International Scholarly Research Network
ISRN Sensor Networks
Volume 2012, Article ID 342461, 13 pages
doi:10.5402/2012/342461

Research Article

An Adaptive Approach to Discriminate the Persistence of Faults in Wireless Sensor Networks

Arunanshu Mahapatro and Pabitra Mohan Khilar

Department of Computer Science and Engineering, National Institute of Technology, Rourkela 769008, India

Correspondence should be addressed to Arunanshu Mahapatro, arun227@gmail.com

Received 20 June 2012; Accepted 18 July 2012

Academic Editors: M. Brandl, L. Reggiani, and Y. Yu

This paper presents a parametric fault detection algorithm which can discriminate the persistence (permanent, intermittent, and transient) of faults in wireless sensor networks. The main characteristics of these faults are the amount the fault appears. We adopt this state-holding time to discriminate transient from intermittent faults. Neighbor-coordination-based approach is adopted, where faulty sensor nodes are detected based on comparisons between neighboring nodes and dissemination of the decision made at each node. Simulation results demonstrate the robustness of the work at varying transient fault rate.

1. Introduction

Node failures and environmental hazards cause frequent topology change, communication failure, and network partition. Such perturbations are far more common in wireless sensor networks (WSNs) than those found in traditional wireless networks. The extent of such a perturbations depends on the persistence of faults. Based on persistence, faults can be classified as transient, intermittent, or permanent. A transient fault will eventually disappear without any apparent intervention, whereas a permanent one will remain unless it is removed by some external agency [1]. After their first appearance, the rate of fault appearance is relatively high for intermittent faults, and finally the intermittent faulty nodes tend to become permanent [2, 3]. Permanent or hard faults are software or hardware faults that always produce errors when they are fully exercised [4].

In fact, experimental studies have shown that more than 80% of the faults that occur in real systems are transient or intermittent faults [3, 5, 6]. These faults are more severe, from both data aggregation and network lifetime perspective. They are much problematic to diagnose and handle. In contrast, permanent faults are considerably easier to diagnose and handle. Since the effect of faults is not always present, detection of intermittent or transient faults requires repetitive testing at the discrete time in contrast to single test to detect permanent faults.

Discrimination of transient from intermittent or permanent faults is crucial as a sensor node with transient fault does not necessarily imply that the sensor node should be isolated although the unstable environment might warrant a temporary shutdown [4]. A discrimination between transient and intermittent or permanent faults solves the following key problems.

Effective Bandwidth Utilization. By isolating permanent faults, the traffic generated by the permanent faulty nodes is restricted.

Effective Energy Utilization. The depletion of sensor node battery energy in forwarding the erroneous data generated by permanent faults can be avoided. Isolation of sensor nodes with transient faults will reduce available sensor nodes in the network. This in turn increases the work load of each sensor node, thus leading to faster depletion of sensor node battery energy and impacting network lifetime.

Network Coverage and Connectivity. Isolation of fault-free nodes with transient faults will reduce the available sensor

nodes in the network thus impacting network coverage and connectivity.

Coverage of the Network Fault Hypothesis [7]. The assumption on the number of faults tolerated by the detection algorithm within a given time window is affected by isolation of fault-free nodes with transient faults.

These issues motivate the need to design an efficient fault discrimination algorithm suitable for WSNs. To discriminate transient from intermittent faults, this chapter is motivated from the count and threshold mechanism adopted in [7]. Similar to [7], our approach uses two counters, namely, reward (r) and penalty (z) counter to discriminate fault types with low latency and low energy overhead. A node detected as faulty enters to observation state. Unlike [7], we first tune the intertest interval T to detect the presence of fault with minimum test repetition. Second, we adopt the earlier discussed two-state Markov chain to model fault appearance and disappearance. We consider the time a node spends in the fault disappearance state (sojourn time) to tune the detection parameters. This chapter demonstrates an effective means to discriminate faults based on the persistence by properly tuning the detection parameters. We consider the following detection parameters.

(i) Intertest Interval (T). The time interval of two consecutive sensor measurements.

(ii) Reward Counter Threshold (θ_2). The number of diagnostic rounds, a node under observation, shows expected behavior, after which a node is reintegrated to the network.

(iii) Penalty Counter Threshold (θ_3). The number of correlated diagnostic rounds, after which a node gets isolated.

(iv) Adoptive Penalty Increments. The penalties assigned after a fault is detected.

The following performance metrics are used to tune the mentioned detection parameters.

a. Accuracy is the probability that a fault-free node with transient fault in the error-free state entering the observation phase is not isolated [7].

b. Coverage is the probability that an intermittent faulty node in the error-free state entering the observation phase is isolated [7].

c. Number of test repetitions is the measure of the number of times the test repeated to discriminate transient from intermittent or permanent faults.

The main contributions of this paper are as follows.

(i) We extend the basic design with two practical considerations. First, we propose a diagnosis scheme that identifies faults with high detection accuracy and low false alarm rate by maintaining low latency, low energy overhead, and less dependency on the average node degree. Second, we propose a robust method to accommodate channel fault.

(ii) Our approach discriminates transient from intermittent faults which in turn maintains low false alarm rate.

(iii) We propose an adaptive increment-based scheme to reduce the detection latency.

(iv) Our fault diagnosis algorithm imposes a negligible extra cost in a network where diagnostic messages are sent as the output of the routine tasks of a network.

The remainder of the paper is organized as follows. Section 2 presents related works. Section 3 presents the system model. Distributed diagnosis algorithm is investigated in Section 4. Simulation results are presented in Section 5, and finally conclusions are given in Section 6.

2. Related Work

The context of sensor networks and the nature of sensor data make the design of an efficient fault diagnosis technique more challenging. The conventional fault diagnosis techniques devised for wired interconnected networks [8–13] might not be suitable for WSNs for reasons, namely, resource constraints, random deployment, dynamic network topology, attenuation, and signal loss.

The problem of identifying faulty nodes (crashed) in WSN has been studied in [15]. This paper proposes the WINdiag diagnosis protocol which creates a spanning tree for dissemination of diagnostic information. Chen et al. [16] proposed a localized fault detection algorithm to identify the faulty sensors. It uses local comparisons with a modified majority voting, where each sensor node makes a decision based on comparisons between its own sensor reading (such as temperature) and sensor reading of one-hop neighbors, while considering the confidence level of its one-hop neighbors. The performance of such an approach depends on the average node degree of the network. Jiang [17] claimed an improvement over the aforementioned scheme [16] by introducing an improved distributed fault detection scheme (DFD).

The two-phase neighbor coordination scheme is suggested by Hsin and Liu [18], where a node waits for its neighbors to update information concerning the faulty node in the first phase. It uses the second phase to consult with its neighbors to reach a more accurate decision. Agnostic Diagnosis (AD) [19], an online lightweight failure detection approach, is motivated by the fact that the system metrics (e.g., radio on time and number of packets transmitted) of sensors usually exhibit certain correlation patterns.

FIND [20] detects nodes with data faults. It ranks the nodes based on their measurements as well as their physical distances from the event. A node is detected faulty if there is a significant mismatch between the sensor data ranks, and its readings violate the distance monotonicity significantly. Gao et al. [21] approached WSN fault detection problems by suggesting a weighted median fault detection scheme (WMFDS) which primly focused on the soft fault. Krishnamachari et al. have presented a Bayesian fault recognition model to solve the fault-event disambiguation problem in sensor networks [22].

Most of the fault detection schemes [19, 23–25] are designed to detect permanent faults. In [2], a class of count-and-threshold mechanisms collectively named α-count is suggested, which are able to discriminate between transient faults and intermittent faults in computing systems. The authors have presented a single-threshold scheme and a better performing double-threshold scheme. Serafini et al. [7] proposed to use a count-and-threshold algorithm on top of the diagnostic protocol to reduce the likelihood of isolation and increase the availability of fault-free nodes in case of external transient faults. Their approach uses two values: a *penalty counter* and a *reward counter* to discriminate transient from intermittent fault. They consider that discrete time Markov chain (DTMC) is used to model the behavior of the proposed algorithm. Both the approaches are designed, analyzed, and tested on wired interconnected networks. Thus, the parameters tuned may not be applicable for wireless sensor networks. For instance, in [7], $T = 2.5$ ms. Such a small value for T cannot be adopted in WSNs since it requires frequent exchange of data and thereby impacting the network lifetime.

Lee and Choi [14] approached WSN fault detection problems where nodes with malfunctioning sensors are allowed to act as a communication node for routing, but they are logically isolated from the network as far as fault detection is concerned. Only those sensor nodes with a permanent fault in the transceiver (including lack of power) are to be removed from the network. Time redundancy is used to tolerate transient faults in sensing and communication. However, detection of transient fault is not addressed. Their scheme uses two thresholds to check whether a node is permanent faulty or fault-free with transient faults. Their scheme does not answer the questions like how many tests required to discriminate the faults and what should be the intertest interval.

In summary, most of the proposed diagnosis approaches are designed to detect permanent faults in sensor networks. Some techniques are proposed to tolerate transient faults in fault detection. Though some techniques investigate the fault discrimination problem, they are designed for wired interconnected networks. To the best of our knowledge, this is the first attempt in discriminating transient from intermittent or permanent faults in WSNs.

3. System Model

3.1. Network Model. The proposed algorithm considers a network with n sensor node nonuniformly distributed in a square area of side L, which is much larger than the communication range (r_{tx}) of the sensor nodes. Every node maintains a neighbor table $N(\cdot)$. Each sensor periodically produces information as it monitors its vicinity. Similar to [14], nodes with malfunctioning sensors are allowed to act as a communication node for routing. However, these nodes are asked to switch off their sensors. Only those sensor nodes with a permanent fault in the transceiver and power supply are to be removed from the network.

3.2. Energy Consumption Model. Similar to [26], this work assumes a simple model for the radio hardware energy dissipation where the transmitter dissipates energy to run the radio electronics and the power amplifier, and the receiver dissipates energy to run the radio electronics. Both the free space (d^2 power loss) and the multipath fading (d^4 power loss) channel models are used, depending on the distance between the transmitter and receiver. The energy spent for transmission of an r-bit packet over distance d is

$$E_{\mathrm{Tx}}(r,d) = rE_{\mathrm{elec}} + r\epsilon d^{\alpha} = \begin{cases} rE_{\mathrm{elec}} + r\epsilon_{\mathrm{fs}} d^2, & d < d_0, \\ rE_{\mathrm{elec}} + r\epsilon_{\mathrm{amp}} d^4, & d \geq d_0. \end{cases} \tag{1}$$

The electronics energy, E_{elec}, depends on factors such as the digital coding and modulation, whereas the amplifier energy, $\epsilon_{\mathrm{fs}} d^2$ or $\epsilon_{\mathrm{amp}} d^4$, depends on the transmission distance and the acceptable bit-error rate. To receive this message, the radio expends energy:

$$E_{\mathrm{Rx}}(r) = rE_{\mathrm{elec}}. \tag{2}$$

3.3. Fault Model. After a fault is activated, we consider that sensor nodes can either continue with the faulty behavior or alternate between periods of correct and faulty behavior. In the latter case, faults are observable for a time, which is termed as fault appearance duration (FAD), before they disappear. Eventually, faults may reappear either because of new transient faults or correlated intermittent faults [7]. The time duration during which fault disappears is termed as fault disappearance duration (FDD). Intermittent faults, after their first appearance, exhibit a high occurrence rate and eventually tend to become permanent [2]. For intermittent fault, the number of time units (sojourn time) that the node remains in the fault appearance state is less than or equals to the number of time units that was in the previous fault appearance state. We use this sojourn time to model subsequent failures of the sensor node over time. Similar to [7], a node is unhealthy if it has internal faults and fails in a permanent or intermittent manner. A node is healthy if it fails only on external intervention like electromagnetic radiations, and so forth. The proposed algorithm assumes that the sensor fault probability p is uncorrelated and symmetric; that is,

$$P(S = xA = \neg x) = P(S = \neg xA = x) = p, \tag{3}$$

where S is the sensor measurement (say temperature) and A is the actual ambient temperature.

3.4. Channel Model. The model used for channel is a two-state Gilbert-Elliott channel (two-state Markov channel model) [27, 28] with two states: G (good) state and B (bad) state. This model describes errors on the bit level. In the good state, the bits are received incorrectly with probability P_{good}, and in the bad state, the bits are received incorrectly with probability P_{bad}. For this model, it is assumed that $P_{\mathrm{good}} \ll P_{\mathrm{bad}}$. The transition probability $T_{\mathrm{GB}} = P(\mathrm{G} \rightarrow \mathrm{B})$ and $T_{\mathrm{BG}} = P(\mathrm{B} \rightarrow \mathrm{G})$ will be small, and the probability

(1) // Each node executes the algorithm at discrete time kT
(2) Broadcast the sensor reading x_i.
(3) set timer T_{out}.
(4) Obtain the sensed data of one-hop neighbors $N(v_i)$.
(5) **if** T_{out} = true **then**
(6) Declare unreported nodes as hard faulty and isolate the node.
(7) **end if**
(8) Determine $\{E\}$, the set of one-hop neighbors report identical sensor measurement x.
(9) **if** ($x_i = x$ and $|\{E\}| < \theta_1$) or ($x_i \neq x$ and $|\{E\}| \geq \theta_1$) **then**
(10) $F_{\text{state}_i} \leftarrow$ soft faulty.
(11) The node enters to observation state.
(12) **else**
(13) $F_{\text{state}_i} \leftarrow$ Fault-free.
(14) **end if**
(15) Broadcast the F_{state_i}.

Algorithm 1: Fault detection.

remaining in G and B is large. The steady-state probability of a channel being in the bad state is $P_{\text{B}} = T_{\text{GB}}/(T_{\text{GB}} + T_{\text{BG}})$. Thus, the average bit error probability of the channel is $P_{\text{cerr}} = P_{\text{bad}}P_{\text{B}} + P_{\text{good}}(1 - P_{\text{B}})$. For the simulations, this work uses this model that independently generates error patterns for all channels between nodes.

3.5. Definitions and Terminologies. The performance parameters used to measure the effectiveness of the proposed detection algorithm is as follows.

(1) *Detection accuracy (DA)* is defined as the number of faulty sensor nodes diagnosed by each node to the total number of faulty sensor nodes in the network.

(2) *False alarm rate (FAR)* is defined as the ratio of the number of fault-free sensor nodes diagnosed as faulty to the total number of fault-free nodes in the network.

(3) *Network lifetime* is the measure of the number of data-gathering rounds when the first node dies due to depletion of battery.

4. The Fault Detection Framework

The fault detection frame work consists of two phases, namely, fault detection phase and isolation phase. The fault detection phase exploits the fact that sensor faults are likely to be stochastically unrelated, while sensor measurements are likely to be spatially correlated. In WSNs, sensors from the same region should have recorded similar sensor readings [29]. For example, let v_i be a neighbor of v_j; x_i and x_j are the sensor readings of v_i and v_j, respectively. Sensor reading x_i is similar to x_j when $|x_i - x_j| < \delta$, where δ is application dependent. As an illustration, in bolt loosening monitoring, a sensor node and its neighbors are expected to have similar voltage. Similarly, in the case of temperature monitoring, a sensor node and its neighbors are expected to have similar temperature reading. Hence, δ is expected to be a small number. In the proposed approach, sensor nodes coordinate with their one-hop neighbors to detect faulty sensor nodes before conferring with the central node. Therefore, this design reduces communication messages, and subsequently, conserves sensor node energy. The isolation phase uses a count and threshold-based approach to isolate unhealthy nodes (see Algorithm 1).

4.1. Fault Detection Algorithm. In this approach, each node in the network broadcasts its sensor reading periodically by using a round-based message dissemination protocol. Upon receiving the sensor readings of one-hop neighbors, a node v_i constructs a set ($\{E\} \subset \{N(v_i)\}$) of nodes with similar reading S. The node v_i is detected fault-free if reading S_i agrees with S and the cardinality of set $\{E\}$ is greater than the threshold (θ_1). Otherwise, v_i is marked as soft faulty. The optimal value for θ_1 is $0.5\,(N - 1)$ [30], where N is the number of neighbors. The node v_i detects node $v_j \in N(v_i)$ as hard faulty, if v_i does not receive the sensor reading from v_j before T_{out}. T_{out} should be chosen carefully so that all the fault-free nodes $v_j \in N(v_i)$ must report node v_i before T_{out}. A node detected as soft faulty is not immediately isolated from the network. The node is allowed to take part in the network activities; however, the node is asked to switch off its sensor. This node next enters the observation stage. The node will be isolated if it is detected as faulty in the observation stage. Otherwise, it is reintegrated to the network and is asked to switch on its sensor. A description of fault detection is given in Algorithm 1 (see Algorithm 2).

4.2. Isolation of Unhealthy Nodes. A node detected as faulty for first time by Algorithm 1 enters to observation stage. This phase decides whether to isolate the node from the network (intermittent faulty) or to reintegrate the node to the network (fault free with transient fault). In this phase, the node under observation first initializes the penalty counter to one and the reward counter to zero. The node does not take any sensor reading. At the discrete time kT, it receives the sensor readings of its one-hop neighbors and executes Algorithm 1. If a fault appears and is detected at time

(1) Upon entering the observation state the node initializes the penalty counter z and reword counter r to one and zero respectively.
(2) At each discrete time kT a node executes Algorithm 1.
(3) **if** Detected as faulty **then**
(4) Reset reward counter, that is, $r = 0$.
(5) **if** $\text{FDD}_i \leq \text{FDD}_{i-1}$ **then**
(6) Increment the penalty counter by ξ, that is, $z = z + \xi$.
(7) **else**
(8) $z = z + 1$.
(9) **end if**
(10) **else**
(11) Increment the reward counter, that is, $r = r + 1$.
(12) **end if**
(13) **if** $r > \theta_2$ and $z < \theta_3$ **then**
(14) Node is reintegrated.
(15) **else**
(16) Node is isolated.
(17) **end if**

Algorithm 2: Observation state.

Table 1: Simulation parameters.

Parameter	Value
Number of sensors	1000
Network grid	From $(0, 0)$ to $(600, 400)$ m
Sink	At $(620, 200)$ m
Initial energy	1 J
Data packet size	4000 bits
E_{elec}	50 nJ/bit
ϵ_{fs}	10 pJ/bit/m^2
ϵ_{amp}	0.0013 pJ/bit/m^4

kT, subsequently it first reset the reward counter. Second, it checks the present fault disappearance duration (FDD_i) with the preceding fault disappearance duration FDD_{i-1}. If $\text{FDD}_i \leq \text{FDD}_{i-1}$, subsequently the penalty counter is incremented by a factor equals to ξ. If $\text{FDD}_i > \text{FDD}_{i-1}$, then penalty counter is incremented by a factor equals to one. This is because intermittent faults usually exhibit a relatively fast occurrence rate. If the penalty counter exceeds its threshold (θ_3), the node is isolated from the network. Similarly, if the reward counter exceeds its threshold (θ_2), the node is reintegrated to the network.

5. Simulation Experiments

The performance of the proposed scheme through simulations is presented in this section. This work uses Castalia-2. 3b [31], a state-of-the-art WSN simulator based on the OMNET++ [32] platform. The simulation parameters are given in Table 1. For these simulations, energy is consumed whenever a sensor transmits or receives data or performs data aggregation.

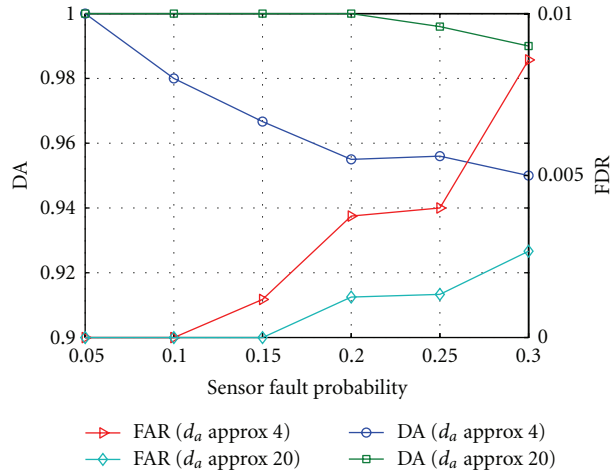

Figure 1: DA and FAR at varying fault rate.

5.1. Experiment 1: DA and FAR in Regard to d_a and p. In this experiment, the performance of the diagnosis algorithm in regard to DA and FAR is evaluated. In this simulation, sensor nodes are assumed to be faulty with probabilities of 0.05, 0.10, 0.15, 0.20, 0.25, and 0.30, respectively. The range is chosen for the sensor network to have the desired average node degree (d_a). In this experiment, faults are assumed to be permanent. Since a faulty node will often report unusually high- or low-sensor measurements, all the nodes with malfunctioning sensors are momentarily assumed to show a match in comparison with a probability of 0.5 regardless of their locations.

The DA and FAR at varying fault rate and average node degree are shown in Figure 1. A high level of DA (>0.96) is reported even when fault rate is as high as 0.2 and $d_a \approx 4$

Table 2: Design and system parameters and their nominal values.

Parameter	Description	Nominal value
θ_2	Reward threshold	10^4
θ_3	Penalty threshold	5
ξ	Penalty increment	2
T	Intertest interval	8 sec
FAD	Continuous distribution of fault appearance duration	Exponential
E [FAD]	Expected fault appearance duration	5 ms
FDD_u	Continuous distribution of fault disappearance duration of unhealthy node	Weibull ($\alpha = 1.4$)
$E[\text{FDD}]_u$	Expected fault disappearance duration of unhealthy node	1 h
FDD_h	Continuous distribution of fault disappearance duration of healthy node	Exponential
$E[\text{FDD}]_h$	Expected fault disappearance duration of healthy node	100 h

(i.e., sparse network). This is because the detection algorithm wrongly detects a fault-free node as faulty only when it has more than θ_1 number of faulty one-hop neighbors. In addition, the proposed approach uses an optimal value for θ_1. As expected, an improvement in DA is observed for higher value of d_a. A very low level (<0.004) of FAR is reported for $d_a \approx 4$ even when fault rate is as high as 0.2. The reason is that a faulty node is detected as fault free only when it has more than θ_1 number of faulty one-hop neighbors, and all produces the same faulty reading. The probability of the mentioned number is very low and decreases for an increase in d_a.

5.2. Experiment 2: Parameter Tuning. There are several design parameters in the proposed approach, namely, T, z, r, and ξ. In this experiment, we tune these parameters with regard to the accuracy, coverage, and detection latency. In this experiment, we have deployed 100 faulty nodes randomly ($P = 0.1$). Each faulty node can exhibit the permanent, intermittent, and transient fault with probability 1/3. While conducting sensitivity analysis on each design parameter, we fix the others to the nominal values as summarized in Table 2. The transmission range of each node is chosen to have $d_a \approx 20$. This ensures that a fault is detected by a test (execution of the Algorithm 1) if it appears at the time of test. In addition, larger value of d_a ensures low FAR (refer to Figure 1). However, this restriction in node degree is relaxed in subsequent experiments to observe the performance of the proposed approach in sparse networks.

A The data-gathering stage is scheduled at GT where G is an integer and is application specific. For instance, applications with short mission time need the data to be gathered more frequently in contrast to applications, where frequency of data gathering is less. For applications with long mission time, GT is large. Thus, to discriminate transient from intermittent faults, G/T number of sensor measurements needs to be broadcasted by each node. This in turn make the packet to grow with G. Since energy consumed by a sensor node is directly proportional to the number of bits it transmits or receives, the energy overhead will be more for large value of G and may not be practically implementable. To address this issue, we suggest to sample the interval GT where each sample constitutes of I consecutive sensor measurements. The standard deviations of these I sensor measurements correspond to each sample interval are calculated and broadcasted along with the routine data. This in turn reduces the packet size and makes the algorithm energy efficient. Each node takes the decision by comparing the corresponding standard deviations of one-hop neighbors. Use of standard deviation instead of individual measurements does not affect the detection performance since rate of change in sensor measurements over time is very low. In addition, a sensor often reports unusually high- or low-sensor measurement during FAD. Thus, the standard deviation of sensor measurements of a sample interval with at least one incorrect measurement will be distinguished from the corresponding standard deviations of one-hop neighbors with all true measurements. In this experiment, we assume temperature sensors.

Figure 2(a) depicts the average accuracy and coverage at varying values of T. This result confirms that T has a strong impact on average accuracy. It is observed that the average accuracy falls after $T = 94$ sec. This is because when T is excessively long, an excessively long time is required to reach the reward threshold. For instance, this time for $T = 20$ sec and $\theta_2 = 10^4$ is 55.56 hours. The mentioned period of correct operation is too long and increases with T. Thus, the occurrence of subsequent transient faults will be viewed as correlated intermittent faults and the node will be isolated. It is observed that the average coverage remains unaffected by change in T. However, as shown in Figure 2(b), the average latency of isolation increases with T. The reason is that if T is too high, then probability that the FAD coincides within

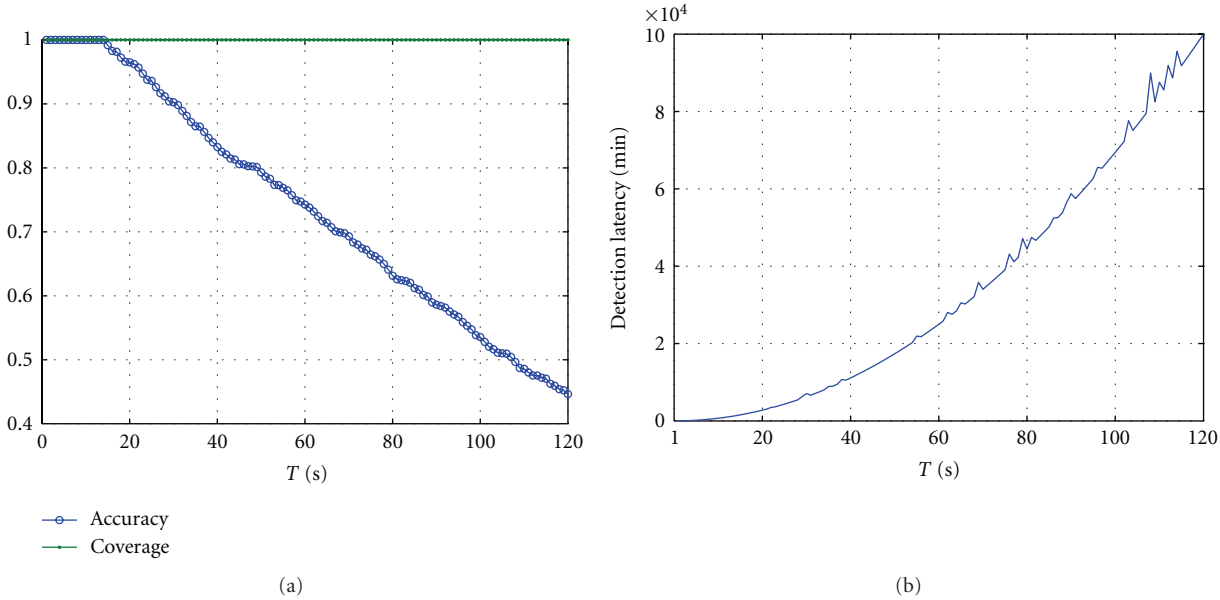

Figure 2: (a) Accuracy and coverage at varying value of T. (b) Detection latency at varying value of T.

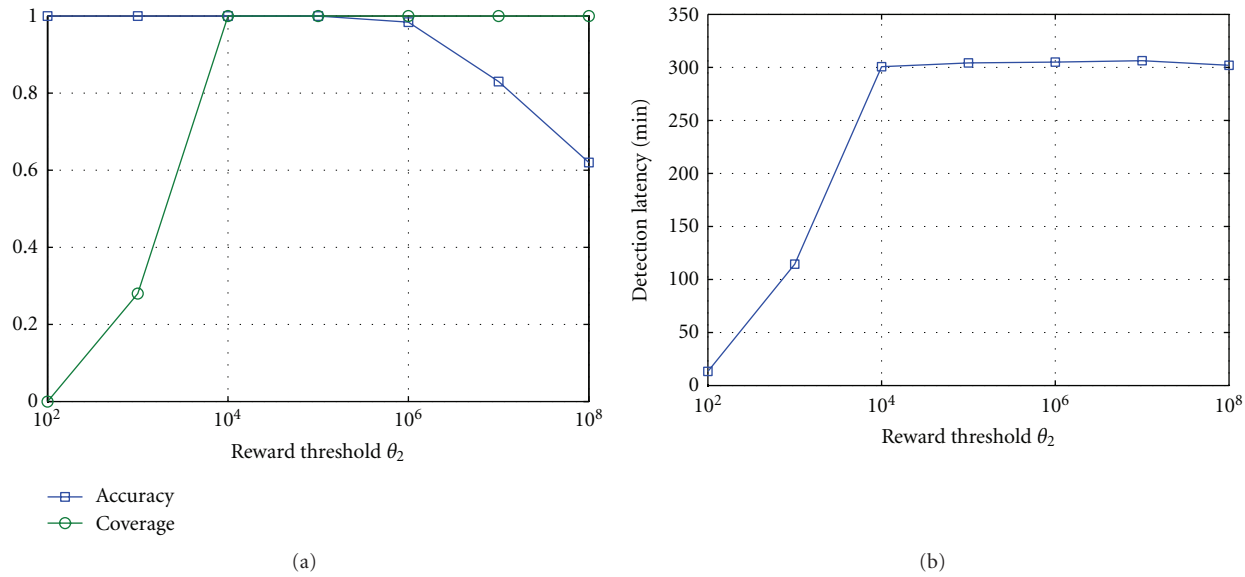

Figure 3: (a) Accuracy and coverage at varying value of θ_2. (b) Detection latency at varying value of θ_2.

the intertest interval is high. This means, the probability that a fault may appear after kT and subsequently disappears before $(k+1)T$ increases with T. This in turn increases the number of test repetitions and thus the average latency of isolation to reach the penalty threshold.

The impact of the reward threshold θ_2 on the average accuracy and coverage is depicted in Figure 3(a). In the proposed detection algorithm, a node is isolated if it fails before reaching the reward threshold. If θ_2 is too large, then a healthy node enters the observation stage may be isolated. The reason is that transient faults appear to be correlated intermittent faults. This in turn affects the accuracy. If θ_2 is too small, then intermittent faults will be treated as transient faults and will be reintegrated to the network causing poor coverage. This is because the value of θ_2 must be greater than the average number of test receptions required to detect the presence of fault. Thus, proper tuning of θ_2 is crucial to achieve a good discrimination. The best tradeoff for the given scenario is observed at $\theta_2 = 10^4$. The time period of correct operation for $\theta_2 = 10^4$ (8×10^4 sec $\approx$ 1333 minutes) is adequate for an unhealthy node to reach the penalty threshold. In addition, as shown in Figure 3(b), accuracy is 100%; that is, the transient faults do not appear as correlated intermittent faults for $\theta_2 = 10^4$. The average detection

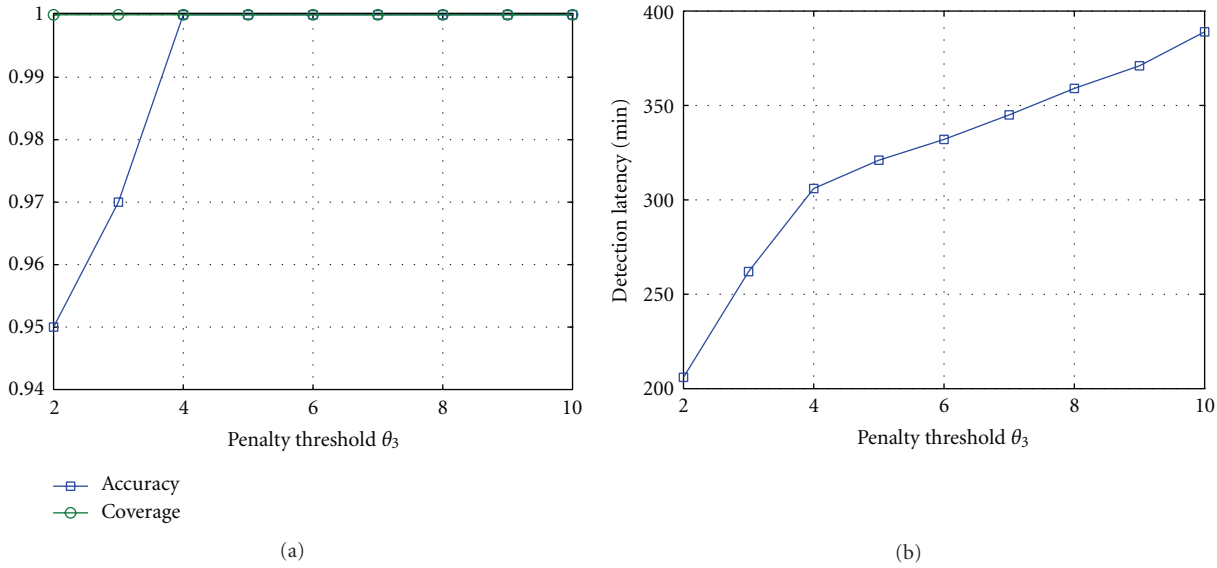

FIGURE 4: (a) Accuracy and coverage at varying value of θ_3. (b) Detection latency at varying value of θ_3.

latency for varied values of θ_2 is shown in Figure 3(b). The average latency of isolation is reported almost unaffected for $\theta_2 \geq 10^4$. This is because 100% coverage is reported for $\theta_2 \geq 10^4$. M detection latency depends only on T and the number of test repetitions required to reach θ_3. Thus, for $\theta_2 \geq 10^4$, the detection latency is negligibly affected.

Figure 4(a) shows the coverage and accuracy at varying value of penalty threshold. As discussed earlier, the penalty counter is incremented by a value ξ if the present FDD is smaller than the preceding FDD. For smaller value of θ_3, the probability of isolation of healthy nodes in the observation state is more as the transient faults are appeared to be correlated intermittent faults. As expected and shown in Figure 4(a), the average coverage is not affected by varying values of θ_3. As shown in Figure 4(b), the average latency of isolation increases with θ_3. This is because the number of test repetitions required to detect the presence of fault for θ_3 time increases with θ_3. Since the proposed approach implements an adaptive penalty increment technique and a relatively high fault occurrence rate is observed in an unhealthy node, the average detection latency grows less after $\theta_3 = 5$.

Finally, we study the effect of ξ on the average detection latency, the average number of test repetitions, the average coverage, and average accuracy. When ξ is set to 1, the proposed algorithm acts similar to the approach proposed in [7] that does not consider the fault disappearance state holding time. Figure 5(a) illustrates the improvement of the detection latency with ξ. When ξ is greater than 2, the detection latency is lower than that of the circumstance when $\xi = 1$. Similarly, Figure 5(b) illustrates the improvement of the number of test repetitions required to discriminate transient from intermittent faults. When ξ is larger than 2, the number of test repetitions is lower than that of the circumstance when $\xi = 1$. The effect of ξ on average accuracy and coverage is depicted in Figure 5(c). A tradeoff is observed where both the average accuracy and coverage attain their highest value form $\xi = 2$ to $\xi = 3$. These results suggest the importance of ξ in discriminating transient from intermittent faults.

In summary, for wireless sensor networks, a setting of $T = 8$ sec, $\theta_2 = 10^4$, $\theta_3 = 5$, and $\xi = 2$ allows to discriminate most of the transient from intermittent faults.

5.3. Experiment 3: Robustness with regard to Transient Faults. In this experiment, we estimate how well the proposed detection algorithm discriminates transient from intermittent faults. We compare the performance of the detection algorithm with the sate-of-art detection algorithm proposed by Lee et al. in [14]. Similar to [14], we redefine FAR as follows. Let n_g, n_t, and n_f represent the number of good nodes, the number of good nodes with a transient fault, and the number of faulty nodes, respectively. Let n_{gf} be the number of nodes wrongly detected as faulty out of the n_g good nodes. Similarly, the number of healthy nodes with a transient fault identified as faulty is denoted by n_{tf}. The FAR is redefined as $(n_{gf} + n_{tf})/(n_g + n_t)$. For better performance evolution, we consider the equal number of permanent and intermittent faults. In this experiment, the impact of transient fault rates (p_t) on DA and FAR has been evaluated for $P = 0.5, 0.10, 0.15$, and 0.20.

As expected and shown in Table 3, the detection accuracy is less affected by varying rate of transient faults in the network. This is because the proposed detection algorithm wrongly detects a faulty node as fault free if the node has more than θ_1 number of faulty neighbors during kth test at time kT and all are reporting the same faulty reading. The probability of the mentioned number is very less due to the following reasons. (1) In this approach, all the nodes with malfunctioning sensors are momentarily assumed to show a match in comparison with a probability of 0.5 regardless of their locations. (2) The appearance of intermittent and transient faults are random in nature. At the time of the kth

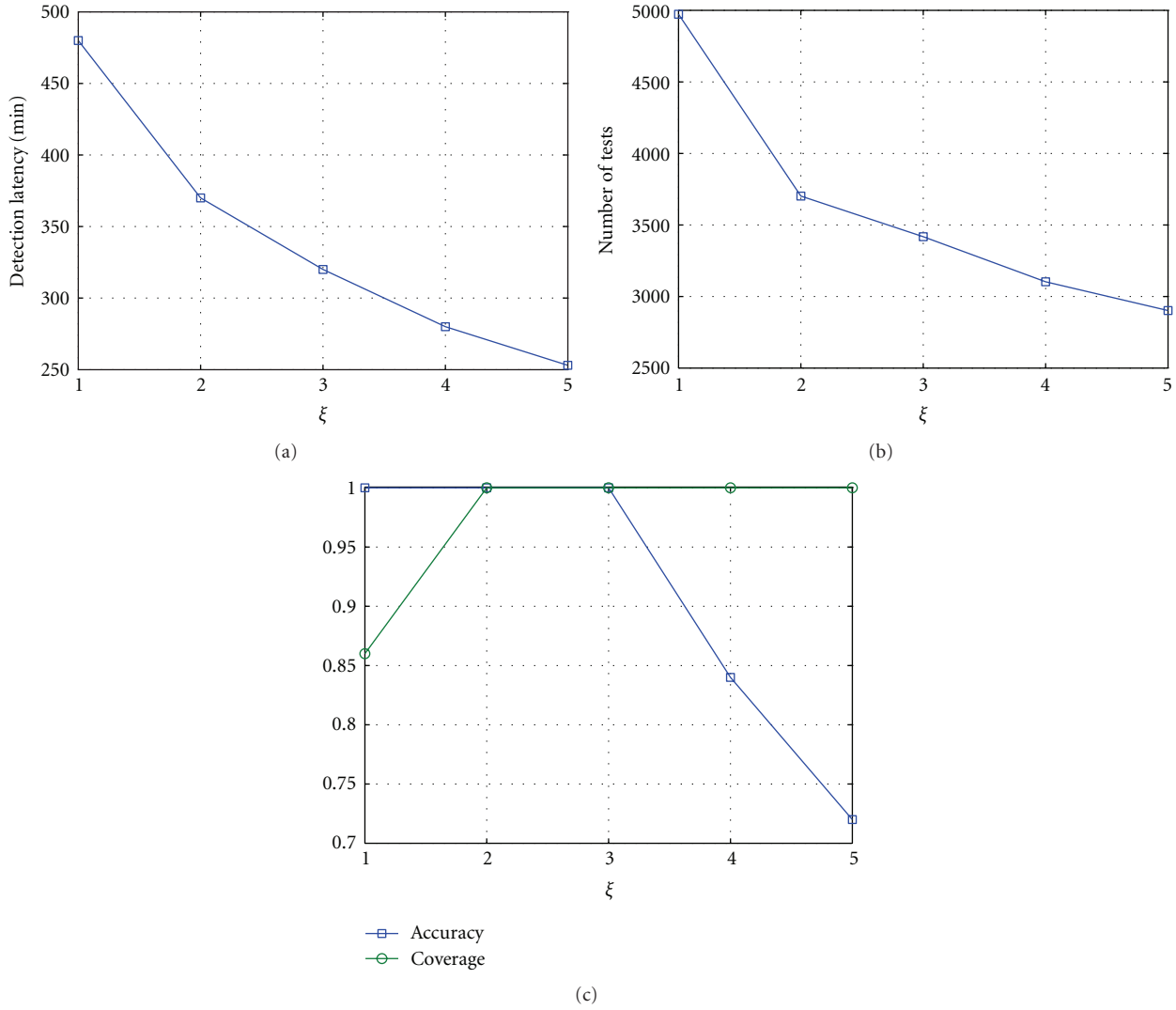

FIGURE 5: (a) Detection latency at varying value of ξ. (b) Average number of tests at varying value of ξ. (c) Accuracy and coverage at varying value of ξ.

test, the probability that all the transient and intermittent fault appears is very low. (3) Simulation results shown in Experiment 1 confirm a better performance in sparse networks. Similar to the proposed detection algorithm, the detection accuracy of the detection algorithm proposed by Lee et al. is less sensitive to change in p_t.

The robustness of the fault detection algorithm to transient faults from FAR perspective is shown in Table 4. As expected the FAR is less affected by varying rate of transient faults in the network. In the proposed approach, a fault-free node detected as faulty only when it has more than θ_1 number of faulty neighbors during kth test. As discussed, the probability of the said number is very low. In addition, proper tuning of detection parameters ensures efficient discrimination of transient from intermittent faults. The fault-free nodes with transient faults are effectively reintegrated into the network which in turn keeps the FAR low. As reported in Table 4, the proposed detection algorithm outperforms Lee's approach from FAR perspectives. The reason is that the two thresholds used in Lee's scheme are not adequate to discriminate transient from intermittent or permanent faults. Thus, their approach isolates the maximum number of fault-free nodes with transient faults.

5.4. Experiment 4: Robustness with Regard to Channel Fault. In this experiment, the robustness of the detection algorithm to faults in the communication channel is analyzed by estimating DA and FAR for various channel error probabilities. In this experiment, we set $p_t = 0.2$. For simplicity in the simulation, P_{good} is taken as 0 and P_{bad} is taken as 1. P_{BG} is fixed to 1/8, and P_{GB} is varied to get different channel error probabilities P_{cerr}. The channel error rate is increased in steps from 10^{-4} to 10^{-1}. Faults in the communication channel might cause some fault-free nodes to fail in receiving the sensor measurements from its neighbors. This in turn decreases the effective neighbor size of a sensor node and might affect

Table 3: Detection accuracy in presence of transient faults: $p_c = 10^{-3}$.

p	$p_t = 00$	$p_t = 0.05$	$p_t = 0.1$	$p_t = 0.15$	$p_t = 0.20$
Lee and Choi [14]					
$d_a \approx 4$					
0.05	1.000	1.000	1.000	0.980	0.980
0.10	0.980	0.980	0.970	0.960	0.960
0.15	0.953	0.953	0.947	0.940	0.933
0.20	0.946	0.933	0.933	0.920	0.906
$d_a \approx 20$					
0.05	1.000	1.000	1.000	1.000	1.000
0.10	1.000	1.000	0.990	0.990	0.980
0.15	1.000	1.000	0.973	0.967	0.946
0.20	1.000	1.000	0.970	0.955	0.930
Proposed					
$d_a \approx 4$					
0.05	1.000	1.000	1.000	1.000	1.000
0.10	0.980	0.980	0.980	0.970	0.960
0.15	0.967	0.967	0.960	0.947	0.940
0.20	0.955	0.955	0.945	0.930	0.925
$d_a \approx 20$					
0.05	1.000	1.000	1.000	1.000	1.000
0.10	1.000	1.000	1.000	0.990	0.990
0.15	1.000	1.000	1.000	0.987	0.987
0.20	1.000	1.000	0.990	0.980	0.975

Table 4: False alarm rate in presence of transient faults: $p_c = 10^{-3}$.

p	$p_t = 00$	$p_t = 0.05$	$p_t = 0.1$	$p_t = 0.15$	$p_t = 0.20$
Lee and Choi [14]					
$d_a \approx 4$					
0.05	0.0053	0.0105	0.0158	0.0326	0.0463
0.10	0.0100	0.0213	0.0350	0.0425	0.0563
0.15	0.0147	0.0341	0.0494	0.0624	0.0765
0.20	0.0200	0.0425	0.0587	0.0737	0.0887
$d_a \approx 20$					
0.05	0.0000	0.0095	0.0136	0.0284	0.0432
0.10	0.0000	0.0200	0.0311	0.0389	0.0533
0.15	0.0000	0.0282	0.0413	0.0537	0.0712
0.20	0.0000	0.0388	0.0525	0.0633	0.0838
Proposed					
$d_a \approx 4$					
0.05	0.0000	0.0000	0.0000	0.0000	0.0011
0.10	0.0000	0.0000	0.0011	0.0022	0.0044
0.15	0.0012	0.0012	0.0024	0.0047	0.0088
0.20	0.0037	0.0050	0.0063	0.0088	0.0125
$d_a \approx 20$					
0.05	0.0000	0.0000	0.0000	0.0000	0.0000
0.10	0.0000	0.0000	0.0000	0.0000	0.0011
0.15	0.0000	0.0000	0.0000	0.0012	0.0035
0.20	0.0000	0.0000	0.0013	0.0050	0.0075

Table 5: Detection accuracy in presence channel faults: $p_t = 0.2$.

p	$p_c = 10^{-4}$	$p_c = 10^{-3}$	$p_c = 10^{-2}$	$p_c = 10^{-1}$
Lee and Choi [14]				
$d_a \approx 4$				
0.05	0.980	0.980	0.980	0.960
0.10	0.960	0.960	0.940	0.920
0.15	0.933	0.926	0.926	0.900
0.20	0.900	0.906	0.890	0.875
$d_a \approx 20$				
0.05	1.000	1.000	1.000	0.980
0.10	1.000	0.980	0.980	0.950
0.15	0.953	0.946	0.940	0.913
0.20	0.940	0.930	0.925	0.905
Proposed				
$d_a \approx 4$				
0.05	1.000	1.000	1.000	0.960
0.10	0.960	0.960	0.950	0.930
0.15	0.940	0.940	0.933	0.907
0.20	0.935	0.925	0.915	0.890
$d_a \approx 20$				
0.05	1.000	1.000	1.000	0.980
0.10	1.000	0.990	0.990	0.960
0.15	0.980	0.973	0.967	0.953
0.20	0.965	0.965	0.950	0.945

Table 6: False alarm rate in presence channel faults: $p_t = 0.2$.

p	$p_c = 10^{-4}$	$p_c = 10^{-3}$	$p_c = 10^{-2}$	$p_c = 10^{-1}$
Lee and Choi [14]				
$d_a \approx 4$				
0.05	0.0463	0.0463	0.0463	0.0495
0.10	0.0563	0.0563	0.0589	0.0656
0.15	0.0729	0.0765	0.0800	0.0882
0.20	0.0862	0.0887	0.0900	0.0975
$d_a \approx 20$				
0.05	0.0432	0.0432	0.0432	0.0453
0.10	0.0533	0.0533	0.0544	0.0578
0.15	0.0694	0.0712	0.0729	0.0765
0.20	0.0813	0.0838	0.0862	0.0925
Proposed				
$d_a \approx 4$				
0.05	0.0011	0.0011	0.0021	0.0053
0.10	0.0033	0.0044	0.0067	0.0089
0.15	0.0082	0.0088	0.0094	0.0129
0.20	0.0113	0.0125	0.0187	0.0238
$d_a \approx 20$				
0.05	0.0000	0.0000	0.0000	0.0011
0.10	0.0000	0.0011	0.0022	0.0044
0.15	0.0024	0.0035	0.0047	0.0082
0.20	0.0063	0.0075	0.0088	0.0125

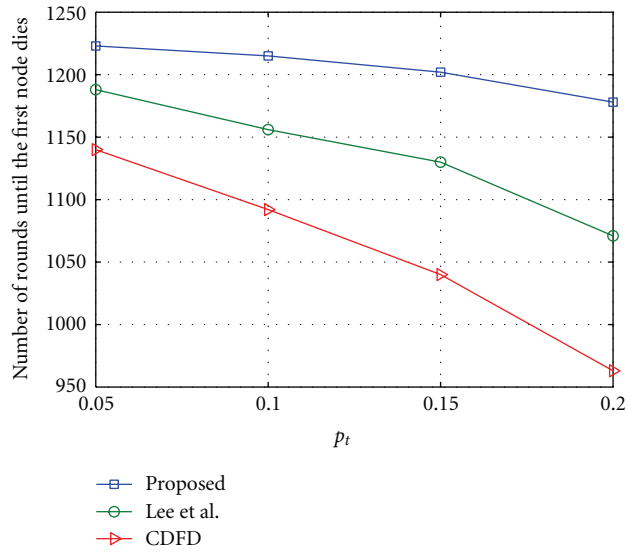

Figure 6: Network lifetime.

the local decision. However, as discussed in Experiment 1, the detection algorithm shows better performance even in sparse networks. Thus, as expected and shown in Tables 5 and 6, the detection algorithm effectively tolerates faults in the communication channel. It is observed that the detection scheme proposed in [14] effectively tolerates faults in the communication channel.

5.5. Experiment 5: Network Lifetime. In this experiment, we evaluate the energy efficiency of the proposed detection algorithm and compare it with Lee's approach. Algorithm 1 has been implemented excluding the observation stage (line 11) and is named as the conventional method. We consider an example network where all sensor nodes are assumed to be fault free or fault free with transient faults. In this simulation, the sensor nodes are assumed to have transient faults with probability 0.05, 0.10, 0.15 and 0.20 respectively. A node is considered dead if it has lost 99 percent of its initial energy. As expected and shown in Figure 6, the proposed detection algorithm outperforms both Lee's approach and the conventional approach. This is because FAR of Lee's approach is worst affected by the increase in transient fault rate. Thus, their approach isolates fault-free nodes with transient faults. This in turn increases the workload of each node in the network, and the nodes depletes energy faster. In contrary, the proposed detection algorithm keeps FAR low and is less sensitive to change in transient fault rate. The conventional method isolates a node when the fault appears and detected by it, thus reducing available resources and impacting network lifetime.

6. Conclusions

This paper has presented a simple distributed fault detection algorithm for wireless sensor networks where permanent, intermittent, and transient faults have been considered. The detection parameters, namely, intertest interval, reward counter, and penalty counter were tuned to effectively discriminate the persistence (permanent, intermittent, and transient) of faults in wireless sensor networks. An adaptive penalty increment is suggested to reduce the detection latency. The performance of the detection algorithm was compared to the state of art approach proposed by Lee and Choi [14]. The simulation results show that the proposed detection algorithm outperforms Lee's approach from FAR perspective. Since the proposed protocols will be used in WSNs, which are known to be energy limited, it would be preferable for a proposed protocol to be efficient from FAR perspective. One of the advantages of the proposed work is that the diagnosis is not considered as an offline but online core fault tolerant mechanism integrated to the WSN fault tolerant strategy.

References

[1] B. Selic, "Fault tolerance techniques for distributed systems," July 2004, http://www.ibm.com/developerworks/rational/library/114.html.

[2] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, and F. Grandoni, "Threshold-based mechanisms to discriminate transient from intermittent faults," *IEEE Transactions on Computers*, vol. 49, no. 3, pp. 230–245, 2000.

[3] D. P. Siewiorek and R. S. Swmlz, *Reliable Computer System Design and Evaluation*, Digital Press, 1992.

[4] M. Barborak, A. Dahbura, and M. Malek, "Consensus problems in fault-tolerant computing," *ACM Computing Surveys*, vol. 25, no. 2, pp. 171–220, 1993.

[5] R. Horst, D. Jewett, and D. Lenoski, "The risk of data corruption in microprocessor-based systems," in *Proceedings of the 23rd International Symposium on Fault-Tolerant Computing*, pp. 576–585, June 1993.

[6] A. Avižienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing,"

IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, pp. 11–33, 2004.

[7] M. Serafini, A. Bondavalli, and N. Suri, "Online diagnosis and recovery: on the choice and impact of tuning parameters," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 4, pp. 295–312, 2007.

[8] M. Malek, "A comparison connection assignment for diagnosis of multiprocessor systems," in *Proceedings of the 7th Annual Symposium on Computer Architecture (ISCA '80)*, pp. 31–36, ACM, 1980.

[9] H. W. Brown and D. M. Blough, "The broadcast comparison model for on-line fault diagnosis in multicomputer systems: theory and implementation," *IEEE Transactions on Computers*, vol. 48, no. 5, pp. 470–493, 1999.

[10] G. M. Megson, X. Yang, and D. J. Evans, "A comparison-based diagnosis algorithm tailored for crossed cube multiprocessor systems," *Microprocessors and Microsystems*, vol. 29, no. 4, pp. 169–175, 2005.

[11] Y. Y. Tang and X. Yang, "Efficient fault identification of diagnosable systems under the comparison model," *IEEE Transactions on Computers*, vol. 56, no. 12, pp. 1612–1618, 2007.

[12] Y. S. Chen and S. Y. Hsieh, "Strongly diagnosable product networks under the comparison diagnosis model," *IEEE Transactions on Computers*, vol. 57, no. 6, pp. 721–732, 2008.

[13] G. Y. Chang, "(t, k)-diagnosability for regular networks," *IEEE Transactions on Computers*, vol. 59, no. 9, pp. 1153–1157, 2010.

[14] M. H. Lee and Y. H. Choi, "Fault detection of wireless sensor networks," *Computer Communications*, vol. 31, no. 14, pp. 3469–3475, 2008.

[15] S. Chessa and P. Santi, "Crash faults identification in wireless sensor networks," *Computer Communications*, vol. 25, no. 14, pp. 1273–1282, 2002.

[16] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS '06)*, pp. 65–71, ACM, September 2006.

[17] P. Jiang, "A new method for node fault detection in wireless sensor networks," *Sensors*, vol. 9, no. 2, pp. 1282–1294, 2009.

[18] C. Hsin and M. Liu, "Self-monitoring of wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 462–476, 2006.

[19] X. Miao, K. Liu, Y. He, Y. Liu, and D. Papadias, "Agnostic diagnosis: discovering silent failures in wireless sensor networks," in *Proceedings of IEEE (INFOCOM '11)*, pp. 1548–1556, April 2011.

[20] S. Guo, Z. Zhong, and T. He, "Find: faulty node detection for wireless sensor networks," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 253–266, ACM, New York, NY, USA, November 2009.

[21] J. L. Gao, Y. J. Xu, and X. W. Li, "Weighted-median based distributed fault detection for wireless sensor networks," *Journal of Software*, vol. 18, no. 5, pp. 1208–1217, 2007.

[22] 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, 2004.

[23] M. Elhadef, A. Boukerche, and H. Elkadiki, "A distributed fault identification protocol for wireless and mobile ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 68, no. 3, pp. 321–335, 2008.

[24] J. Y. Choi, S. J. Yim, Y. J. Huh, and Y. H. Choi, "A distributed adaptive scheme for detecting faults in wireless sensor networks," *WSEAS Transactions on Communications*, vol. 8, no. 2, pp. 269–278, 2009.

[25] A. Weber, A. Kutzke, and S. Chessa, "Energy-aware test connectionassignment for the self-diagnosis of a wireless sensor network," *Journal of the Brazilian Computer Society*, vol. 18, no. 1, pp. 19–27, 2012.

[26] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[27] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell System Technical Journal*, vol. 39, pp. 1253–1265, 1960.

[28] E. O. Elliott, "Estimates of error rates for codes on burst error channels," *Bell System Technical Journal*, vol. 42, pp. 1977–1997, 1963.

[29] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Computer Networks*, vol. 45, no. 3, pp. 245–259, 2004.

[30] A. Mahapatro and P. M. Khilar, "Detection of node failure in wirelessimage sensor networks," *ISRN Sensor Networks*, vol. 2012, pp. 1–8, 2012.

[31] A. Boulis, *Castalia: A Simulator For Wireless Sensor Networks and Body Area Networks*, National ICT Australia, 2009.

[32] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, pp. 1–10, 2008.