we assume that the sensor fault probability is symmetric and spatially uncorrelated.) At the same time, FEBD should include as few nodes as possible that do not belong to $\mathbb{B}$. We further require that the included bogus boundary nodes should be as close as possible to the real event boundary. Therefore, on top of *boundary width* r, the notion of *tolerance radius* is introduced to characterize the distribution of the boundary nodes detected by FEBD. In particular, any falsely detected boundary node that has its distance to real boundary $\mathbb{B}$ no more than $\frac{R-r}{2}$ is said to be within *tolerance radius*. We are more interested in the fraction of falsely detected boundary nodes that are far from the event boundary $\mathbb{B}$. An illustration of this definition is shown in Fig. 2.

B. The Scheme

The proposed FEBD is designed to be robust against node random faults, while keeping accuracy in mind. The key idea of FEBD is that, by allowing each sensor to collect event measurements from other nodes in the neighborhood, the boundary is present; by further requiring sensor nodes to adjust their measurements based on those of the neighborhood, the presented boundary can be more accurate and fault-tolerant. FEBD detects an event boundary in two steps. In Step 1) *local sensing and measurement adjusting*, each node exchanges its event measurement in the neighborhood. Then, every node adjusts its own measurement result according to *the majority rule*. Next, in 2) *distributed boundary detection* stage, each node independently determines whether or not it is a boundary node according to the predefined statistic model and the updated measurements distribution in its neighborhood.

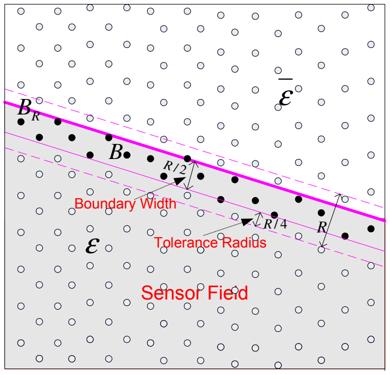

Fig. 2. An illustration of boundary width and tolerance radius with $r = R/4$

The observation behind FEBD is two-fold: 1) The original event measurements collected from neighbor nodes usually contains faulty measurements due to node measurement error and fault; if faulty measurements can be suppressed (in the statistical sense), boundary detection can certainly be more accurate and thus more tolerant against node random faults. 2) Statistically, a boundary node can be determined by comparing the event measurements among its neighbor nodes by assuming that the neighbor area of a sensor node is so "small" in comparison to the area covered by the entire event that the ground truth boundary can be approximated by a straight line in this area. In particular, a boundary node (i.e., belonging to $\mathbb{B}$) will always have the difference between the numbers of '0' and '1' measurements in its neighborhood limited by a certain threshold. The value of the threshold is determined by *boundary width* r given uniform node distribution.

In FEBD, the following specific rules are specifically designed to reflect the above observation:

- *Majority rule*: A node maintains its own measurement only when this result is the majority result within its neighborhood. Statistically, this rule could lead to error suppression, as long as sensor fault probability is less than 50%. To see this, we give an intuitive explanation as follows. We show that the adjusted results are at least no worse than the original ones. Assume that the WSN is deployed in a grid based manner, that is, each sensor node is placed with equal distance to each other, as in Fig. 2. In this case, the larger the area is, the more number the nodes reside in. Then, for every node $S_i \in \overline{\mathcal{E}}$ where $|S_i \perp \mathcal{B}_R| \geq \frac{R}{2}$, the majority rule obviously yields better (at least no worse) results, since sensor fault probability is less than 50% and we assume node fault probability is spatially uncorrelated. This is because the majority of the neighbors that S_i has will have the same measurement of '0'. The same argument also goes to the event nodes whose distances to $\mathcal{B}_R$ are no less than $\frac{R}{2}$. The difference is that now the majority of the neighbors that S_i has will have the same measurement of '1'.

Next, we consider an event node S_i nearby the event boundary, i.e., $|S_i \perp \mathcal{B}_R| < \frac{R}{2}$. Still, the number of event nodes in the neighborhood of S_i is obviously larger than that of non-event nodes, assuming sensor fault probability is 0%. Then, given a non-zero sensor fault probability, the situation is still the same in the statistic sense, since the sensor fault probability is symmetric[2]. The same argument also goes to the non-event nodes nearby the boundary, because of the same reason. Hence, the majority rule always leads to at least no worse results as compared to original ones in case of the grid-based deployment. This conclusion can be further extended to the case of node uniform distribution, since a larger area still statistically contains a larger number of nodes. Note that the *majority rule* has been proved to be optimal based on an Bayesian fault recognition model in suppressing node random measurement faults [10]. We refer the interested reader to [10] for the detailed proof.

- *Determination rule*: A node recognizes itself as a boundary node only when

$$1 - \frac{n_+ - n_-}{n_u} \geq \gamma, \quad (1)$$

where n_+ is the number of '1' measurements in a node u's neighborhood, n_- is the number of '0's, and $n_u = n_+ + n_-$ is the actual neighborhood size. Furthermore, γ is a preset system parameter, called *normalized acceptance threshold*. The performance of the scheme relies on an appropriate value of γ. In FEBD, the optimal choice of γ is not related to the sensor fault probability, which is in contrast to the previous schemes. In FEBD, we set

$$\gamma = 1 - \frac{II(r) - I(r)}{\pi R^2}, \quad (2)$$

where the areas of II and I are illustrated in Fig. 3. This selection of γ is obviously optimal in case of grid-based node deployment, since the area size directly determines the number of nodes that are located inside the area. In case of uniform node distribution, it is still the case though in the statistic sense.

To the best of our knowledge, this is the first statistic model that combines faulty data adjustment in event boundary detection.

III. Simulation Studies of FEBD

A. Performance Evaluation Metrics

The following three metrics are used to evaluate the performance of FEBD. Let $\mathbb{B}'$ be the set of boundary nodes detected by FEBD. Let $\mathbb{B}$ be the set of actual boundary nodes as defined in Section II.A. First of all, we want FEBD to detect as many real boundary nodes (i.e., nodes belonging to $\mathbb{B}$) as possible. Hence, *hit rate* is evaluated.

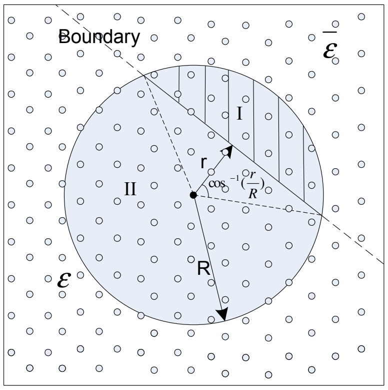

Fig. 3. An illustration of areas I and II

Hit Rate e_f: e_f represents the fraction of sensors in $\mathbb{B}$ that are correctly detected by FEBD:

$$e_f = \frac{\#\{\mathbb{B} \cap \mathbb{B}'\}}{\#\{\mathbb{B}\}} \quad (3)$$

False Detection Rate e_d: e_d represents the fraction of falsely detected sensors[3] with respect to the size of $\mathbb{B}$. Here, only those falsely detected sensors whose distances to the boundary are at least $\frac{R-r}{2}$ are counted. Let $\mathbb{A}$ denote the set of falsely detected nodes whose distances to the boundary are larger than $\frac{R-r}{2}$ (cf. Fig. 2).

$$e_d = \frac{\#\{\mathbb{A}\}}{\#\{\mathbb{B}\}} \quad (4)$$

Further, we denote the mean distance of the nodes in $\mathbb{B}'$ to $\mathbb{B}_R$ as $d_{\mathbb{B}'}$, and evaluate *normalized mean distance* of the boundary nodes detected by FEBD.

Normalized Mean Distance e_w: e_w represents the normalized mean distance of $\mathbb{B}'$ with respect to boundary width:

$$e_w = \frac{d_{\mathbb{B}'}}{r} \quad (5)$$

B. Simulation setup

In all simulations, sensor nodes are located in a 200m by 200m area, their locations drawn from a uniform distribution over the area. The radio range of all the sensors is 10m and assumed omni-directional. In all simulations, we arbitrarily choose the *boundary width* $r = R/2$. And γ is chosen to be $1 - \frac{II(r=\frac{R}{2}) - I(r=\frac{R}{2})}{\pi R^2} = \frac{2\pi - 3\sqrt{3}}{3\pi} \approx 0.12$. Note that our γ value is independent of sensor fault probability.

We report the results for event regions with ellipses or straight lines as the boundaries. Straight lines are selected because when the network area is large, the view of the boundary of one sensor near the boundary is approximated by a line segment in most cases. An ellipse represents a curly boundary.

[2] Since the sensor fault probability is symmetric, a sensor node will have equal probability in giving a '1' when the truth is '0', and giving a '0' when the truth is '1'.

[3] Here the false detection only counts the false positives, i.e., sensors that are not boundary nodes but are erroneously detected as boundary nodes.

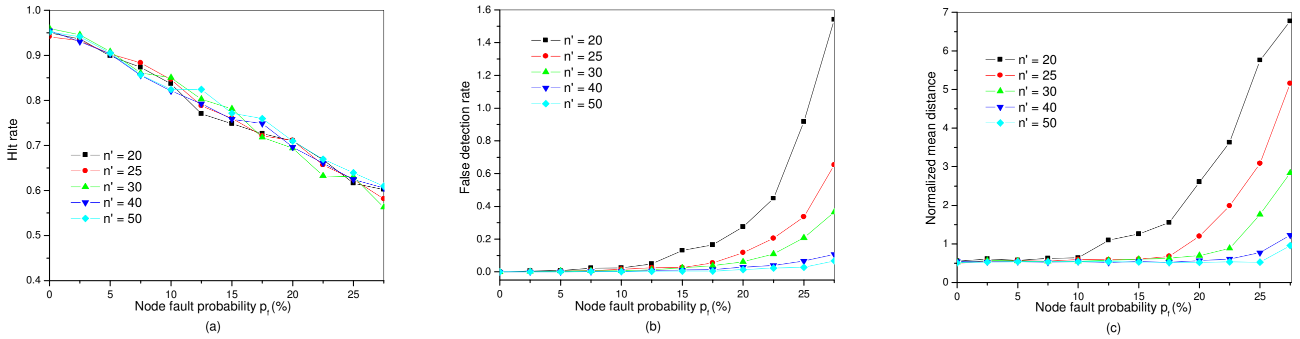

Fig. 4. Simulation results with respect to three evaluation metrics

Our simulation produces similar results for event regions with other boundary shapes. The event regions are generated as follows. For a linear boundary, a line $y = kx + b$ is computed, where $k \tan\theta$ is the slope, with θ drawn randomly from $(0, \frac{\pi}{2})$, and b is the intercept, drawn randomly over the entire x-axis within the sensor field. The area below the line is the event region. For a curly boundary, the event region is within an ellipse that can be represented by $E(a, b, x_0, y_0, \theta) = 0$ [6]. Here $2a$ and $2b$ are the lengths of the major and minor axes of the ellipse, with a, b drawn randomly over the length of the sensor field. (x_0, y_0) is the center of the ellipse drawn uniformly over the entire sensor field. θ, the angle between the major axis of the ellipse and the x-axis, is a random number between 0 and π.

To examine the impact of density, we chose five values of node density (i.e., neighborhood size): 20, 25, 30, 40 and 50. To capture the impact of node random faults, we used a simple bit flipping technique. In this model, a sensor toggles its event predicate value from its true value with probabilities p_f. We used twelve different choices for p_f: 0%, 2.5%, 5%, 7.5%, 10%, 12.5%, 15%, 17.5%, 20%, 22.5%, 25%, and 27.5%. Thus, a single simulation run represents one line/ellipse chosen randomly, for one value of density and sensor fault probability. For each given density and sensor fault probability, the results for the three performance metrics are averaged over 50 simulation runs corresponding to 50 randomly chosen lines.

C. Simulation results

In this subsection, the simulation results are reported in details. The three performance evaluation metrics defined in section V.A with regard to network density and sensor fault probability are reported in Fig. 4 (a), (b) and (c), respectively. In contrast to the previous schemes, we do not change any setting on parameters as sensor fault probability increases from 0% to 27.5%. That is, our simulation results does not rely on the pre-knowledge of sensor fault probability, which, in fact, may not be available as a priori in many practical applications.

Firstly, we observe that the proposed FEBD performs very well, when sensor fault probability equals to zero. In this case, the hit rate e_f is always as high as 95%, no matter what the network density n' is. In the previous schemes [6], [8], e_f is generally no more than 85%. Hence, FEBD has the highest hit rate in the ideal situation as compared to the previous schemes.

Secondly, Fig. 4 (a) shows that 1) the hit rate e_f in FEBD does not rely on network density; this is because we intentionally used normalized threshold value in boundary node detection process. 2) FEBD is very good at detecting boundary nodes: e_f remains to be larger than 55%, when sensor fault probability p_f reaches as high as 27.5%. This result significantly outperforms any of the previous schemes [6]–[8].

Thirdly, FEBD presents a high security strength as shown in Fig. 4 (b). When n' is as low as 20, the false detection rate e_d is still less than 5%, given $p_f = 12.5\%$. And for the same p_f, e_d can be kept as low as 0.3% when $n' = 50$. Further, given $n' = 50$, e_d increases very slowly as p_f increases; e_d equals to only 0.67%, when p_f reaches to 27.5%.

Fourthly, Fig. 4 (c) shows that the detected boundary nodes by FEBD are very close to the defined boundary $\mathbb{B}$. It is shown that as long as $n' \geq 25$, the normalized mean distances of the detected boundary nodes always kept to be around the ideal value 0.5, given $p_f \leq 17.5\%$.

In summary, the simulation results shown in Fig. 4 indicate that 1) FEBD performs well until p_f is up to 12.5%, even when n' is as low as 20; 2) FEBD keeps presenting a very good performance and security strength even when p_f goes up as high as 25%, given a reasonable high n'; 3) FEBD significantly outperforms the previous schemes in all the three metrics [6], [8].

Fig. 5 gives several visualized results to illustrate the performance of FEBD, where the detected boundary nodes are denoted in red '*', faulty nodes are denoted in blue '□', and the normal nodes are omitted in the figure. The left figure gives the performance of FEBD at low sensor fault probability. Clearly, when $p_f 7.5\%$, FEBD has a very high hit rate: $e_f = 85\%$. The middle figure gives the performance of FEBD at medium sensor fault probability: $e_f = 79\%$, when $p_f = 15\%$. Obviously, the detected event boundaries in both left and right figure are very good approximations of the real event boundaries as defined in Fig. 1. In the right figure, we can find that as sensor fault probability continues to be higher, the detected boundary presents a larger false detection rate as compared to the previous ones. But still, we have $e_f = 60\%$, given p_f as high as 27.5%.

One can easily see that the performance of FEBD improves as we require each node to collect more event measurements from more sensor nodes in its neighborhood. This is because

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

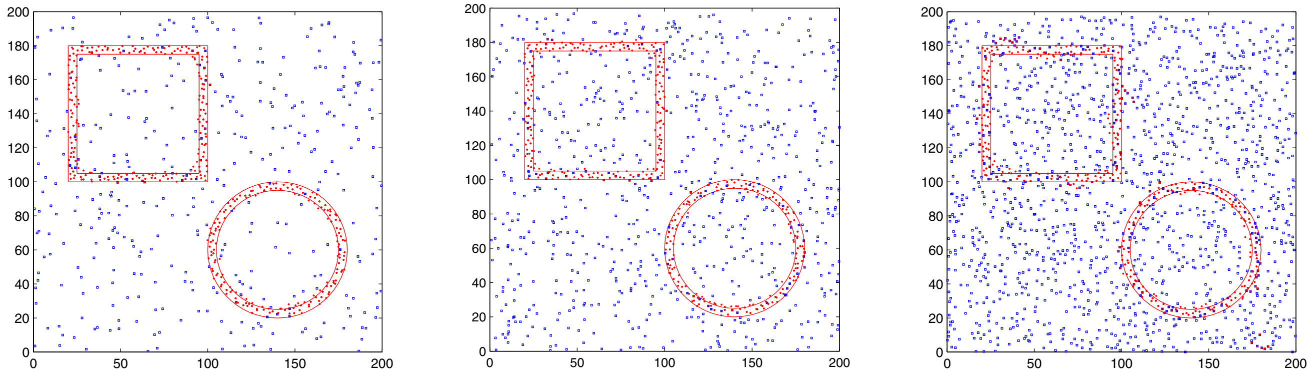

Fig. 5. Simulation results: left): $e_f = 85\%$ with $p_f = 7.5\%$; middle) $e_f = 79\%$ with $p_f = 15\%$; right) $e_f = 60\%$ with $p_f = 27.5\%$.

each node can get more samples from both the interior and exterior of the event, and makes more accurate estimate in the presence of node random faults. However, collecting more measurements from more nodes other than the immediate neighbors incurs much higher communication overhead. As mentioned in [6], communication overhead increases roughly quadratically as the neighbor range increases. This will result in a much higher energy consumption. In FEBD, we assume that the underlying network is well connected, that is, neighborhood size is reasonably large to support fine grained collaborative event detection. Hence, a good energy-accuracy tradeoff is achieved by letting each node collect the measurements from their immediate neighbors only. As we have shown in Fig. 4 (a), e_f is larger than 80% with n' as low as 20, given p_f up to 12.5%.

D. *Communication overhead*

One can easily see that the performance of FEBD improves as we require each node to collect more event measurements from more sensor nodes in its neighborhood. This is because each node can get more samples from both the interior and exterior of the event, and makes more accurate estimate in the presence of node random faults. However, collecting more measurements from more nodes than the immediate neighbors incurs much higher communication overhead. As mentioned in [6], communication overhead increases roughly quadratically as the neighbor range increases. This will result in a much higher energy consumption. In FEBD, we assume that the underlying network is well connected, that is, neighborhood size is reasonably large to support fine grained collaborative event detection. Hence, a good energy-accuracy tradeoff is achieved by letting each node collect the measurements from their immediate neighbors only. As we have shown in Fig. 4 (a), e_f is larger than 80% with n' as low as 20, given p_c up to 10%.

IV. Concluding Remarks

In this paper, we have studied a special instance of fault-tolerance collaborative in-network processing tasks in WSNs, i.e., distributed event boundary detection. We presented our Fault-tolerant Event Boundary Detection (FEBD) scheme. Along with FEBD, we also proposed an enhanced nonparametric statistic model for localized event boundary detection, which allows faulty measurements adjustment within the neighborhood, and obtains the decision based on the adjusted measurements distribution. Our model is much more accurate and robust against node faults as compared to existing schemes. Moreover, our model is nonparametric and does not rely on any a priori knowledge of sensor fault probability, which, on the other hand, are assumed by existing schemes to get the optimal results. We also used extensive simulations to evaluate FEBD, and shows a very good performance and security strength of FEBD, even when sensor fault probability achieves as high as 20%.

References

[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, “Wireless Sensor Networks for Habitat Monitoring,” ACM WSNA02, Atlanta GA, September 2002.

[2] N. Xu, A Survey of Sensor Network Applications, http://enl.usc.edu/ ningxu/papers/survey.pdf

[3] J. Hill, R. Szewczyk, A. Woo, S. Hollar, and J. Heidemann, “System architecture directions for networked sensors,” In Proc. 9th International Conference on Architectural Support for Programming Languages and Operating Systems, Nov. 2000.

[4] S. Madden, M. J. Franklin and J.M. Hellerstein and W. Hong, “TAG: a tiny aggregation service for ad-hoc sensor networks,” OSDI, Dec. 2002.

[5] T. Clouqueur, K.K. Saluja, and P. Ramanathan, “Fault Tolerance in Collaborative Sensor Networks for Target Detection,” IEEE Transactions on Computers, pp. 320-333, Vol. 53, No. 3, March 2004.

[6] K. Chintalapudi and R. Govindan, “Localized Edge Detection in Sensor Fields,” IEEE Ad Hoc Networks Journal, pp. 59-70, 2003.

[7] 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.

[8] M. Ding, D. Chen, K. Xing, and X. Cheng, “Localized Fault-Tolerant Event Boundary Detection in Sensor Networks,” IEEE INFOCOM 2005, 13-17 March 2005.

[9] R. Nowak and U. Mitra, “Boundary Estimation in Sensor Networks: Theory and Methods,” In Proc. of IPSN 2003, LNCS 2634, pp.80-95, 2003.

[10] B. Krishnamachari1 and S. Iyengar, “Efficient and Fault-tolerant Feature Extraction in Wireless Sensor Networks,” In Proc. of IPSN 2003, LNCS 2634, pp.488-501, 2003.

[11] P. Liao, M. Chang and C. Kuo, “Distributed edge detection with composite hypothesis test in wireless sensor networks,” in Proc. IEEE Global Telecommunication Conference (Globecom) 2004.