Distance Based Fault detection in wireless sensor network

Ayasha Siddiqua*, Shikha Swaroop*, Prashant Krishan*, Sandip Mandal*
*Department of Information Technology, DIT, Dehradun, Uttarakhand.
ayashasiddiquaster@gmail.com

***Abstract*— Wireless Sensor Network (WSNs) have become a new information collection and monitoring solution for a variety of application. In WSN, sensor nodes have strong hardware and software restriction in terms of processing power, memory capability, power supply and communication throughput. Due to these restrictions, fault may occur in sensor. This paper presents a distance based fault detection (DBFD) method for wireless sensor network using the average of confidence level and sensed data of sensor node. Simulation results show that sensor nodes with permanent faults and without fault which was judged as faulty are identified with high accuracy for a wide range of fault rate, and keep false alarm rate for different levels of sensor fault model and also correct nodes are identified by accuracy.**

Keywords-wireless sensor networks; sensor nodes; communication throughput; distance based fault detection; confidence level;

I. INTRODUCTION

A sensor network is composed of a large number of sensor nodes, which are densely deployed either inside the phenomenon or very close to it. The position of sensor nodes need not be engineered or pre-determined. This allows random deployment in inaccessible terrains or disaster relief operations [3]. On the other hand, this also means that sensor network protocols and algorithms must possess self-organizing capabilities. Another unique feature of sensor networks is the cooperative effort of sensor nodes. Sensor nodes are fitted with an on-board processor. Instead of sending the raw data to the nodes responsible for the fusion, sensor nodes use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data.

A. Components of Sensor nodes:

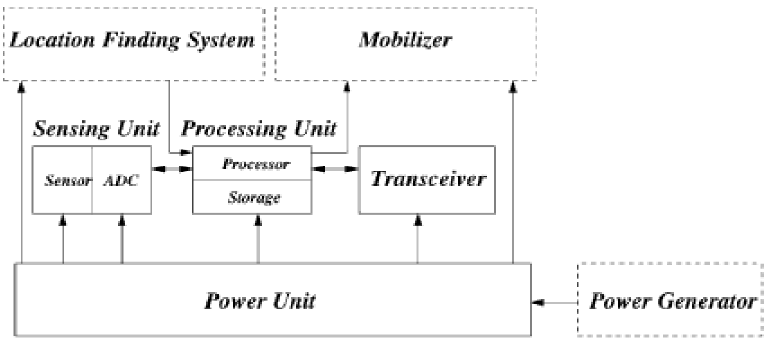

Fig. 1 Components of sensor nodes

B. Factors influencing sensor network design:

A sensor network design is influenced by many factors, which include fault tolerance; scalability; production costs; operating environment; sensor network topology; hardware constraints; transmission media; and power consumption.

Nodes in WSNs are prone to failure due to energy depletion, hardware failure, communication link errors, malicious attack, and so on. Unlike the cellular networks and ad hoc networks where energy has no limits in base stations or batteries can be replaced as needed, nodes in sensor networks have very limited energy and their batteries cannot usually be recharged or replaced due to hostile or hazardous environments. So, one important characteristic of sensor networks is the stringent power budget of wireless sensor nodes. Two components of a sensor node, sensing unit and wireless transceiver, usually directly interact with the environment which is subject to variety of physical, chemical, and biological factors. It results in low reliability of performance of sensor nodes. Even if condition of the hardware is good, the communication between sensor nodes are affected by many factors, such as signal strength, antenna angle, obstacles, weather conditions, interference.

In this paper, we embrace a distance based algorithm for detecting faulty sensor nodes in wireless sensor networks using the average value of confidence level and sensed data. Examples and simulations have validated its effectiveness.

II. RELATED WORK

Fault tolerance issue is widely considered a key part of network management due limited energy and communication link failure. To address this issue, many fault tolerance mechanisms are proposed [13-18]. In [16], fault tolerance mechanisms vary in form of architecture, protocols, detection algorithms, detection decision fusion algorithms [19-22]. Some research focuses on only bias fault tolerance. To address this issue fault detection by quartile method (FTQM) was proposed. However, this technique detects only the bias fault by sensing the data. Whereas the sensor may give the random noise fault, offset fault, gain fault and stuck at fault. Some research are also done on these faults, but it uses the sensed data for creating the neighbor table.

Let node(i) and node(j) are sensor nodes and data(i) and data(j) are data sensed by node(i) and node(j) respectively.

If $|data(i) - data(j)| \leq \delta$ then node(i) and node(j) are said to be neighbor of each other. Where δ is the threshold value. It is fixed according to the application. Because of this, the neighbor table contain the lots of data, therefore it become complex to manage. Some research paper uses the communication radius of $\sqrt{3}$, which we will use for creating the neighbor table.

In this paper, we comprise the Distance based fault detection (DBFD) method for wireless sensor network for detecting random noise fault, gain fault, offset fault and stuck-at fault with the less complexity. In this method we create the neighbor table using the network radius of $\sqrt{3}$.

We take the network radius of all nodes with radius $\sqrt{3}$. The sensor node come into the radius of any node will be considered as a neighbor of that node.

III. MODELS

A. Network Model

In the fault detection of wireless sensor networks, we assume that all the sensor nodes have the same transmission range. Sensor nodes can be deployed or placed in grid and random locations. Nodes with faulty sensors and permanent communication faults (including lack of power) are to be identified, and to be removed from the network. Sensor nodes which generation incorrect sensing data or fail in communication sporadically are treated as usable nodes, and thus are diagnosed as fault-free. Let p represent the probability of failure of a sensor, and let r denote the probability that a faulty node has a fault-free communication unit. If n sensor nodes are under detection, np nodes are faulty. Among them np(1-r) nodes are unable to communicate with their neighbours. Only n(1 - p) + npr nodes are involved in fault detection. Hence the new probability is defined by:

$$P = \frac{npr}{n(1-p)+npr}$$

$$\therefore\ P = \frac{pr}{1-p+pr}$$

It is the probability that a sensor node is faulty after excluding those nodes not responding at all.

B. Data Model

We can define the data model of the sensor networks as follows. As sensors are deployed densely, usually the spatial correlations are satisfied. We calculate the distance of each node from base station. For each fault-free sensor node its distance will be less than δ. Where δ is the threshold value. In this, δ is set to 50.

C. Fault Model

Fault may occur at different levels of WSN, such as physical layer, hardware, system software, and middleware [11]. As sensors are most prone to malfunction, we focus on the fault-free sensors. That is to say, nodes are still able to communicate and process when their sensors are faulty. Taking account of sensor measurements, we assume the following sensor fault models [11, 12]: (i) Stuck-at fault, a fault sensor constantly report a fixed reading. (ii) Gain fault, the measured data of a fault sensor is manifested as a calibration gain to the right value. (iii) Offset fault, the measured data of a faulty sensor is manifested as a calibration offset to the right value. (iv) Random noise fault, the measured data of a faulty sensor is affected by a zero-mean noise with high variance.

IV. DISTANCE BASED FAULT DETECTION

A. Fault Detection

For a wireless sensor network with *n* number of deployed sensors, we can define it as communication graph. A communication graph of a wireless sensor network can be represented as a digraph *G*(*V*,*E*), where *V* represents the set of all sensor nodes in the network and *E* represents the set of edges connecting sensor nodes. Two nodes node(i) and node(j) are said to have an edge in the graph if the distance between them is less than transmission range. For the digraph *G*(*V*,*E*) , node(i) Є *V* and $1\leq\ i \leq n$, the set of the neighbors of node(i) , N(node(i)) is defined to be N(node(i)) = {node(j) Є V : (node(i),node(j) Є V) }. Let the measured data of a sensor node(i) be data(i) , some of them may be incorrect data. Now the sensor node(i) , it has *M* neighbors and their measured values are data(j) (1,2... *M*), and their corresponding weights are confi(j) (1,2... *M*) , which represent their corresponding confidence levels. Table summarizes the notations we will use in this.

TABLE I. SUMMARY OF NOTATIONS

Symbol	**Definition**
P	Probability of failure of a sensor
N	Total number of deployed sensors
V	Set of all the sensors
N(node(i))	Set of neighbours of node(i)
M	Number of neighbour sensors
data(i)	Measurement of sensor node(i)
av(i)	Weighted average of node(i)’s neighbour’s measurement
Confi	Sensor’s confidence level
confi(max)	Initial confidence level of node(i)’s each neighbour
δ	Predefined Threshold value

B. Weighted Average

Now we consider the weighted average of the neighbor sensor’s measurement. Let data(j) (j= 1, 2...*M*) are the sensor '*s* node(j)’s *M* neighbor’s measured values and confi(j)(j = 1,2...*M*) are corresponding weights. The weighted average can be originated as follows:

$$av(i) = AVRG\left\{data(j)\Big|\frac{M}{j=1}\right\}$$

$$av(i) = \frac{\sum_{j=1}^{M} confi(j) * data(j)}{\sum_{j=1}^{M} confi(j)}$$

According to the measurement data(i) of the sensor node node(i) and the weighted average av(i) of its neighbor’s sensor measurements, the decision function f(data(i), av(i)) is as follows:

$$f\big(data(i), av(i)\big) = f(x) = \begin{cases} 0, & |data(i) - av(i)| \leq \delta \\ 1, & otherwise \end{cases}$$

Where δ is a predefined threshold. If deviation of measure value from the true value is less than δ, the node node(i) is considered as fault-free. Based on the decision function, confidence level will be explained. Let a positive integer ‘confi’ represent the confidence level of a sensor. confi(max) is the initial confidence level for all sensor. At the beginning, we can set each sensor with the same confi(max) as an initial confidence level. During the fault detection, we set confi(i) = confi(i) – 1, if f(data(i),av(i)) = 1. When confi(i) reaches zero, the sensor node(i) should be reported to a base station.

C. Algorithm for distance based fault detection

Create the structure and visualize it.

1. For i = 1:node
    a. For each node(i) plot the circle of √3 range.
    b. Find out how many sensors fall inside the range.
    c. Find out the number except the node selected in for loop
    d. If any of the nodes has no neighbours, break the loop.
        i. Else initialize the position of sensors.
2. End for loop.

```
Initialize the faulty node.                  /* For checking, we initialize the faulty nodes. */
1.  For p = 0.05:0.05:0.5                    /*We take the probability of sensor for being faulty*/
    a.  For i = 1:node
        i.   node(i).confi = 5               /* Initialize the confidence level */
        /* Initialization of faulty node*/
        ii.  if  rand(1,1) < p
             1.  node(i).faulty = 1          /* Attribute faulty in sensor node.*/
                 /* for stuck at fault*/
             2.  node(i).data = Initialize with constant value.
                 /* for random noise fault */
             3.  node(i).data = Initialize with any random number between 1 to 1000
                 /* for gain / offset fault */
             4.  node(i).data = distance from base station + any random value
             5.  Create the list which contains all faulted sensor information.
        iii. Else
                 /* initialize the non- faulty nodes */
             1.  node(i).faulty = 0
             2.  node(i).data = distance from base station
        iv.  End if
    b.  End inner for loop
2.  End outer for loop

Find the faulty nodes
1.  For i = 1:10                             /* for each time instance */
    a.  For j = 1:node
        i.    orred = node(i).data           /* sensed data */
        ii.   ls = node(i).neighbour         /* contain neighbour list of sensor node(i) */
              /* weighted average calculation */
        iii.  av = 0
        iv.   denom = 0
        v.    for k = 1:length(ls)
              1.  av = av + node(i).confi * node(i).data
              2.  denom = denom + node(i).confi
        vi.   End for loop
        vii.  orav = av / denom
              /* check faulty node */
        viii. if |orred - orav| > 50
              1.  node is faulty
              2.  node(i).confi = node(i).confi – 1
        ix.   End if
        x.    If node(i).confi = 0
              1.  Faulted_node = node(i)
        xi.   End if
    b.  End inner for loop
2.  End outer for loop
```

V. SIMULATION – RESULTS

The performance of the distance based fault detection algorithm is evaluated by computer simulation. It depends on two parameters: the probability that a sensor node is faulty P , and the kinds of fault models (stuck-at fault, gain fault, offset fault and random noise fault). It is difficult to differentiate the gain fault from an offset fault without some knowledge of the ground truth measurement. So we combine them in this simulation. In the simulation, we assume that faults are independent of each other. The following two metrics, correct detection rate (CDR) and false alarm rate (FAR) are used to evaluate the performance, where CDR is defined to as the ratio of the number of faulty sensor nodes detected to the total number of faulty nodes and FAR is the ratio of the number of fault-free sensor nodes diagnosed as faulty to the total number of fault-free node. Let α be the number of faulty sensors that are judged as faulty in the network and β be the number of fault-free sensors that are judged as faulty. Thus the CDR can be represented as α/nP and FAR can be denoted as $\beta/n(1-P)$. Obviously, it is important for use to improve the CDR and keep FAR in a relatively low level at the same time. Computer simulation is carried out in a sensor network, where 900 sensor nodes are deployed randomly in a rectangular region of size 30 by 30 units. All the nodes are assumed to have a common communication radius which is set to √3. In the simulation, sensor nodes are assumed to be faulty with probabilities of 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, and 0.50. The results for stuck-at fault shown in Fig. 2, Fig. 3, for FAR and CDR respectively and random noise detection are shown in Fig. 4 and Fig. 5 for CDR and FAR respectively. With the sensor fault probability increasing, the CDR of stuck-at fault and random noise decrease rapidly, and the FAR increase quickly simultaneously. When $p = 0.50$, for example, extremely low CDR and extremely high FAR achieved at the same time.

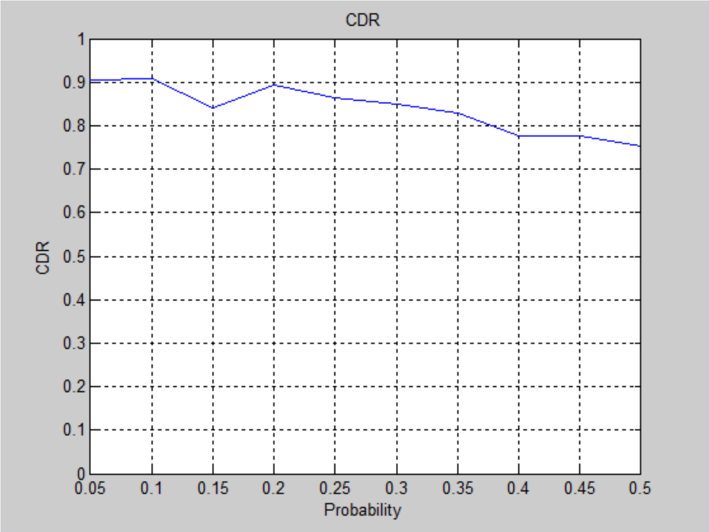

Fig. 2 CDR for Stuck-at fault

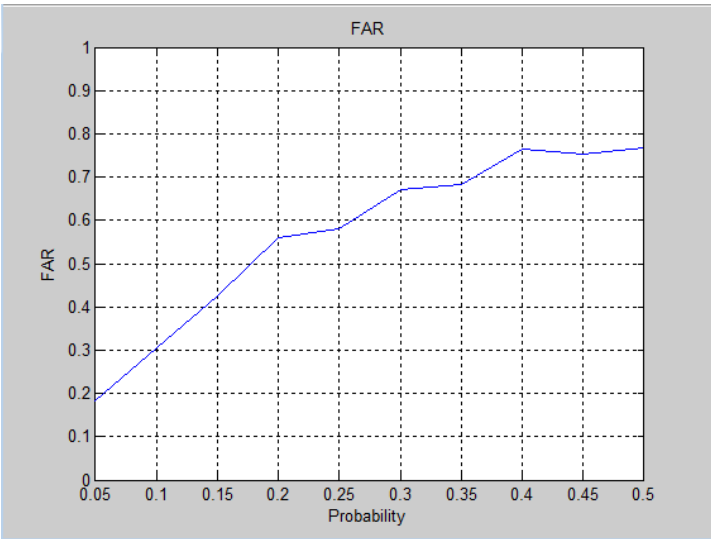

Fig 3. FAR for Stuck-at fault

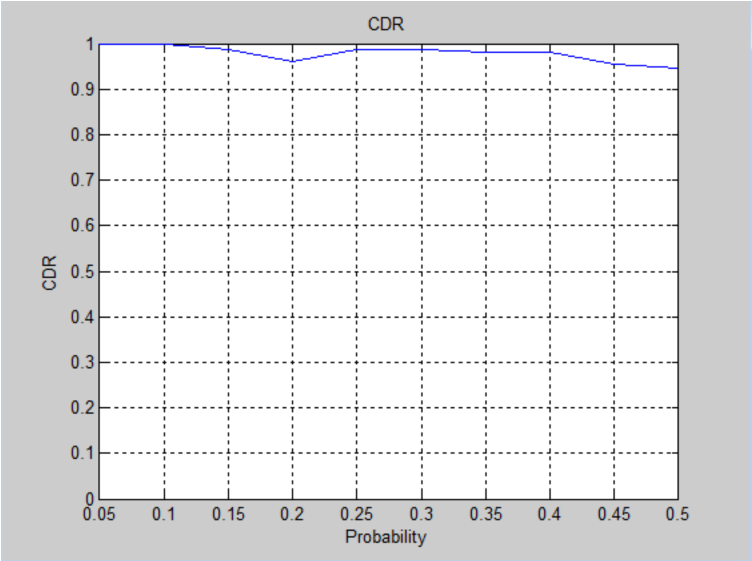

Fig 4. CDR for Random noise fault

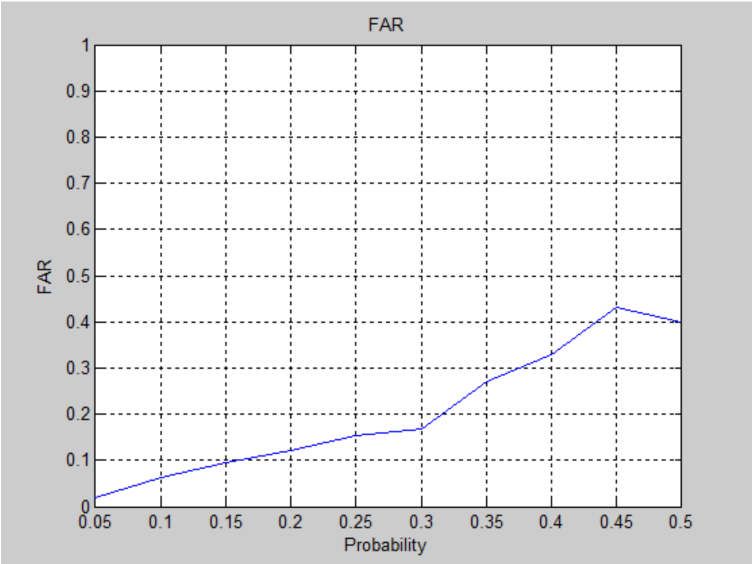

Fig 5. FAR for Random noise fault

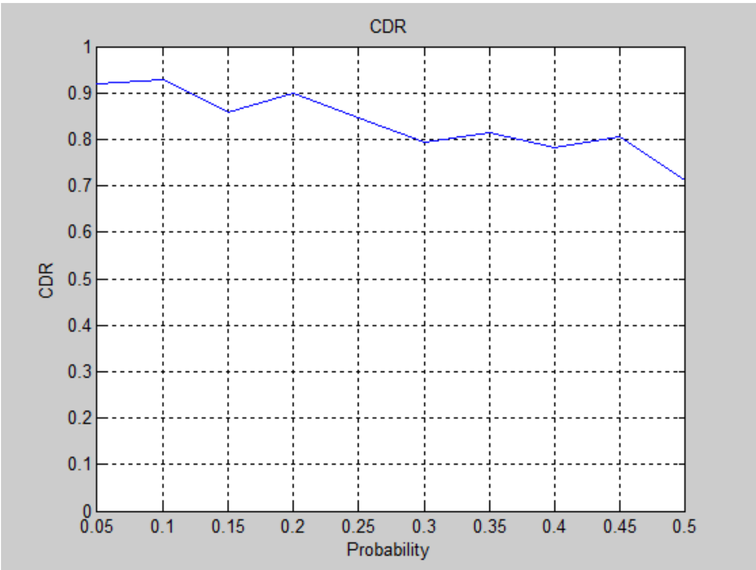

Fig 6. CDR for Gain/offset noise fault

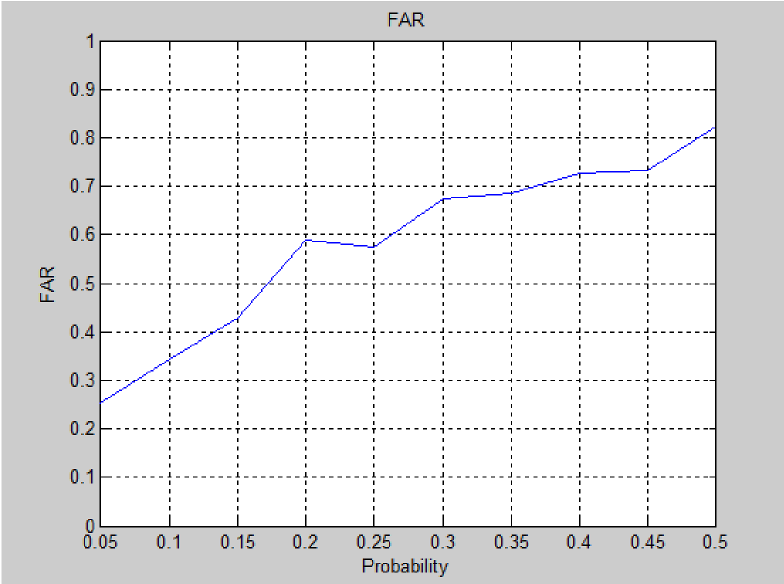

Fig 7. FAR for gain/offset noise fault

Fig. 6 and Fig. 7 show the results of CDR and FAR in gain/offset fault detection for various values of *P*.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have comprises a distance based fault detection algorithm for wireless sensor networks. The algorithm detects the sensor fault using spatial and time information simultaneously. Each sensor node identifies its own status based on local neighbor's average sensed data with some thresholds. Different weights to neighbor measurements are used to detect faulty sensor nodes with high accuracy for a wide range of fault probabilities, while maintaining low false alarm rate. In our future work, we will extend and modify the fault detection algorithm to tolerate transient faults in sensor reading and inter-node communication.

REFERENCES

[1] B. R. Badrinath, M. Srivastava, K. Mills, J. Scholtz,and K. Sollins. Special issue on smart spaces and environments. IEEE Personal Communications, Oct 2000.

[2] S. Lindsey, C. Raghavendra, and K. Sivalingam. Data gathering in sensor networks using the energy delay metric. In International Workshop on Parallel and Distributed Computing: Issues in Wireless Networks and Mobile Computing, San Francisco, USA, Apr 2001.

[3] Aboelaze M, Aloul F. Current and future trends in sensor networks: A survey. Wireless and Optical Communications Networks, 2005: 551-555.

[4] Koushanfar F, Potkonjak M, Sangiovanni-Vincentell A. Fault tolerance techniques for wireless ad hoc sensor networks. In: Proceedings of IEEE on Sensors. 20022: 1491-1496.

[5] B. Krishnamachari, S. Iyengar, Distributed Bayesian algorithms for fault tolerant event region detection in wireless sensor networks, IEEE Transactions on Computers 53 (3) (2004).

[6] F. Koushanfar, M. Potkonjak, A. Sangiovanni-Vincentelli, On-line fault detection of sensor measurements, IEEE Sensors 2 (2003) 974–980.

[7] L.B. Ruiz, I.G. Siqueira, L.B. Oliveira, H.C. Wong, J.M.S. Nogueira, A.A.F. Liureiro, Fault management in event-driven wireless sensor networks, in: MSWIM'04, 2004.

[8] M. Ding, D. Chen, K. Xing, X. Cheng, Localized fault-tolerant event boundary detection in sensor networks, IEEE Info com (2005) 902–913.

[9] X. Luo, M. Dong, Y. Huang, On distributed fault-tolerant detection in wireless sensor networks, IEEE Transactions on Computers 55 (1) (2006) 58-70.

[10] J. Chen, S. Kher, A. Somani, Distributed fault detection of wireless sensor networks, in: Proceedings of 2006 Workshop DIWANS, pp. 65-72.

[11] Basile C, Gupta M, Kalbarczyk Z, Iyer R K, An approach for detecting and distinguishing errors versus attacks in sensor networks. In: Proceedings of the 2006 International Conference on Dependable Systems and Networks. (2006) 473-484.

[12] Gao Jianliang, Xu Yongjun, Li Xiaowei,Online Distributed Fault Detection of Sensor Measurements, Tsinghua science and technology,(2007)192-196.

[13] K. Ozaki, K. Watanabe, S. Itaya, N. Hayashibara, T. Enokido, and M. Takizawa, "A Fault-Tolerant Model of Wireless Sensor-Actor Network," *Proc. of IEEE International Conference on Object and Component-Oriented Real-Time Distributed Computing*, Tokyo Denki Univ., Japan, pp. 186-193, April 2006.

[14] Lilia, Paradis, and Qi Han, "A Survey of Fault Management in Wireless Sensor Networks," *Journal of Network and SystemsManagement*, vol. 15, no. 2, pp. 171-190, June 2007.

[15] Myeong-Hyeon Leea and Yoon-Hwa Choi, "Fault Detection of Wireless Sensor Networks," *Computer Communications*, vol. 31, issue 14, pp. 3469-3475, September 2008.

[16] M. Yu, H. Mokhtar, and M. Merabti, "Fault Management in Wireless Sensor Networks," *IEEE Wireless Communications*, vol. 14, issue 6, pp. 13-19, December 2007.

[17] X. Luo, M. Dong, and Y. Huang, "On Distributed Fault-Tolerant Detection in Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 55, issue 1, pp. 58-70, January 2006.

[18] M. Cardei, Shuhui Yang, and Jie Wu, "Fault-Tolerant Topology Control for Heterogeneous Wireless Sensor Networks," *Proc. Of IEEE International Conference on Mobile Ad hoc and Sensor Systems*, Florida Atlantic Univ., USA, pp. 1-9, October 2007.

[19] L. B. Ruiz, I. G.Siqueira, L. B. Oliveira, H. C. Wong, J.M. S. Nogueira, and A. A. F. Loureiro, "Fault Management in Event-Driven Wireless Sensor Networks," *Prof. of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, New York, USA, pp. 149-156, June 2004.

[20] Jinran Chen, Shubha Kher, and Arun Somani, "Distributed Fault Detection of Wireless Sensor Networks," *Prof. of ACM International Conference on Mobile Computing and Networking*, Los Angeles, USA, pp. 55-64, September 2006.

[21] T. Clouqueur, K. Saluja, and P. Ramanathan, "Fault Tolerance in Collaborative Sensor Networks for Target Detection," *IEEE Transactions on Computers*, vol. 53, issue 3, pp. 320-330, March 2004.

[22] Tsang-Yi Wang, Y.S. Han, P.K. Varshney, and Po-Ning Chen, "Distributed Fault-Tolerant Classification in Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, issue 4, pp. 724-734, April 2005.