A New Final Decision Scheme with On-Line Sensor Fault Detection in Wireless Sensor Networks

Leu-Chin Chen[1], Jeng-Yang Wu[1], Dyi-Rong Duh[1,*], and Tsang-Yi Wang[2]

[1]*Department of Computer Science and Information Engineering*
National Chi Nan University, Puli, Nantou Hsieh 54561, Taiwan
`{s96321527, s94321522, drduh}@ncnu.edu.tw`
[2]*Institute of Communications Engineering*
National Sun Yat-Sen University, Kaohsiung 80424, Taiwan
`tcwang@mail.nsysu.edu.tw`

***Abstract*—A wireless sensor network (WSN) are composed of a lot of number of sensor nodes and usually used to monitor a region of interest. The distributed network musts collect all local decisions to make a decision to decide whether an event actually happened in accuracy. However, the noise in the environment may affect the result of a final decision and degrade the performance of the fusion center. Additionally, faulty nodes may also degrade the outcomes of a final decision. This work addresses a final decision scheme with on-line fault detection in a WSN where the fusion center becomes more event-sensitive and has a lower final decision error rate. Because of the computing capability constraint in a WSN, a low-complexity final decision scheme, which adopts a simple queue, is proposed. The proposed scheme can detect whether the event truly changed in low delay time. The simulation results show that the proposed scheme is more effective to detect events in WSN.**

***Keywords*— Wireless sensor networks, decision fusion, sensor fault detection, final decision.**

1. INTRODUCTION

A wireless sensor network (WSN) usually consists of a lot of sensor nodes deployed into the harsh or unapproachable environment for collecting the interested information such as debris flow and landslide. Therefore, WSN has received much attention recently because of many important applications [1–9].

However, the sensor nodes are very prone to damage because of the low-cost design and random deployment. Therefore, the failure sensor node may reduce the application performance. Wu et al. modeled the types of sensor faults in WSN ranging from simple stuck-at faults to random sensor faults and present an algorithm based on a record table for on-line sensor fault detection [10–12]. Wu et al.'s scheme ignores the faulty nodes to make a more believable final decision. However, the final decision is not sensitive while the environment condition has not been changed for a long time. Because the final decision is decided by the summation of ratios of local decision '1' of all normal sensor nodes, which were summed up from the beginning time or event changing time to current time, the ratios summation is very hard to reach the threshold such that the event happening or event changing cannot be detect in short time.

For an example, four sensor nodes are deployed into an interested environment, which has a serious noise to affect the local decisions of sensor nodes, to detect whether the event is happening as shown in TABLE 1, where N is the number of nodes and s_i represents the i^{th} node.

TABLE 1
DETAIL OF THE RECORD TABLE WHEN N = 4

Time	***t*=1**	***t*=2**	**…**	***t*=200**	***t*=201**	**…**	***t*=396**	***t*=397**
Event	*E*=0	*E*=0	…	*E*=0	*E*=1	…	*E*=1	*E*=1
s_1	0	0	…	0.245	0.245	…	0.497	0.498
S_2	1	0.5	…	0.64	0.645	…	0.740	0.740
S_3	0	0	…	0.3	0.305	…	0.510	0.511
S_4	0	0	…	0.26	0.265	…	0.491	0.492
Final	*F*=0	*F*=0	…	*F*=0	*F*=0	…	*F*=0	*F*=1

Because of the noise, each normal sensor node has the same probability to make a wrong decision while the environment condition is event or non-event. Assume that the environment condition is truly non-event in time interval $t = 1$ to $t = 200$, and the environment begins event

---

[*] *Correspondence to:* D.-R. Duh; e-mail: drduh@ncnu.edu.tw

happening while $t = 201$. Also, assume the threshold of ratios summation T_r is 0.5 per sensor node. At $t = 200$, the local decision 1 ratios of s_1, s_2, s_3 and s_4 are 0.245, 0.64, 0.3 and 0.26, individually. According to the on-line fault detection scheme, local decision ratio of s_2 is ignored by the fusion center and the summation of ratios is 0.805. Obviously, the summation of ratios is smaller than 1.5, which is $T_r \times 3$, so the final decision F is 0, non-event. Until $t = 397$, the summation of ratios is 1.51 which is greater than $T_r \times 3$, the final decision F becomes 1 to represent event happening.

Consider that event is truly happening at $t = 201$, but the fusion center detects the event at $t = 397$. The final decision scheme takes a long time to detect the event. In other words, the event change cannot be detected as soon as possible. Because the original final decision scheme is not enough sensitive to event change, this work propose a new final decision scheme for improving the event-sensitivity of previous detection schemes.

The rest of this paper is organized as follows. Section 2 describes a system model and defines the problem of our work. Section 3 demonstrates our final decision scheme to mitigate the event changing detection delay, which is the primary goal of this study. Section 4 presents the Monte Carlo simulation results. Finally, conclusions are drawn in Section 5.

2. SYSTEM MODEL

This section introduces the system model of this work. First, the network operation is illustrated. Second, local decision rule employed at each node is then given. Finally, three kinds of sensor fault types in sensor network are presented.

2.1. Network Operation

As illustrated in Fig. 1, a two-layer detection system is considered in this work. This system consists of a fusion center and N identical sensor nodes. The fusion center is used to decide whether the unknown condition of environment is event happening. Assume that the environment condition is a binary hypothesis, H_0 or H_1. The hypothesis H_0 (H_1) indicates the phenomenon is normal (event happening). Each member of N sensor nodes is denoted by s_i, where $i = 1, \ldots, N$. Let x_i^t denote the observation of the ith sensor node and u_i^t denote the binary decision of the ith sensor node, and t is the time index.

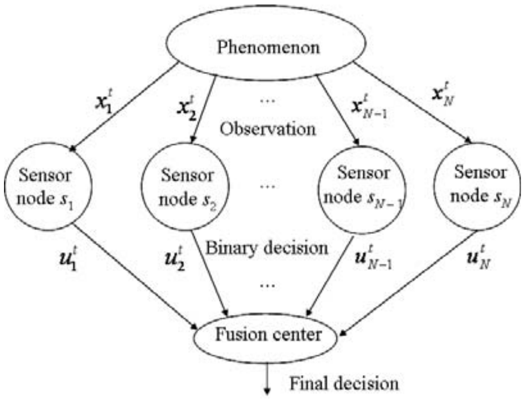

Fig. 1 System model of a parallel fusion network

Each sensor node observes the phenomenon independently and gets the x_i^t at each time slice. All sensor nodes make local decisions in binary through the local decision rule γ, which is employed at each sensor node, as (1)

$$u_i^t = \gamma(x_i^t). \tag{1}$$

Every node sends its binary decision to the fusion center after making a local decision, and then fusion center make a final decision for this time step t. A decision '0' is sent if the sensor node makes a decision in favor of H_0; otherwise, a decision '1' is transmitted.

Moreover, the Gaussian noise is considered in this study. The noise in the phenomenon may affect the local decisions of sensor nodes to make a wrong final decision. In other words, the final decision is still possible to make a wrong decision even if there is no faulty sensor node.

Let's consider the fusion center is processing its information at time step t. All preliminary decisions up to time step t from all nodes are available at the fusion center. The fusion center begins to identify faulty members by Wu et al.'s fault detection scheme [10–12]. In the fusion process, the fusion center isolates the data of faulty nodes and finally makes the believable decisions by adopting our proposed final decision scheme as described in Section 3.

2.2. Sensor Fault Type

A sensor node is very prone to failure, because of its low-cost design. The faulty nodes may transmit the unreliable information to affect the final decision in the monitored environment. There are three kinds of sensor fault considered in this work: stuck-at-one, suck-at-zero and random fault.

These three kinds of fault are without respect to the real condition of the environment. Stuck-at-one (stuck-at-zero) fault means the failure sensor node always transmits ‘1’ (‘0’) to the fusion center and random fault means the local decision is decide ‘0’ or ‘1’ randomly.

3. FINAL DECISION SCHEME WITH FAULT DETECTION

This section introduces a new final decision scheme, which uses a queue to keep the results of history of majority vote. The queue plays an important role for a final decision. The more details of the proposed final decision scheme are described in the following.

3.1. Queue

This work considers that every sensor node sequentially transmit a local decision to fusion center at each time interval, then the fusion center decides a final decision in terms of these local decisions. Because of the noise, the local decisions have probability of miss which is denoted as P_M^L and probability of false alarm which is denoted as P_F^L. Obviously, a normal sensor node has the probability of detection (normal) which the event is truly happening and detected (which the environment condition is normal and no any alarms), $P_D^L = 1 - P_M^L$ ($P_N^L = 1 - P_F^L$) when H_1 (H_0).

Since the P_F^L or P_M^L may affect the final decision if only using the current local decisions to decide a final decision, the current final decision must also refer to the history final decisions. From this viewpoint, a queue is designed to keep the history “pre-decisions”, which are decided by majority vote of local decisions, and can decrease the probability of false alarm in non-event or probability of miss in event really happening.

At each time, the fusion center decides a pre-decision by majority vote of local decisions, such as ‘0’ or ‘1’. Additionally, the pre-decision is put into the queue, which is set up in the fusion center as shown in Fig. 2.

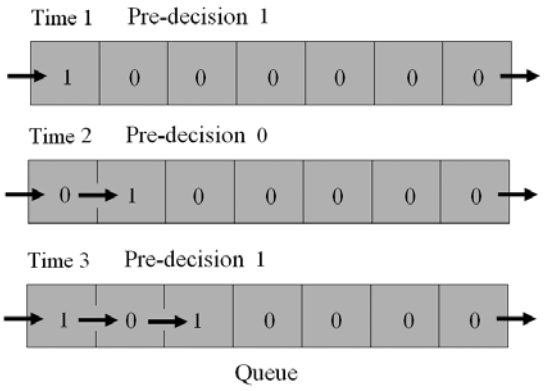

Fig. 2 Pre-decision is put into the queue

In the following, the fusion center checks out the summation of values in the queue whether it is greater than or equals the high threshold value. If the summation of values in the queue is greater than or equals the high threshold value, the final decision is ‘1’. If the summation of values in the queue is smaller than or equals the low threshold value, the final decision is ‘0’. Let the summation of values in queue at time step t be denoted by Q_{sum}^t as

$$Q_{sum}^t = \sum_{i=1}^{L_Q} Q^t[i]. \tag{2}$$

In (2), $Q^t[i]$ is the i^{th} element of the queue while time step t, and L_Q denotes the length of queue. For example, if the fusion center has the length of queue by 7 and the conditions of queue are 1, 0, 0, 1, 0, 0 and 0 at time step $t = 20$, so the summation of values in queue Q_{sum}^{20} is 2.

3.2. Proposed Scheme

The fusion center gets the summation of values at every time interval and checks out the summation whether greater than or equals the critical value, which is given by the length of queue L_Q and the critical threshold of queue T_Q. Let $D_F(t)$ denote the final decision at time step t, and the final decision function is defined as (3). Significantly, $D_F(0)$ is defined to be 0.

$$D_F(t) = \begin{cases} 1 & \text{if } D_F(t-1)=0 \text{ and } Q_{sum}^t \geq \lfloor T_Q \times L_Q \rfloor \\ 0 & \text{if } D_F(t-1)=1 \text{ and } Q_{sum}^t \leq \lceil (1-T_Q) \times L_Q \rceil \\ D_F(t-1) & \text{otherwise} \end{cases} \tag{3}$$

As shown in (3), the fusion center decides that the environment condition is event happening if $D_F(t-1) = 0$ and Q_{sum}^t is greater than or equals $\lfloor T_Q \times L_Q \rfloor$; the environment condition is non-event if $D_F(t-1) = 1$ and Q_{sum}^t is smaller than or equals

$\lceil(1-T_Q)\times L_Q\rceil$, and the event is non-changed. In addition, two critical thresholds $\lfloor T_Q\times L_Q\rfloor$ and $\lceil(1-T_Q)\times L_Q\rceil$ have the hysteresis phenomenon to mitigate the probability of miss or false alarm when the environment statement has not been changed for a long time. Notably, the hysteresis phenomenon let the WSN system become more stable and reliable.

At time step $t = 1$, the fusion center is initialized by setting all queue elements and final decision '0' in the first, since the event usually does not happen in the beginning. Then all sensor nodes transmit their local decisions to the fusion center, and the fault detection scheme using a record table is applied [12]. The faulty nodes are identified by the fusion center and ignored their local decisions to make a majority vote. Based on the majority vote, the fusion center can get a pre-decision at each time interval and puts the pre-decision into the queue. Then the fusion center checks the summation of values in queue to get a final decision and to find out whether the event is changed. If the fusion center decides the event change, the entire queue values are reset to all '0' or '1', which are depend on the new outcome of final decision. Because the resetting can reduce the affection of opposite pre-decision factor, the probability of bounce in final decisions becomes lower. Finally, the flow chart of the proposed final decision scheme is illustrated in Fig. 3.

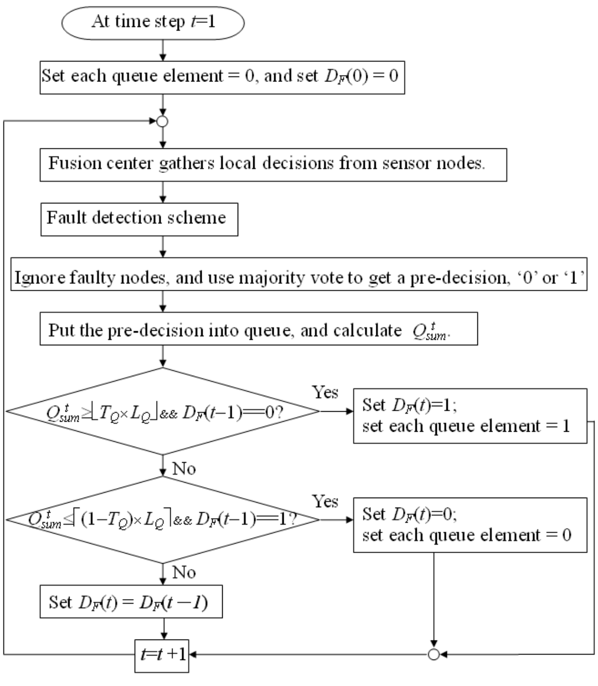

Fig. 3 The flow chart of the proposed scheme

4. SIMULATION RESULTS

The error number which is used to evaluate the performance of the proposed scheme in final decision is first described. Finally, the simulation setup is introduced and the experimental results are shown.

4.1. Error Number of Final Decision

This section defines the error number of a final decision by the proposed scheme comparing to the real condition. For example, the proposed approach sets up a counter to count the error number of the final decision. In the beginning, the counter is set to be zero. If the final decision is different from the real condition at a time step, the counter is added by one. Restated, whenever the decision made by the fusion center does not match the real condition of environment, an error occurs. An example of the error number of the finale decision is given in TABLE 2.

TABLE 2
EXAMPLE OF ERROR NUMBER IN FINAL DECISION

	t_1	t_2	t_3	t_4	t_5	t_6	Error Number
Real Condition	0	0	0	0	1	1	2
Final Decision	0	1	0	0	0	1	

4.2. Simulation Setup

The detection of known signals in Gaussian noise is considered. According to the signal-to-noise ratio (SNR) at 0dB, the value of P^L_D is 0.691462461, and the values of P^L_M and P^L_F are 0.308537539 respectively. Assume the number of deployed sensor nodes N is set to 10. Since a longer queue makes the event changed detection delay much longer and the shorter queue makes the probability of miss or probability of false alarm larger, the queue length L_Q is set to 20 in all simulated conditions. However, different environment conditions have different optimal combinations between L_Q and T_Q. How to get the optimum is our future work and not discussed in this study.

Each simulated scenario is iterated 10^5 times to obtain the simulated performance and only observed initial 300 time intervals to get the average of every error number. Let P_e represent the probability of the event. Significantly, two different probabilities often make two different outcomes of the error number. In this simulation, P_e is set to 0 and 0.002. To use these values is according to the low event probability can get much better performance in this work. The probability of event means the probability of event happening in one time step. Furthermore, assume that the length of an event is at least the length of the queue to ensure the event can be captured by the proposed scheme; the maximum last time of an event is four times of the queue length to ensure the event can be alternated. The numbers of faulty nodes N_F are 0 or 2, and the faulty type is shown stuck-at-one and random.

4.3. Results and Analysis

Fig. 4 shows the error number of the final decision when $P_e = 0$ and $N_F = 0$. Obviously, the error number of the final decision is the number of false alarm. Notably, the higher threshold can make lower error rate of false alarm. Also, the final decision scheme becomes more reliable but takes a longer time to detect an event change. Interestedly, the proposed scheme is better than Wu et al.'s scheme even $P_e = 0$ and $N_F = 0$ when $T_Q \geq 0.6$.

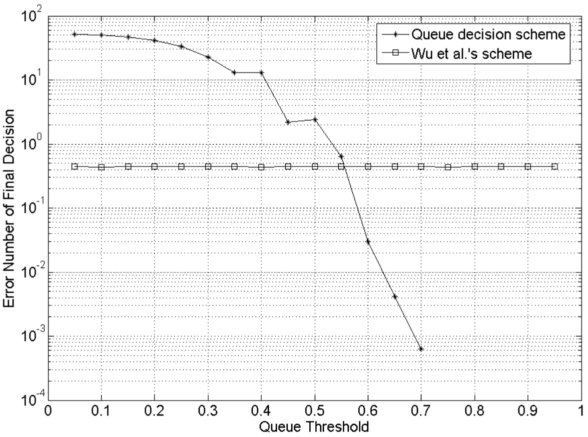

Fig. 4 Error number when $P_e = 0$ and $N_F = 0$

Fig. 5 indicates the error number of the final decision when $P_e = 0.002$ and $N_F = 0$. Because P_e is increased, the event changing probability is also increased. While the environment condition has not been changed for a long time and is changed suddenly, Wu et al.'s scheme cannot detect the event change immediately. In other words, the summation of ratios of local decision 1 is hard the get the threshold, which is used to decide the event happening, to change the final decision of Wu et al.'s scheme. Obviously, the proposed scheme is more sensitive to detect the event change.

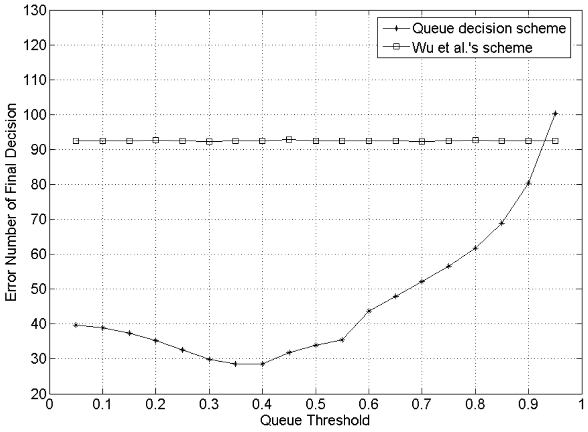

Fig. 5 Error number when $P_e = 0.002$ and $N_F = 0$

Fig. 6 shows the error number of final decision when $P_e = 0$, $N_F = 2$, and stuck-at-one. Obviously, the error number of final decision is higher than $N_F = 0$, because the faulty nodes can affect the final decision outcome, and the fault detection scheme cannot find out the faulty in the initial time intervals. However, the proposed scheme is better than Wu et al.'s scheme when $T_Q \geq 0.6$ and $L_Q = 20$.

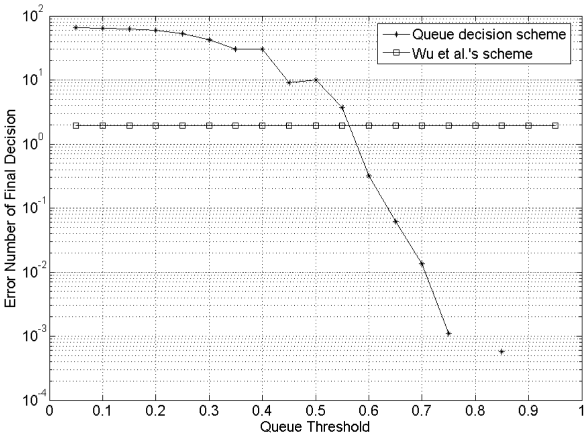

Fig. 6 Error number when $P_e = 0$, $N_F = 2$, and stuck-at-one fault

Fig. 7 illustrates the error number of the final decision when $P_e = 0.002$ and $N_F = 2$. Because faulty nodes make the unreliable local decisions no matter in the proposed scheme or Wu et al.'s scheme, the final decision may be affected for several time steps. Therefore, the error number

$N_F = 2$ is higher than $N_F = 0$ in general. Obviously, the proposed scheme is better than Wu et al.'s scheme when $T_Q < 0.9$.

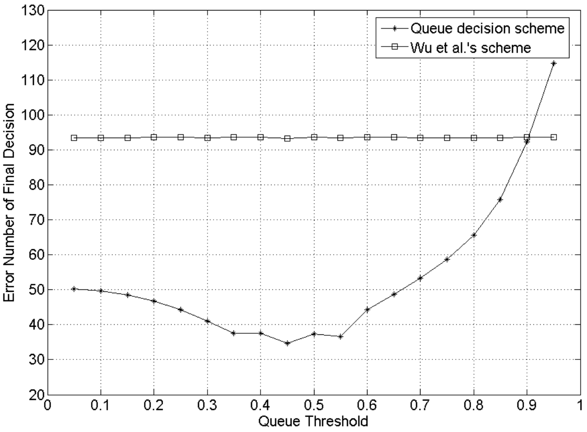

Fig. 7 Error number when $P_e = 0.002$, $N_F = 2$, and stuck-at-one fault

Figures 8 and 9 show that $N_F = 2$ and the faulty type are random, and event probabilities are 0 and 0.002, respectively. Apparently, these outcomes are similar to the previous Figures 4–7. But the error number in random faulty type is higher than stuck-at-one in general, because the random faulty nodes are more difficult to be detected than stuck-at-one. Also, the behavior of random faulty node is more similar to a normal node than the stuck-at-one node. Therefore, the random fault may not be detected by the fault detection scheme, and the fusion center makes a final decision, which is affected by the faulty local decisions.

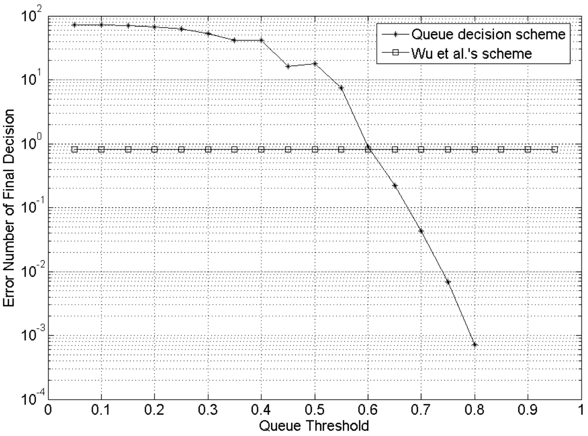

Fig. 8 Error number when $P_e = 0$, $N_F = 2$, and random fault

According to the simulation results, the final decision of the proposed scheme in each experiment condition can get a lowest error number in some T_Q. Therefore, determining the value of T_Q becomes the key point to improve the performance of the proposed scheme. However, setting the value of T_Q to be about 0.7, the proposed scheme is almost better than Wu et al.'s scheme.

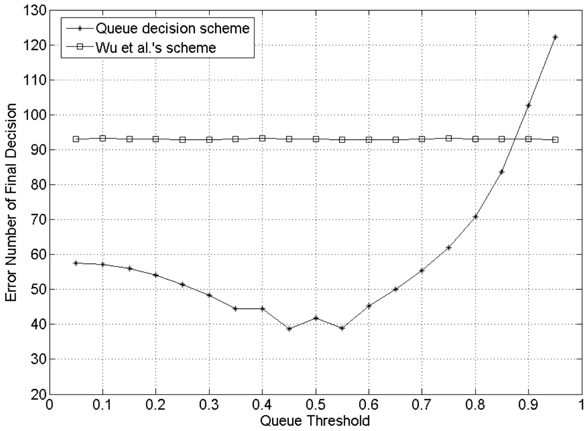

Fig. 9 Error number when $P_e = 0.002$, $N_F = 2$, and random fault

5. CONCLUSION

Because faulty sensor nodes in a WSN always report unreliable information, the fusion center may make wrong decisions according to inaccurate local decisions. Furthermore, the noise lets the fusion center has a higher probability to make a wrong decision. This work investigates the final decision scheme of a WSN and proposes a more event-sensitive scheme to detect the environment condition change with a shorter delay which depend on L_Q and T_Q. To determine the optimal queue length and threshold is our future work.

REFERENCES

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

[2] S.A. Aldosari and J.M.F. Moura, "Detection in decentralized sensor networks," in *Proc. of the 2004 IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, vol. 2, May, 2004, pp.277–280.

[3] J.-F. Chamberland and V.V. Veeravalli, "Asymptotic results for decentralized detection in power constrained wireless

sensor networks," *IEEE Journal of Selected Areas in Communications*, vol. 24, no. 6, pp. 1007–1015, Aug., 2004.

[4] D. Culler, D. Estrin, M. Srivastava, "Overview of wireless sensor networks," *IEEE Computer, Special Issue in Sensor Networks*, vol. 37, issue 8, pp. 41–49, August 2004

[5] A.M. D'Costa and A.M. Sayeed, "Data versus decision fusion for distributed classification in sensor networks," in *Proc. of the Sixth International Conference of Information Fusion*, vol. 2, 2003, pp. 889–894.

[6] W. Du, j. Deng, Y.S. Han, and P.K. Varshney, "A Witness-Based Approach for Data Fusion Assurance in Wireless Sensor Networks," in *Proc. of GLOBECOM 2003*, vol. 3, Dec., 2003, pp. 1435–1439.

[7] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "On-Line Fault Detection of Sensor Measurements," *Proc. of IEEE Sensors*, vol. 2, pp. 974–979, Oct., 2003.

[8] H.-T. Pai and Y.S. Han, "Power-Efficient Direct-Voting Assurance for Data Fusion in Wireless Sensor Networks," *IEEE Trans. on Computers*, vol. 57, no. 2, pp. 261–273, Feb., 2008.

[9] P.K. Varshney, Distributed Detection and Data Fusion, Springer-Verlag, New York, 1997.

[10] J.-Y. Wu, D.-R. Duh, T.-Y. Wang, and L.-Y. Chang, "On-line sensor fault detection scheme for wireless sensor networks," in *Proc. of the 2007 Int. Conf. on Wireless Networks*, Monte Carlo Resort, Las Vegas, Nevada, USA, June, 2007, pp. 329–334.

[11] J.-Y. Wu, D.-R. Duh, T.-Y. Wang, and L.-Y. Chang, "On-line sensor fault detection based on majority voting in wireless sensor networks," in *Proc. the 24th Workshop on Combinatorial Mathematics and Computation Theory*, National Chi Nan University, Puli, Nantou, Taiwan, April, 2007, pp. 292–297.

[12] J.-Y. Wu, *On-Line Fault-Tolerant Decision Fusion Scheme Based on a Record Table in Wireless Sensor Networks*, Master Thesis, National Chi Nan University, Puli, Nantou, Taiwan, May 2007.