

Article

## An Adaptive Fault-Tolerant Event Detection Scheme for Wireless Sensor Networks

Sung-Jib Yim and Yoon-Hwa Choi \*

Department of Computer Engineering, Hongik University, 72-1 Sangsu-Dong, Mapo-Gu, Seoul, Korea;  
E-Mail: yimsungjib@gmail.com

\* Author to whom correspondence should be addressed; E-Mail: yhchoi@cs.hongik.ac.kr;  
Tel.: +82-2-320-1688.

Received: 20 January 2010; in revised form: 25 February 2010 / Accepted: 10 March 2010 /

Published: 19 March 2010

---

**Abstract:** In this paper, we present an adaptive fault-tolerant event detection scheme for wireless sensor networks. Each sensor node detects an event locally in a distributed manner by using the sensor readings of its neighboring nodes. Confidence levels of sensor nodes are used to dynamically adjust the threshold for decision making, resulting in consistent performance even with increasing number of faulty nodes. In addition, the scheme employs a moving average filter to tolerate most transient faults in sensor readings, reducing the effective fault probability. Only three bits of data are exchanged to reduce the communication overhead in detecting events. Simulation results show that event detection accuracy and false alarm rate are kept very high and low, respectively, even in the case where 50% of the sensor nodes are faulty.

**Keywords:** sensor networks; fault tolerance; event detection; adaptive

---

### 1. Introduction

Wireless sensor networks often consist of a large number of small sensor nodes that cooperate to monitor real-world events and enable applications such as target tracking, military tactical surveillance, and emergency health care [1]. The detection and reporting of the occurrence of an interesting event is one of the important tasks of sensor networks. Due to limitations in available resources, such as power, memory and computing capability, sensor nodes deployed in a harsh environment, operating in

an unattended mode, are prone to failure. Faulty nodes might issue an alarm even though they are not in an event region. They degrade the network reliability, unless some provisions are made to tolerate them.

Several distributed schemes for detecting events in the presence of faulty sensor nodes have been proposed in [2–5]. Krishnamachari and Iyengar [2] have mathematically proven that the majority voting is an optimal decision for the given model to detect events and correct faults. A single binary variable is used to represent a local event detection, resulting in low communication cost. Their simulation results show that 85~95% of faults can be reduced when fault rate is about 10%. Luo *et al.* [3] proposed a fault-tolerant energy-efficient event detection paradigm for wireless sensor networks. For a given detection error bound, minimum neighbors are selected to minimize the communication volume. Both Bayesian and Neyman-Pearson detection methods are presented. A localized event boundary detection scheme, exploiting the notion that readings from the event region and the normal region have different means but the same standard deviation due to noise, has been proposed in [4]. Actual sensor readings, encoded in 32 bits each, are transmitted and used in making a decision. The corresponding estimation may be more precise at the cost of increased communication overhead. Jin *et al.* [5] have employed a variable length event coding mechanism in event and event boundary detection to balance the communication cost and the estimation quality. Sensor nodes near the event boundary send the original sensor readings of 32 bits (with a 1-bit flag), whereas all others nodes use only two bits of message, instead.

In [6], a fault-tolerant event boundary detection algorithm using a clustering technique based on maximum spanning trees is presented. Difference in sensor readings between any two sensor nodes is represented as the distance between them. Using the distances sensor nodes are classified into two clusters. With some additional computation on the clusters, event boundary nodes are determined.

Most of the proposed event detection schemes based on a statistical model of noise may work effectively for a relatively low fault probability. As the fault probability increases, however, their performance degrades considerably. Moreover, the actual performance might differ significantly from the estimated one if faults behave differently from the model.

In this paper, we present a distributed adaptive fault-tolerant event detection scheme for wireless sensor networks. It achieves high performance for a wide range of fault probabilities by employing a filter for tolerating transient faults and by dynamically adjusting the threshold for event detection depending on the fault status of sensor nodes. Confidence levels are used to manage the status of sensor nodes. Sensor nodes with a permanent fault (or behaving incorrectly for an extended period of time) are isolated from the network and reinstated later if some required conditions on confidence levels are met. Due to the adaptability of the proposed scheme both high event detection accuracy and low false alarm rate can be maintained even with increasing number of faults.

The remainder of the paper is organized as follows. In Section 2, the system model and fault model are briefly described. Section 3 presents our adaptive event detection scheme employing a dynamic threshold selection. Filtering transient faults is also proposed to reduce the effective fault probability of sensor nodes. Simulation results are shown in Section 4. Conclusions are made in Section 5.

## 2. System Model and Fault Model

As the system model we assume that sensor nodes are randomly deployed in the target area and all sensor nodes have the same transmission range  $r$ . Each sensor node receives the sensor readings of neighboring nodes and makes a decision on an event locally in a distributed manner. We define the average node degree  $d$  to represent the connectivity of the network. For convenience an event region is a circle with radius  $l$ . The proposed adaptive scheme, however, is expected to perform well even with different event region shapes. Each sensor node is assumed to know the range of normal sensor readings, and thus can make a decision on its own whether the sensed data lies in the range of normal readings or not and report a 1(abnormal) or 0(normal) accordingly. Apparently a faulty sensor or an event may produce abnormal data, and thus they are indistinguishable based on the readings of a single sensor node. All the sensor readings are assumed to be binary, without loss of generality. In the case of arbitrary values, comparison diagnosis presented in [7, 8] may be used instead.

Three different types of faults in sensor readings, depending on their temporal behavior, are considered in this paper: permanent, transient, and intermittent [9–11]. In the case of a permanent fault, we assume that it causes an incorrect reading, either 1 or 0, consistently, with the same probability of 0.5, irrespective of the region it is in. Transient faults are assumed to be independent both spatially and temporally. A special type of intermittent fault which generates erroneous data periodically is also taken into account to estimate the adaptability of the proposed scheme. Although we focus on faulty sensors in this paper, the proposed scheme can possibly be extended to cover faulty communications with some degradation in performance by modeling faults in communication as sensor faults in the associated sensor nodes.

Sensor networks are assumed to conduct fault detection periodically to manage fault status of sensor nodes. The period, however, is expected to be long enough to reduce the overhead incurred. Nevertheless the event detection performance can be maintained extremely high as long as most of the faulty sensors nodes are identified and isolated.

## 3. Adaptive Event Detection Scheme

In this section, we first describe the confidence levels of sensor nodes to be used in the proposed event detection scheme. We then present our adaptive event detection scheme using the confidence levels defined. Some erroneous readings due to transient faults will be corrected by employing a moving average filter to further enhance event detection performance. For convenience we list the notation to be used in this paper.

## Notation

---

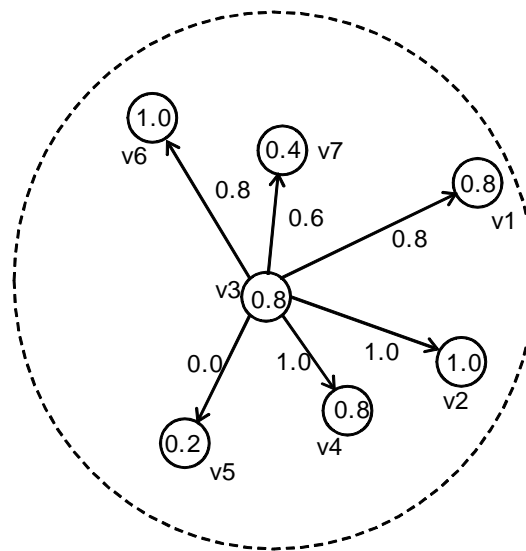
|          |  |
|----------|--|
| $v_i$    | sensor node  |
| $x_i^k$  | sensor reading at node $v_i$ at time $k$   |
| $y_i^k$  | filtered output of the input $x_i^k$ (to tolerate most transient faults)                                     |
| $R_i$    | threshold test result at $v_i$ based on $x_i$ and $x'_j$ s ( <i>i.e.</i> , neighbors')                       |
| $H_i$    | threshold test result at $v_i$ based on $y_i$ and $y'_j$ s   |
| $D_i$    | final decision on an event at $v_i$  |
| $F_i$    | fault status of $v_i$ (good,faulty)  |
| $F_{ij}$ | fault status of $v_j$ from the viewpoint of $v_i$ (good,faulty)  |
| $d_i$    | node degree of $v_i$   |
| $d_i^k$  | effective node degree of $v_i$ at time $k$ ( <i>i.e.</i> , number of neighboring nodes with $F_{ij} = 0$ )   |
| $l$      | radius of an event region  |
| $r$      | transmission range   |
| $d$      | average node degree of a sensor network ( <i>i.e.</i> , $d = \frac{\sum d_i}{N}$ )                           |
| $d^k$    | average effective node degree of a sensor network at time $k$ ( <i>i.e.</i> , $d^k = \frac{\sum d_i^k}{N}$ ) |
| $M$      | window size for tolerating transient faults  |
| $\delta$ | threshold for filtering transient faults   |
| $c_i$    | self confidence level of $v_i$   |
| $w_{ij}$ | confidence level of $v_j$ from the viewpoint of $v_i$  |
| $p_p$    | permanent fault probability  |
| $p_t$    | transient fault probability  |
| $\theta$ | threshold for event detection  |

---

### 3.1. Confidence Levels

In order to describe confidence levels of a sensor node and its neighbors a sensor network is modeled here as a weighted directed graph,  $G(V, E)$ , where  $V$  represents the set of sensor nodes and  $E$  represents the set of edges connecting sensor nodes. Two nodes  $v_i$  and  $v_j$  are said to be connected if the distance between them  $\text{dist}(v_i, v_j)$  is less than or equal to  $r$  (transmission range). Each node  $v_i$  is assigned a self-confidence level  $c_i$ . Each edge  $e_{ij}$  is also assigned a weight  $w_{ij}$ , indicating the confidence level of  $v_j$  from the viewpoint of  $v_i$ . The confidence levels will be used to isolate potentially faulty sensor nodes from the rest of the network. They are also used to reinstate an isolated node if the confidence levels associated with it satisfy the required conditions to be addressed shortly. We use  $c_{min}$  and  $c_{max}$  to denote the range of the confidence level  $c_i$ . Also  $w_{min}$  and  $w_{max}$  will be used to indicate the range of  $w_{ij}$ .

An illustration is given in Figure 1, where six nodes are neighbors of the node  $v_3$  (*i.e.*, six nodes are located within the communication range of  $v_3$ ) and confidence levels  $c_i$  and  $w_{ij}$  are assumed to be in the range of 0 to 1. In the figure, from the viewpoint of node  $v_3$ ,  $v_2$  and  $v_4$  are nodes with the highest confidence while  $v_5$  is a node with the lowest confidence. Among the six neighboring nodes of  $v_3$ ,  $v_5$  is the most likely to be faulty, and will be ignored from  $v_3$  if  $w_{min} = 0$ .

**Figure 1.** An illustration of confidence levels.

The confidence levels will be updated each time a fault detection or event detection is performed. All the  $c_i$  and  $w_{ij}$  are initialized to 1 (*i.e.*,  $c_{max}$  and  $w_{max}$ ). They are increased or decreased by  $\alpha$  ( $0 < \alpha < 1$ ) when the required conditions to be explained later are met.

### 3.2. Filtering Transient Faults

Event detection performance will degrade as the fault probability  $p$  increases. Hence reducing the effective  $p$  is desirable to make an event detection scheme robust to faults occurring in sensor networks. In order to do that, we use the confidence levels defined above to isolate faulty nodes and employ a modified moving average filter, to be discussed here, to correct some erroneous sensor readings due to transient faults.

Let  $x_i^k$  represent sensor reading at node  $v_i$  at time  $k$ . Then the filter we employ takes an average of the last  $M$  readings,  $x_i^n, x_i^{n-1}, \dots$ , and  $x_i^{n-M+1}$ , and sets the output  $y_i^n$  to 1 if it passes a given threshold  $\delta$ . Hence the output  $y_i^n$  (*i.e.*, filtered output at node  $v_i$ ) can be expressed as follows:

$$y_i^n = \begin{cases} 1 & \sum_{j=n-M+1}^n x_i^j \geq M\delta, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Parameters,  $M$  (*i.e.*, window size) and  $\delta$  (threshold) need to be properly chosen, depending on applications, for the best performance. They can be dynamically adjusted to enhance adaptability. As long as most of erroneous readings due to transient faults can be corrected, however, a high event detection performance can be obtained as will be shown in the simulation results in Section 4. Due to the fact that an event may cause abnormal sensor readings for an extended period of time, most transient faults can be filtered unless they occur repeatedly within the window. Although the types of faults may differ depending on applications, most random transient faults can be corrected even with a small window size. The resulting reduction in effective fault probability can affect positively on event

detection performance.

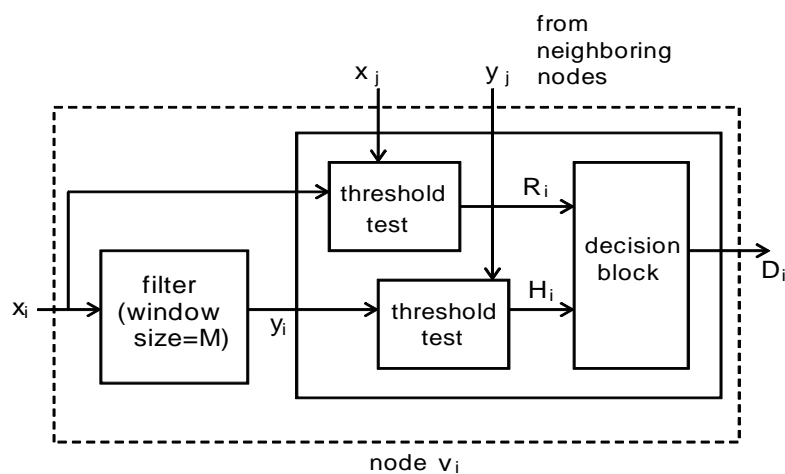
Table 1 shows how erroneous readings due to some transient faults are corrected when  $M = 4$  and  $\delta = 0.75$ . For  $i = 1$ , the filter at node  $v_1$  will generate 0's even if  $x_1^4$  and  $x_1^6$  are 1. In the case of  $i = 5$ , where an event occurs at time 1 and  $v_5$  is assumed to be in the event region, the output becomes 1 with a delay of two cycles. That is,  $y_5^3$  becomes 1.

**Table 1.** An illustration of filtering transient faults when  $M = 4$  and  $\delta = 0.75$ .

| $i$ | $x_i^1$ | $x_i^2$ | $x_i^3$ | $x_i^4$ | $x_i^5$ | $x_i^6$ | $y_i^1$ | $y_i^2$ | $y_i^3$ | $y_i^4$ | $y_i^5$ | $y_i^6$ |
|-----|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 1   | 1       | 0       | 0       | 1       | 0       | 1       | -       | -       | 0       | 0       | 0       | 0       |
| 2   | 0       | 0       | 0       | 1       | 1       | 0       | -       | -       | 0       | 0       | 0       | 0       |
| 3   | 0       | 1       | 0       | 1       | 1       | 1       | -       | -       | 0       | 0       | 1       | 1       |
| 4   | 1       | 1       | 1       | 1       | 0       | 0       | -       | -       | 1       | 1       | 1       | 0       |
| 5   | 1       | 1       | 1       | 1       | 0       | 1       | -       | -       | 1       | 1       | 1       | 1       |

Both  $x'_j$ s and  $y'_j$ s will be used in event detection as shown in Figure 2, where two identical blocks are employed to perform threshold tests (to be addressed shortly) with  $x'_j$ s and  $y'_j$ s, respectively. The resulting binary decisions,  $R_i$  and  $H_i$ , will be given to the subsequent decision block to make a final decision  $D_i$  on an event.

**Figure 2.** Proposed event detection scheme.



In the majority voting in [2], only the upper left threshold test block is employed like most other schemes, although the block could be functionally different. In our proposed event detection scheme both  $R_i$  and  $H_i$  are used. The final decision  $D_i$  on an event will be made based on  $H_i$ , while  $R_i$  is used as a warning of an event.

### 3.3. Dynamic Threshold Selection

In this subsection, we present our adaptive event detection scheme, focusing on the threshold test block in Figure 2, where the confidence levels introduced in the previous subsection will be used to dynamically adjust the threshold for event detection. The confidence levels, updated each time event

detection/fault detection is performed, are utilized to isolate potentially faulty sensor nodes and reinstate them if some given conditions are met. The resulting changes are to be reflected in the number of neighboring nodes (*i.e.*, the effective node degree  $d_i^k$  at time  $k$ ) of each node  $v_i$ , and it will in turn modify the threshold  $\theta$  for the next event detection cycle. In order to realize this adaptivity, each sensor node  $v_i$  holds its fault status  $F_i$ , its self-confidence level  $c_i$ , the confidence levels of its neighboring nodes  $w_{ij}$ , and the fault status of node  $v_j$  from the viewpoint of  $v_i$ ,  $F_{ij}$ .

The proposed event detection scheme, where the threshold  $\theta$  is dynamically adjusted depending on the effective node degree, can be depicted as follows. Majority voting is used in the threshold test.  $F_i$  and  $F_{ij}$  are initialized to 0 (good).

---

#### Adaptive Event Detection Scheme

1. Obtain sensor reading  $x_i$  and filter it to get  $y_i$
  2. Obtain sensor readings  $x_j$ , filtered outputs  $y_j$ , and  $F_j$  from neighbors
  3. Set the threshold  $\theta$  to  $d_i/2$
  4. Determine  $b_i$ , the number of neighbors with  $x_j = x_i$   
     Determine  $q_i$ , the number of neighbors with  $y_j = y_i$
  5. If  $q_i \geq \theta$ , then  $H_i \leftarrow y_i$ , else  $H_i \leftarrow \neg y_i$   
     If  $b_i \geq \theta$ , then  $R_i \leftarrow x_i$ , else  $R_i \leftarrow \neg x_i$
  6. Report an event (*i.e.*,  $D_i = 1$ ) if  $H_i = 1$   
     Report a warning if  $R_i = 1$
  7. Update the confidence levels  $c_i$  and  $w_{ij}$
- 

In steps 1 and 2, each sensor node receives its own and neighbors' sensor readings (including filtered ones). Steps 3 to 5 are functions to be performed in the two threshold test blocks in Figure 2. In step 3, the threshold value for majority voting to be used in step 5 is determined. Step 5 will set  $R_i$  ( $H_i$ ) to either 0 or 1 depending on the number of matching neighbors obtained in step 4.  $R_i$  and  $H_i$  at node  $v_i$  can be set against its own readings if the node fails to pass the threshold. In step 6, the decision on an event will be made.  $R_i = 1$  will be taken as a warning since it might occur due to transient faults. If it is an indication of an event, the decision on an event will be made at the time  $H_i$  becomes 1. The warning must be given to its neighboring nodes to shorten the cycle time momentarily so that an event can be reported quickly. Confidence levels are updated in step 7. The confidence level of  $v_j$  from the viewpoint of  $v_i$ ,  $w_{ij}$ , is updated according to Table 2.

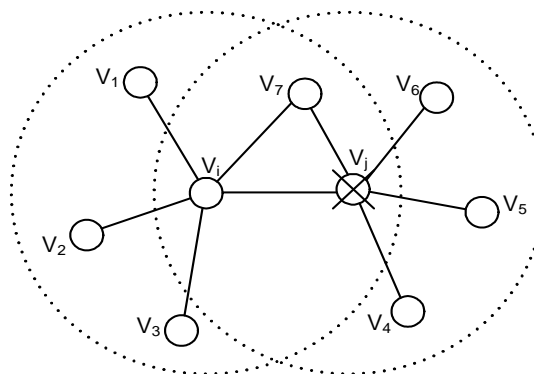
As shown in Table 2,  $w_{ij}$  is increased by  $\alpha$  only when  $F_j = 0$  (good) and  $D_i = y_j$ . In other words, confidence level of  $v_j$  from the viewpoint of  $v_i$  becomes higher when both  $v_i$  and  $v_j$  have similar sensor readings and  $v_j$  is currently in the good state. The second and fourth rows decrease  $w_{ij}$  by  $\alpha$  since  $F_j = 1$  (faulty).

**Table 2.** Updating  $w_{ij}$  at node  $v_i$ .

| $D_i = y_j$ | $F_j$     | $w_{ij}$           |
|-------------|-----------|--------------------|
| yes         | 0(good)   | up                 |
| yes         | 1(faulty) | down               |
| no          | 0(good)   | down for $D_i = 0$ |
| no          | 1(faulty) | down               |

The third row can be explained using the following three representative cases among others. It lowers the confidence level of its neighboring node  $v_j$  only when  $D_i$  is equal to 0.

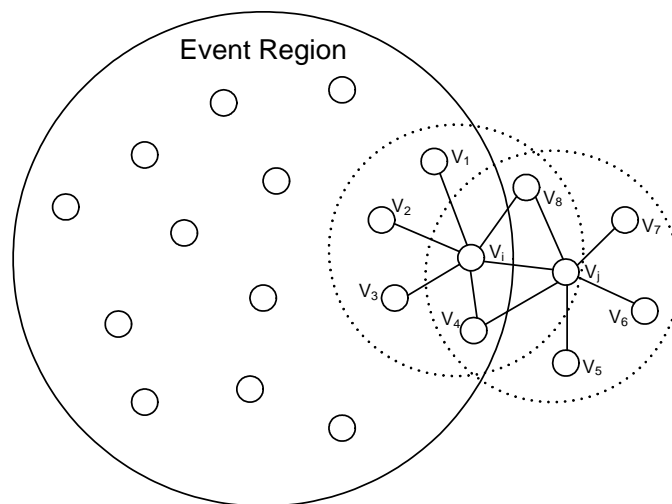
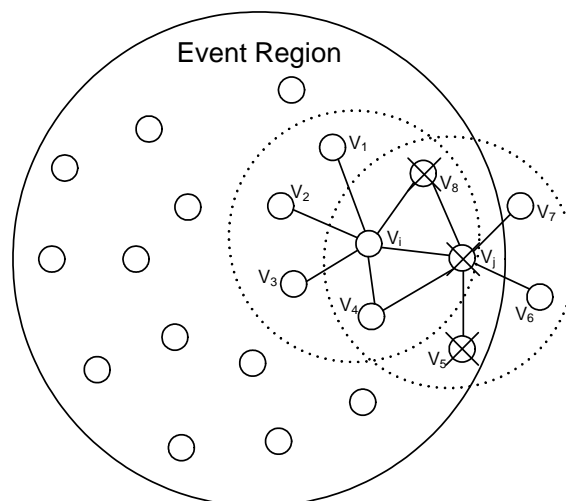
Case 1: Suppose that two good nodes  $v_i$  and  $v_j$  are neighboring each other and each of them is surrounded by sufficient number of good nodes to pass the threshold test. The first case occurs when  $v_j$  becomes faulty and sends a 1 as shown in Figure 3. In this case,  $v_i$  will have  $D_i = 0$ ,  $y_j = 1$ , and  $F_j = 0$  (until  $v_j$  sets  $F_j$  to 1). Hence the conditions are met. The desired action at node  $v_i$ , as far as confidence level is concerned, is to lower the confidence level of  $v_j$  (*i.e.*,  $w_{ij}$ ).

**Figure 3.** Case 1 for the third row in Table 2.

Case 2: The conditions can also be met when two good nodes,  $v_i$  and  $v_j$ , neighboring each other are located in such a way that only one of them is in the event region, as illustrated in Figure 4. In the figure,  $v_i$  is in the event region and receives a 1 from  $v_1$  through  $v_4$  and will eventually report an event (*i.e.*,  $D_i = 1$ ). Meanwhile,  $v_j$  also makes the right decision of no-event (*i.e.*,  $D_j = 0$ ). When  $y_i = 1$  and  $y_j = 0$ , as expected,  $v_i$  will have  $D_i = 1$ ,  $y_j = 0$ , and  $F_j = 0$ , satisfying the conditions. The conditions are also met for  $v_j$  since  $D_j = 0$ ,  $y_i = 1$ , and  $F_i = 0$ . The correct action in case 2, as far as confidence level is concerned, is as follows: (a) at node  $v_i$ ,  $w_{ij}$  needs to be increased, (b) at node  $v_j$ ,  $w_{ji}$  also needs to be increased.

Case 3: It occurs when faulty nodes in close proximity, claiming to be good, are in an event region as shown in Figure 5 such that their readings are 0 as opposed to 1 (abnormal). Suppose that two nodes in the event region,  $v_i$  and  $v_j$ , are neighboring each other and  $v_j$  is one of the faulty nodes. Apparently  $v_j$  may have  $D_j = 0$  since  $v_6$  and  $v_7$  are likely to report a 0 since they are outside the event region. Both  $v_i$  and  $v_j$  meet the conditions. The proper actions in this case are (a) at node  $v_i$ , where  $D_i = 1$ ,  $y_j = 0$ , and  $F_j = 0$ ,  $w_{ij}$  has to be lowered, (b) at node  $v_j$ , where  $D_j = 0$ ,  $y_i = 1$ , and  $F_i = 0$ ,  $w_{ji}$  needs to be increased to eventually change  $F_j$  to 1.



**Figure 4.** Case 2 for the third row in Table 2.**Figure 5.** Case 3 for the third row in Table 2.

For node  $v_i$  the above cases can be divided into two groups, depending on the value of  $D_i$ . The first group ( $D_i = 0$ ) includes case 1, case 2(b), and case 3(b). Although the three cases in the first group cannot be distinguished based on the given information, the desired actions may differ. Only case 1 wants to lower the confidence level. The second group ( $D_i = 1$ ) includes case 2(a) and case 3(a), requesting conflicting actions. The third row in the table allows only case 1 to update the confidence level, ignoring all other cases. The reasons for taking this action are as follows. Confidence levels are maintained to isolated nodes with permanent faults or nodes behaving incorrectly for some extended period of time. Hence it is primarily intended to handle case 1. All other cases are related to events, which in general consume a relatively small portion of the entire monitoring time. In the case of an event, due to the conflicting requests, correctly updating confidence levels needs some additional information on the exact boundary of the event region, requiring more sophisticated computations. Hence momentarily stopping the updates in the case of an event may be appropriate since the network continues its monitoring function with most of the faulty nodes isolated.

Based on Table 2 the confidence level  $w_{ij}$  is updated as follows.

$$w_{ij} = \begin{cases} \max(w_{min}, w_{ij} - \alpha) & \text{if } (D_i \neq y_j \text{ and } D_i = 0) \text{ or } F_j = 1 \\ \min(w_{max}, w_{ij} + \alpha) & \text{if } D_i = y_j \text{ and } F_j = 0 \\ w_{ij} & \text{otherwise} \end{cases} \quad (2)$$

It is increased or decreased by  $\alpha$  each time the conditions are met. The value of  $\alpha$  needs to be chosen depending on the types of faults and applications. If  $\alpha$  is relatively small, a node with transient faults is highly unlikely to be removed from the neighbor list. As  $\alpha$  increases, however, it can be removed with an increased probability. Even if it is isolated, the node with only transient faults will be reinstated in our adaptive scheme.

A potentially faulty neighboring node  $v_j$  of node  $v_i$  will be removed from the effective neighbor list of  $v_i$  as follows. If  $F_{ij} = 0$  (good) and  $w_{ij} = w_{min}$ ,  $F_{ij}$  is set to 1 (faulty) and  $v_j$  is removed from  $v_i$ 's effective neighbor list. On the other hand, if  $F_{ij} = 1$  (faulty) and  $w_{ij} = w_{max}$ ,  $F_{ij}$  will be set to 0 (good) and  $v_j$  will rejoin the  $v_i$ 's effective neighbor list. Once a node is removed from the list (*i.e.*,  $w_{ij} = w_{min}$ ), it can rejoin the list only when  $w_{ij}$  is increased and reaches  $w_{max}$ . Similarly, once a removed node rejoins the effective neighbor list, it will remain there unless  $w_{ij}$  reaches  $w_{min}$  again.

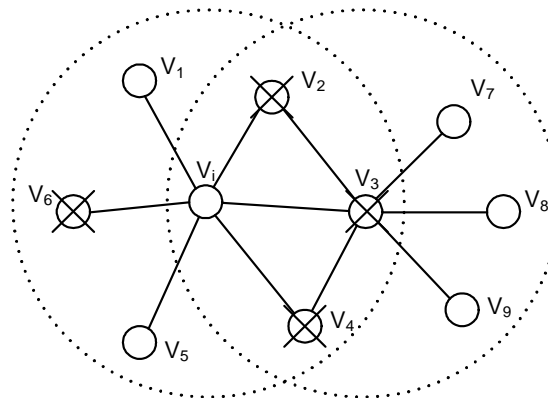
Similarly the self-confidence level of  $v_i$ ,  $c_i$ , is also updated in step 7. It is lowered if the decision made at  $v_i$ ,  $D_i$ , is different from its own sensor reading filtered,  $y_i$ , except for an event.

$$c_i = \begin{cases} \max(c_{min}, c_i - \alpha) & \text{if } D_i \neq y_i, \\ \min(c_{max}, c_i + \alpha) & \text{otherwise} \end{cases} \quad (3)$$

Fault status  $F_i$  changes depending on the self confidence level  $c_i$ .  $F_i$  will be set to 1 (faulty) when  $c_i$  becomes  $c_{min}$ . Once it is set to 1, it will stay there until  $c_i$  reaches  $c_{max}$  again.

In the case where a good sensor node has more faulty neighbors, the node might be determined to be faulty, as illustrated in Figure 6, where  $c_i$  for  $v_i$  will be lowered due to the inequality  $D_i \neq y_i$ . It, however, will highly likely be determined to be a good node with time. The node,  $v_3$ , a neighbor of  $v_i$ , will determine itself to be faulty if it cannot pass the threshold such that its confidence level  $c_3$  reaches 0. In the figure,  $v_3$  has more good neighbors than faulty ones. Hence  $D_3$  is highly unlikely to be  $y_3$ . Once  $F_3$  is set to 1,  $v_i$  will remove  $v_3$  from its neighbor list. As a result, its effective node degree  $d_i^k$  will be lowered. If this also happens at  $v_4$ , for example, the node is also removed from the list, and the node degree of  $v_i$  is further lowered. Finally,  $v_i$  passes the threshold, changes its fault status to 0 (good) some cycles later, and it can then be treated as a good node. If a larger number of faulty nodes are in close proximity, this recovery might not happen. The case, however, is extremely unlikely since our adaptive scheme removes faulty nodes as soon as identified. Unless all the nodes become faulty almost simultaneously, such a situation is unlikely to occur.

**Figure 6.** A good node failing to pass the threshold due to neighboring faulty nodes.



#### 4. Simulation Results

Computer simulation is conducted to evaluate the performance of the proposed event detection scheme. Our simulated sensor network consists of 1,024 sensor nodes, randomly deployed in a  $32 \times 32$  square region. Initially each node has about 12 neighboring nodes on average (*i.e.*,  $d \approx 12$ ) in the simulation. Event region is assumed to be a circle with radius  $l = 2r$ , where  $r$  is the transmission range of each sensor node. Nodes with a permanent fault are assumed to consistently report an unusual reading (similar to stuck-at-1) or a normal reading (similar to stuck-at-0) with the same probability of 0.5, irrespective of the regions they are in. Both permanent and transient faults are considered and their probabilities are denoted by  $p_p$  and  $p_t$ , respectively. Hence the overall fault probability  $p$  is equal to  $p_p + p_t$ . In filtering transient faults,  $M$  (window size) and  $\delta$  (threshold) are set to 4 and 0.75, respectively. In the simulation, three different values of  $\alpha$ , 0.1, 0.2 and 0.3, are chosen for comparison purposes.

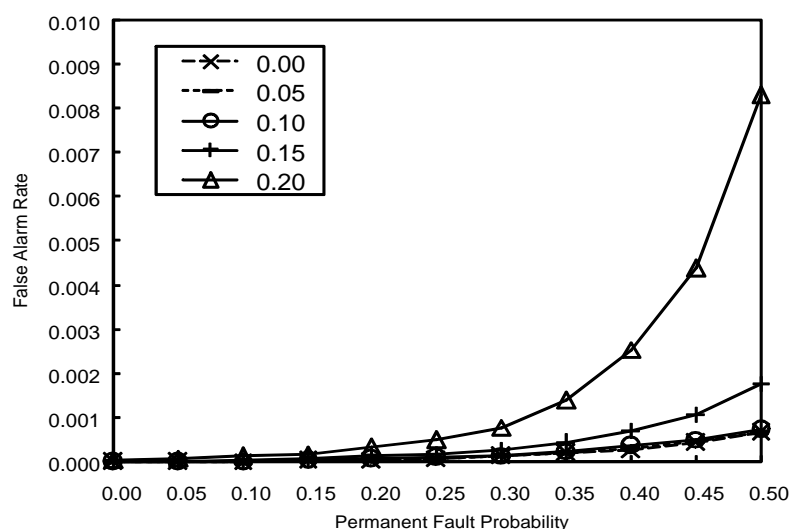
Three metrics, DA(event detection accuracy), FAR (false alarm rate) and ERDR (event region detection rate), are used to evaluate the performance of the proposed event detection scheme. FAR is defined as the ratio of the number of nodes reporting an event, in the case of no event, to the total number of sensor nodes. DA is the ratio of the number of times that events are detected to the total number of event occurrences. ERDR is the ratio of the number of nodes, in the event region, reporting an event (*i.e.*,  $D_i = 1$ ) to the total number of nodes in the event region. Our objective is to keep high DA and low FAR simultaneously even when the fault probability is high. Although ERDR is not the main concern in this paper, statistical data for event region detection are obtained for future research.

Table 3 shows DA for the proposed event detection scheme for various values of  $p_t$  when  $p_p$  is increased by 0.01 every 20 cycles up to 0.5. Based on the results we can claim that DA can be maintained high even with increasing number of faults.

**Table 3.** DA for various values of  $p_p$  and  $p_t$ .

| $2^*p_p$ | $p_t$ |       |       |       |       |
|----------|-------|-------|-------|-------|-------|
|          | 0.00  | 0.05  | 0.10  | 0.15  | 0.20  |
| 0.00     | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 0.10     | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 0.20     | 1.000 | 1.000 | 1.000 | 1.000 | 0.999 |
| 0.30     | 1.000 | 1.000 | 1.000 | 1.000 | 0.995 |
| 0.40     | 1.000 | 1.000 | 1.000 | 0.993 | 0.949 |
| 0.50     | 0.999 | 0.999 | 0.997 | 0.933 | 0.753 |

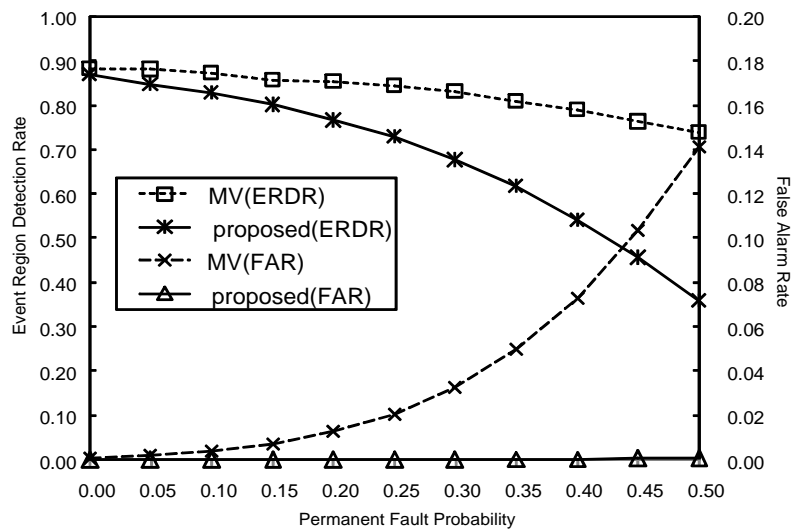
Figure 7 shows FAR with increasing permanent fault probability  $p_p$  for various values of transient fault probability  $p_t$  when  $\alpha = 0.2$ . To see how the proposed scheme adapts to the increase in the number of faults,  $p_p$  is increased by 0.01 every 20 cycles. FAR is kept very close to zero even when  $p_p$  is 0.5. In the case of  $p_t = 0.1$  and  $p_p = 0.2$ , for example, FAR is about 0.00006. That is, only 0.06 nodes out of 1,024 make a false alarm even in the combined fault probability of 0.3. Sensor nodes with a permanent fault (producing erroneous data repeatedly for an extended period of time) can hardly affect the decision making process since they will be isolated from the network until they exhibit normal behavior again. In addition, the increase in transient fault probability  $p_t$ , up to 0.2, does not cause any notable performance degradation due to the effective filtering of transient faults.

**Figure 7.** FAR with increasing  $p_p$  for various values of  $p_t$ .

We have compared the performance of the proposed scheme with that of the majority voting. The results for  $p_t = 0.1$ ,  $0.0 \leq p_p \leq 0.5$ , and  $\alpha = 0.2$  are shown in Figure 8. Unlike the proposed scheme, FAR for the majority voting increases with  $p_p$ , exhibiting a significant amount of false alarms. These false alarms will waste the network resources, resulting in a considerable reduction in network lifetime. On the other hand, ERDR for our scheme is lower than that of the majority voting. The reason for this degradation in ERDR is that correcting erroneous readings by employing a filter may reduce the number

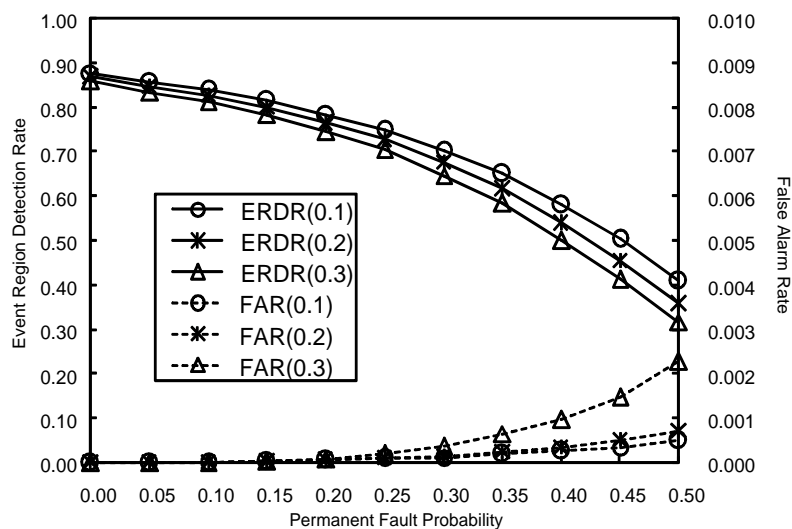
of non-event sensor nodes incorrectly reporting a 1 (abnormal). In fact incorrect readings due to faulty sensor nodes near but outside an event region may affect positively on the event detection.

**Figure 8.** Comparison between the proposed scheme and majority voting(MV) with increasing  $p_p$  when  $p_t = 0.1$ .



Similar simulation is done to compare the performance for three different values of  $\alpha$ : 0.1, 0.2 and 0.3. The resulting FAR and ERDR are shown in Figure 9, where the number in the parenthesis represents the value of  $\alpha$ . As can be seen, the best performance is obtained for  $\alpha = 0.1$ , although the performance difference between 0.1 and 0.2 is marginal. A notable degradation in performance can be observed for  $\alpha = 0.3$ . This stems from the fact that some good nodes are removed from the neighbor list due to transient faults.

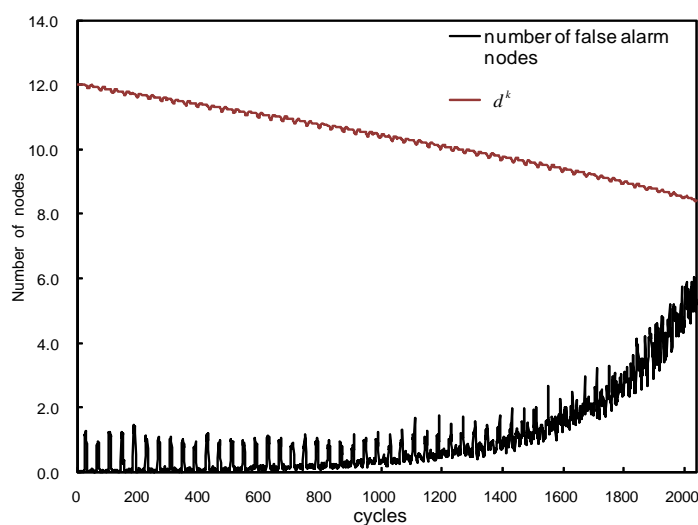
**Figure 9.** ERDR and FAR for three different values of  $\alpha$  when  $p_t = 0.1$ .



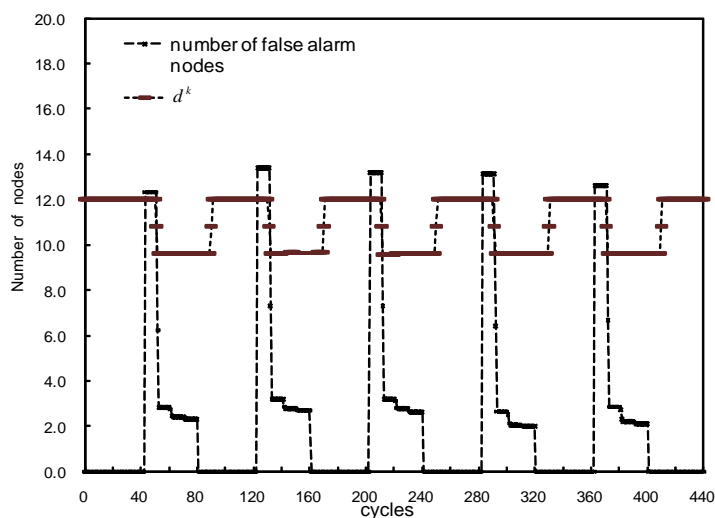
In the proposed adaptive scheme, a sensor node  $v_i$  treats a potentially faulty sensor node  $v_j$  as a faulty node at the time the confidence level  $w_{ij}$  reaches 0. The resulting reduction in effective node degree

of each sensor node,  $d_i^k$ , will accordingly change the threshold  $\theta$  to adapt to the new network topology. Consequently faulty nodes can only affect the decision making process until they are identified and isolated. Due to the dynamic threshold selection, high event detection performance can be maintained even with increasing fault probability as shown in Figure 10, where  $p_p$  is increased by 0.01 every 40 cycles and an event is assumed to occur every 40 cycles. As expected, the average node degree  $d^k$  (at time  $t = k$ ) decreases and the number of false alarm nodes slowly increases with  $p_p$ . The number of false alarm nodes moves up and down periodically due to the artificially generated periodic events.

**Figure 10.** Average node degree  $d^k$  and the number of false alarms when  $p_p$  increases up to 0.5 and  $p_t = 0.2$ .



**Figure 11.** Average node degree  $d^k$  and the number of false alarms when intermittent faults occur simultaneously every 80 cycles with the probability of 0.2.



Another simulation is performed to show how the proposed scheme adapts to a special type of fault, producing erroneous readings periodically for some period of time. For simplicity, each node is assumed to have such an intermittent fault with probability of 0.2 every 80 cycles, producing incorrect readings

for 40 cycles. The results are shown in Figure 11, where the number of nodes that make a wrong decision soars up to more than 12 at the time such a fault occurs, but goes down to below 4 after a few threshold adjustments. Once the erroneous data due to the faults disappear, the threshold goes back to the original position, as expected.

The proposed adaptive scheme has the potential to adapt to different fault patterns. The performance of the scheme will further be investigated by generating various types of faults discussed in [12].

## 5. Conclusions

In this paper, we proposed an adaptive fault-tolerant event detection scheme for wireless sensor networks. It maintains high performance, in terms of detection accuracy and false alarm rate, for a wide range of fault probabilities, by employing a dynamically adjusted threshold and a filter for tolerating transient faults. Simulation results show that the scheme mitigates the negative influence of various types of faults by exploiting adaptation to temporal behavior of faults. Although we focused on faulty sensors, the scheme can be extended to cover faults in communication with minor modifications. Only three bits of information are exchanged each event detection cycle to reduce the communication cost. More extensive simulation is currently being conducted to estimate how the scheme performs for various event region shapes.

## Acknowledgements

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (KRF-2008-313-D00902).

## References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cyirci, E. Wireless sensor networks: a survey. *Comput. Net.* **2002**, *38*, 393-422.
2. Krishnamachari, B.; Iyengar, S. Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Trans. Comput.* **2004**, *53*, 241-250.
3. Luo, X.; Dong, M.; Huang, Y. On distributed fault-tolerant detection in wireless sensor networks. *IEEE Trans. Comput.* **2006**, *55*, 58-70.
4. Ding, M.; Chen, D.; Xing, K.; Cheng, X. Localized fault-tolerant event boundary detection in sensor networks. *IEEE Infocom.* **2005**, *2*, 902-913.
5. Jin, G.; Nittel, S. NED: an efficient noise-tolerant event and event boundary detection algorithm in wireless sensor networks. In *7th International Conference on Mobile Data Management (MDM'06)*, Nara, Japan, 2006; pp. 153-160.
6. Li, C.-R.; Liang, C.-K. A fault-tolerant event boundary detection algorithm in sensor networks. *ICOIN 2007*, *5200*, 406-414.
7. Lee, M.H.; Choi, Y.-H. Fault detection of wireless sensor networks. *Comput. Commun.* **2008**, *31*, 3469-3475.
8. Choi, J.Y.; Yim, S.-J.; Huh, Y.; Choi, Y.-H. A distributed adaptive scheme for detecting faults in wireless sensor networks. *WSEAS Trans. Commun.* **2009**, *8*, 269-278.

9. Koren, I.; Krishna, C.M. *Fault-Tolerant Systems*; Morgan Kaufmann: San Francisco, CA, USA, 2007.
10. Wu, J.; Duh, D.; Wang, T.; Chang, L. Fast and simple on-line sensor fault detection scheme for wireless sensor networks. *Lect. Note. Comput. Sci.* **2007**, *4808*, 444-455.
11. Elhadeif, M.; Boukerche, A.; Elkadiki, H. A distributed fault identification protocol for wireless and mobile ad hoc networks. *J. Par. Dist. Comput.* **2008**, *68*, 321-335.
12. Ni, K. Sensor network data fault types. *ACM Trans. Sensor Net.* **2009**, *5*, 1-29.

© 2010 by the authors; licensee Molecular Diversity Preservation International, Basel, Switzerland. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license <http://creativecommons.org/licenses/by/3.0/>.