Improving Network Performance in Wireless Sensor Network Using Self-Maintenance Clustering Model

Walaa M. Elsayed(1), Sahar F. Sabbeh(1), and Alaa M. Riad(2)

(1) Information Systems Dept., Faculty of Computer Science & Information Systems, Banha University, Egypt

(2) Information Systems Dept., Faculty of Computer Science & Information Systems, Mansoura University, Egypt

Received 21 December 2016 - Accepted 28 March 2017

ABSTRACT

In Wireless Sensor Networks (WSNs), fault tolerance of a sensor node is a demanding issue since sensors are usually deployed in unattended environments. Limited memory, processing power, and communication range of sensor nodes make conventional fault tolerance schemes infeasible for WSNs. This work introduces a distributed self-healing methodology in which the detection, diagnosis and healing processes were performed at both node and cluster head levels. At node level, battery, sensor and receiver faults were diagnosed. At cluster head level, transmitter and mal-functional nodes were detected and recovered. The simulation results showed that the proposed methodology is precise in locating malfunctioning nodes and fast in finding a cover for such nodes.

Key Words: Clustering Model, Node Failure, Self-Healing, WSNs.

INTRODUCTION

WSNs have recently received a lot of attention from researchers all over the world. The term “Self-healing” denotes the capability of a software system to examine, diagnose, and react to system malfunctions. Self-healing components must be able to observe system failures, evaluate constraints imposed by the outside, and apply appropriate corrections (Liu *et al*., 2009). Self-healing is a critical solution to deal with dynamic systems, imprecise specification, uncontrolled environment, and reconfigure systems, according to their dynamics for enhancing the significant performance of WSN to recover such systems for delivering a desired well functionality level, and avoids the presence of such faults.

Fault tolerance of sensor nodes clustering is a demanding issue, especially in unattended environments. Hence, forming sensor clusters is considered an effective way to increase flexibility and longevity of wireless sensor networks. The cluster head (CH) within each sensor cluster is usually responsible for data aggregation and consume more energy than the member nodes causing early termination due to energy exhaustion. Furthermore, fault detection, diagnosis, and recovery of faulty nodes demand more computation power, thus it shorten the network life.

To address the aforementioned challenges, many methods have been developed to detect, diagnose, and recover faulty nodes. The bulk of fault detection techniques in WSNs are usually conducted using a centralized or a distributed methodology. Centralized methodology is the most common methodology to detect and diagnose node failures caused by anomalous data reading, which occurs through monitoring process, misbehavior of sensor node components or environmental events. In centralized fault detection, each sensor node collects its reading periodically and sends it in a packet over its radio to the central base node, which is responsible for identifying the faulty sensor nodes in WSN. In distributed fault detection, the sensor nodes themselves monitor their own conditions and detect misbehavior by themselves or their neighbors. In addition, they exploit spatial correlation of environmental events to distinguish unusual (anomalous) events from faults. Sensor nodes can snoop on their neighbors’ packets and compare them with their own readings.

The contribution of this paper is: 1) Suggesting a distributed self-healing methodology (SHM) that includes four phases. The first phase is the "monitoring phase". During this phase, system's monitor will inspect the environment for any improper conduct. In the second phase "Error detection and diagnosis", the analysis of gathered data and key performance identifiers (KPI) are used to diagnose system components. Then, the root caused the failure can be identified and the healing method can be determined in third phase "Healing methodology selection". Finally, in the fourth phase "Execute repair operation", the selected recovery plan is executed and all required repairs will be completed. 2) Application of the suggested SHM, as a diagnostic tool, to identify the different types of hardware parameters failures in the sensor node in WSN, and 3) Practical Evaluation of the suggested SHM methodology in detecting failure, diagnose, and recover such failure.

This paper is organized as follows: Related works of fault detection techniques in WSNs is presented in the following section, while the proposed distributed self-healing methodology section describes the proposed approach to overcome WSNs failures. The experimental results are discussed in final section.

Related Works

For an efficient comprehension of self-healing methodology, it is important to clarify the difference between faults, errors, and failures. A fault is any kind of defect that leads to an error. An error corresponds to an incorrect (undefined) system state; such state may lead to a failure. A failure is the observation of an error, which occurs when the system deviates from its specification and cannot deliver its intended functionality (Rodrigues *et al*., 2013). In this field, many self-healing techniques are used to detect, diagnose and recover faulty nodes. Mamta and Singh (2014) improved Distributed Fault Detection (DFD) scheme for detecting status of each sensor node in WSN whether good or faulty, the detection is based on readings of the neighboring nodes. The authors used the false alarm rate for detecting the fault detection accuracy, in order to detect intermittently faulty sensor nodes and to stringent power budget during fault diagnosis process on sensor nodes in wireless sensor network. In 2014, Banerjee *et al.* proposed an Effective Fault Detection and Routing (EFDR) scheme for WSNs. EFDR is a centralized data routing schema capable of detecting hardware failures via neighbor node's temporal and spatial correlation of sensing information. Data routing scheme used L-system rules to determine optimal routing path between cluster head and base station (sink) (Banerjee *et al*., 2014). Lau *et al.* proposed a centralized hardware fault detection methodology for a structured WSN based on Centralized Naïve Bayes Detector (CNBD) to maximize the network's life. CNBD analyzed the collected data at the sink via end-to-end transmission time, in order to conserve higher power of the battery of each sensor node (Lau *et al*., 2014). Saihi *et al*.(2014) used centralized fault detection method based on clustering methodology, by exchanging heartbeat messages in active manner, for building scalable and energy balanced applications for WSNs.

Additionally, In February 2015, Dhumale *et al.* proposed a fault diagnostics framework composed of a pattern recognition system, having machine learning technology as its integral part is utilized for failure detection of different switches and tracing multiple types of faults in an inverter. This work is carried out to detect faults and classify the switches that cause the fault in an inverter and diagnosing breakdown, to make it possible to run an emergency operation in case of a fault. The authors have utilized Discrete Wavelet Transform (DWT) and Fuzzy Inference Logic (FIL) to process the generated signal (Dhumale *et al*., 2015). In October 2015, Diongue and Thiare provided a decentralized self-healing mechanism based on probabilistic sentinel scheme for reducing energy consumption in the wireless sensor networks. The proposed solution was based on two main algorithms; the first algorithm used node adaptation technique for detecting activity nodes and sleep nodes. On another beside, second algorithm discussed a link control adaptation to ensure better connectivity between sentinel nodes (Diongue and Thiare. 2015).

Generally, energy depletion is malfunction that can lead WSNs into misbehaving and affect the functional performance of the network. Therefore, it tops concerns of the recent literature review. Table (1) shows summery of the previous works that handled solutions for various WSN challenges in this field. In January 2016, Tapas Bapu and Siddanna Gowd presented a fault prediction technique to recover battery drain and estimate the network lifetime. They used backup module during self-healing operation to detect functionality fault (Tapas Bapu and Siddanna Gowd, 2016). In February 2016, Thangaraj and Anuradha studied a method for consolidating all the initiatives of energy drain and integrating these models in a panel framework in order to help manage the energy in WSN and support the energy engineering processes involved in WSN. Thangaraj and Anuradha designed an energy economics calculator (EEC), which calculates the operational cost of energy consumption for a WSN profile over a timing period via capturing both activity energy level and operational energy levels. Temperature readings and usage duration were used to find the power average and cost of electricity to calculate weather the cost is applicable or not? (Thangaraj and Anuradha, 2016). In March, Jewel *et al*. (2016) proposed the distributed and localized healing algorithm that deals with the holes of various forms and sizes despite node distribution and density. The work aimed to overcome the energy loss, by hole detection and replacement that are based on three distinct phases; 1) Hole identification is related to comparison the energy level with threshold level. 2) Hole discovery and border detection where failure of node is detected using beacon signals. These signals carry information like energy level, location of node into neighboring nodes. 3) Node healing (or Hole healing) is done by moving the nearby nodes. Then replacement by neighboring node with high energy is selected (Jewel *et al.*, 2016). In April 2016, Tahir *et al.* have evaluated the energy consumption in wireless sensor networks by proposing a scheme by a small number of high-energy nodes that gather location information and residual energy status of the sensing nodes then transmit it to the Base Station. The authors proposed an algorithm in which, a small percentage of high-energy nodes are used to convey each node's information to the Base Station. These nodes remain in sleep mode most of the time. In order to reduce energy load, further cluster head advertisement is eliminated as this function is performed by Base Station, which has sufficient energy resources (Tahir *et al.,* 2016).

Table 1: The previously related works carried out on self-maintenance clustering model.

No.	Year	Work	References
1	2014	Banerjee *et al.* tried to manage by using cellular automata (CA) for illustrating the hardware status of the node.	Banerjee *et al.*
2		Saihi *et al.* presented clustering model to detect the suspicious nodes by exchanging heartbeat messages in an active manner.	Saihi *et al.*
3		Mamta *et al.* studied power budget depletion resulting in occurrence of intermittently faulty sensor nodes at node level.	Mamta and Singh
4		Lau *et al.* analyzed the exchanged data via end-to- end the collected transmission time at the base station level.	Lau *et al*

Table 1, Cont.:

No.	Year	Work	References
1	2015	Diongue and Thiare provided self-healing mechanism based on two main algorithms for reducing energy consumption.	Diongue and Thiare
2		Dhumale *et al.* studied Fuzzy inference Logic (FIL) for fault detection and avoiding breakdown in wireless sensor networks.	Dhumale *et al.*
1	2016	Tapas Bapu *et al.* tried to recover less energy consumption nodes to enhance the network lifetime, at sensor node level.	Tapas Bapu *et al.*
2		Thangaraj and Amerada presented an energy economics calculator (EEC) computes the operational cost of energy consumption for a WSN.	Thangaraj and Amerada
3		Jewel *et al.* tried to overcome the energy loss by the hole detection and replacement are based on previously mentioned three phases.	Jewel *et al.*
4		Tahir *et al.* studied set of high-energy nodes to convey each node's information to the base station, for overcoming energy depletion.	Tahir *et al.*

MATERIALS AND METHODS

Network Model

The proposed method assumes that sensor nodes are randomly distributed in a two-dimensional field and the sensor network has the following properties:

1. WSN consists of N sensor nodes, which are distributed in clusters and the cluster head of each cluster is selected.
2. All nodes are able to transmit data to the sink point, when having sufficient energy.
3. All sensor nodes are homogeneous and all of them have the same features.
4. Good sensor node can send its sensed data to its neighbors as well as to the cluster head. The neighbor nodes are farther away from each other and hence, their readings will likely differ.
5. Synchronous mode of communication is used to send the data from all sensor nodes to cluster head within a fixed time interval.
6. There is only one cluster head has the property of data checking and data transmission to the sink node. It is deployed outside the network with larger energy source as shown in figure1.

The proposed method is built upon the clustering method in which Genetic Algorithm (GA) was used to determine the optimum network structure that reduces the energy exhaustion after each transmission round (Elhoseny *et al.*, 2015). In the optimization process, each GA chromosome represents a designation map of cluster heads. A gene in a chromosome specifies if the corresponding node serves as a cluster head. Given a cluster head, the node clusters are then formed following the nearest neighbor rule, and the fitness of a WSN structure prescribed by a chromosome is hence determined by the evaluation of all clusters. This method greatly extends the network lifetime by balancing the energy consumption among all sensor nodes in the WSN.

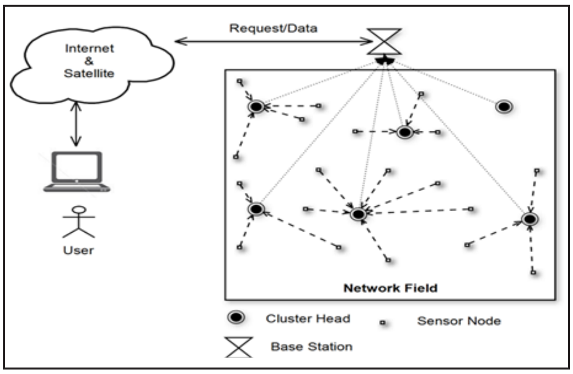

Figure 1: The network scheme of WSNs.

7. The parameters of hardware faults are represented as:

- Na is the vector that introduced to the cluster nodes and contains set of the transmitter circuit/ microcontroller faults , and represented as:

$N_a = \{a_1, a_2, \ldots, a_n\};\ N_a \in N$ (1)

- N_b is the vector that introduced to the cluster nodes and contains set of the receiver circuit faults , and represented as:
  $N_b = \{b_1, b_2, \ldots, b_n\}; N_b \in N$ (2)
- N_c is the vector that introduced to the cluster nodes and contains set of the battery faults , and represented as:
  $N_c = \{c_1, c_2, \ldots, c_n\}; N_c \in N$ (3)
- N_h is the vector that introduced to the cluster nodes and contains set of the sensor circuit faults, and represented as::
  $N_h = \{h_1, h_2, \ldots, h_n\}; N_h \in N$ (4)
- N_D is the dead vector of the network and illustrated as:
  $N_D = \{N_a \cup N_c\}, \quad N_D < N$ (5)
- N_K is the active sensor nodes vector in the network and illustrated as:
  $N_K = \{N - N_D\} \quad 0 < N_K \leq N$ (6)
- Then, the vector of traffic nodes in the network is given by:
  $N_{Traf} = \{N_K - N_b\}; N_{Traf} < N$ (7)
- The vector of sense nodes in the network is:
  $N_{Sens} = \{N_K - N_h\}; N_{Sens} \leq N$ (8)

The Energy Estimation

The consumed energy (E) in a cluster consists of the energy used to transmit messages to the cluster head and then forward the aggregated messages to the sink point. Using the modified first order radio model, the energy consumption models were represented by the following equations:
Remaining energy after transmission:

$$E_{RT} = E_o - (E_T * N_T) \qquad t_1 \leq t \leq t_2 \qquad (9)$$

Remaining energy after receive:

$$E_{RR} = E_{RT} - (E_R * N_R) \qquad t_1 \leq t \leq t_2 \qquad (10)$$

where: E_{RT} is the remaining energy after transmission of l bit data (joule), E_o is the initial energy of each node (joule), E_T is the consumed energy in transmission l bit data via a certain time (joule), N_T is number of bits transmitted through the transmission time (µs), E_{RR} is the remaining energy after receiving l bit data (joule), E_R is the consumed energy in receiving l bit data (joule), N_R is number of bits received through the receiving time (µs). The power amplification loss is calculated by comparing the energy of node after every receiving data with threshold.

The Proposed Algorithm

The proposed SHM was used to handle hardware failures of WSN nodes, by performing the operations of the proposed fault detection, fault diagnosis, and fault recovery methods. Thus, in this methodology, each node within the cluster, expressed its operational status via a periodic heartbeat message to cluster head. The proposed methodology was performed in four main phases at two levels; cluster head and node levels:

a. *Initialization phase*. In this phase, each sensor node knows its cluster head. In addition, all sensor nodes send a periodic heartbeat message to the cluster head. This message indicates that the node is alive and the transmitter efficiently works.

b. *Computation phase*. This phase was performed at two levels at node and cluster head levels. At node level, each sensor node regularly sends and receives data from all neighbor's nodes at time amounted to 300 µs, which is the time consumed between consecutive packets in sensor node (Ajofoyinbo, 2013). After that, the sensor node matches its own sensed data with its neighbor's data via the processors (*microcontroller circuits*) to diagnose its sensor circuits. Then, it transfers its private data to the cluster head. At cluster head level, the cluster head forwards the aggregated data from the nodes, and calculates the mean operation over them based on one of distributed computation theories in WSNs (Giridhar and Kumar, 2006). In SHM methodology, The integrity of microcontroller circuits, for each node present in the cluster, was evaluated using the following equation (Banerjee *et al.*, 2014):

$$e(t) = x(t) - cy(t) \qquad t_1 \leq t \leq t_2 \qquad (11)$$

where: e(t) is the information difference vector, x(t) is the value of sensed information by neighbors during a time interval (t), *c* is the difference factor, y(t) is the value of the sensed information by a sensor node during same time interval (t), and *t* is a given time interval for calculating the difference between both x(t) vector and y(t) vector through this duration. Two sensing vectors were evaluating them during various time intervals represented as t_1, t_2.

c. *Fault Detection phase.* Detection phase is the basic phase in the proposed methodology to discover the malfunction nodes within the cluster. Practically, when the cluster head receives the calculated mean and heartbeat message from each node present within the cluster during an estimated period (*t*), the head declares the sender node as *connected*. After this, it analyzes the collected data to estimate the function and detect faulty sensor nodes. The cluster head extracts the sender identifier from received data to identify which sensor nodes are able to send their data to the cluster head. The sensor nodes, which are unable to send heartbeat message during estimated time (*t*) to the cluster head, are identified as function faulty sensor nodes. After identifying faulty sensor nodes, the cluster head will detect the cluster nodes status of node either alive or dead. Then, the cluster head updates the topology by removing the malfunction node from the topology. Hence, the quality of the network can be maintained and the cluster head assigns a task to all faulty sensor nodes for further diagnosis and recovery.

d. *Fault diagnosis & recovery phase:* This phase handles the hardware components failures of sensor nodes, which are issued by receiver circuits, defects sensor unit of the node or depletion of battery's energy. Therefore, this phase is performed at main two levels as illustrated below:

1. Sensor node level.

- If the sensor node did not sense an environmental event periodically within time (t), it will diagnose this as a "*sensor error*" and sends a notification message to the cluster head. In response to this fault, the cluster head declares this node as a "*traffic node*".
- If the sensor node exceeds the threshold value and does not receive any data from its neighbors through a certain period, the receive circuit of sensor node could be declared as "*faulty*". It will announce that to the cluster head. Then, the cluster head declares this node as a "*sense node*" which only senses data and sends them to the cluster head without receiving any data from its neighbors as illustrated in Algorithm (1).
- Each node maintains its internal energy above critical level with respect to a predefined threshold value. When the power reaches a level that is less than the threshold, the node announces itself as "*sleeping node*" and sends that to the cluster head which in turn modifies the topology to remove this node and assign its functionality to its posterior.

Algorithm 1: Fault detection algorithm at sensor node level

Input: N_K Active vectors, $N_{Cluster}$ where $\forall$ N $\in N_{Cluster}$

Output: N_{Sens}, and N_{Traf}

1. Initialize: $N_K = [0]$.
2. Initialize: $N_{Sens} = [0]$, $N_{Traf} = [0]$.
3. FOR EACH N in $N_{Cluster}$ DO
   a. IF there is no received or sensed data by N THEN
      i. BREAK IF
   b. ELSE
      i. IF periodically receive data neighbor nodes THEN
         Receiver vector is active ——— non faulty "healthy"
      ii. ELSE
         $N_{Sens} = N_{Sens} + N$
      iii. END IF
   c. IF battery power threshold value THEN
      i. Battery vector is active ——— non faulty "healthy"
      ii. $N_K = N_K + N$

d. ELSE
   i. Sensor node sends *notification message* to cluster head
   ii. cluster head remove that node
e. END IF
f. IF required environmental data periodically sensed THEN
   i. Sensor vectors are active ——— non faulty "healthy"
g. ELSE
   i. Sensor node sends *notification message* to cluster head
   ii. Cluster head consider the node as *traffic node*
   iii. $N_{Traf} = N_{Traf} + N$
h. END IF

4. END IF

END FOR EACH

RETURN

The faulty node's battery may be replaced, to enter in active mode and this called feedback from the network (Watfa and Abu Assi, 2011). Hence, it starts transmitting its heartbeat message to cluster head, which in turn updates the topology by adding the node to the network. On other hand, if the faulty node did not get a feedback, it will remain in a sleep mode and thereby the cluster head isolates it from the cluster topology and selects new path, Algorithm (1). The proposed SHM methodology applied the following mathematical comparisons for diagnosing either the continuity or the recovery of battery.

If $E_{RR} > E_{thr}$ Active ⟨ Node continues its task

If $E_{RR} \leq E_{thr}$ Sleep ⟨ Node is covered

Where E_{RR} is the energy remaining after receiving l bit data (joule), and E_{thr} is the threshold value of battery (joule).

2. Cluster head level.

- As mentioned earlier, each node in the cluster sends a periodical heartbeat message to the cluster head. If the sensor node exceeded the predefined threshold value as (t) and did not send the heartbeat message to the cluster head. The cluster head will diagnose the sensor node transmitter to be faulty. In this case, the head will modify the topology to set a replacement for this node.
- Likewise, if the cluster head received a sleeping declaration from sensor node, it also modifies the network topology in the same procedure.
- If the cluster head received a sensor error notification from the sensor node, it will declare this node as a "*traffic node*". Then, it broadcasts the information about the faulty node to the base station (sink), and updates its topology. Algorithm 2 illustrates the fault detection method in the cluster head.
- The cluster head receives the processed data from all nodes within the cluster. The data are compared to diagnose the status of the cluster nodes' processors. In this case, the cluster head exploits the spatial correlations phenomenon for determining integrity of the cluster nodes' microcontroller circuits. If nearby sensors data are similar, they can be captured by spatial correlation (Krishnamachari and Iyengar, 2004). Thus, cluster head compares the nearby sensors data (referred to as the witness set) in order to determine whether the reading is faulty or not, likewise comparing their processed data in order to ensure whether status of cluster nodes' microcontroller circuit is faulty or not.

Algorithm 2: Fault detection algorithm at cluster head level

Input: Heartbeat message ($HB_{msg.}$), CH (cluster head), V_{data} (Processed data vector in CH), N where N N $_{Cluster}$, $N_{Connect}$, $N_{Connect-data}$ (Processed data of $N_{Connect}$).

Output: N_{status} , new topology

1. Initialize: $N_{Fault} = [0]$, $N_K = [0]$.
2. **For** N= 1 to $N_{cluster}$ **DO**
3. Transfer $HB_{msg.}$ **to** CH
4. **While** heartbeat message periodically sensed in CH

a) CH set $N = N_{Connect}$
b) $HB_{msg} = HB_{msg} + N_{HB\ msg}$
c) CH examines the processed data transferred from $N_{Connect}$
d) IF CH sensed the closely processed data of $N_{Connect}$ **THEN**
  i. CH set $V_{data} = V_{data} + N_{Connect}$_data
  ii. $N_K = N_K + N$
e) ELSE
  i. Broadcast the information about faulty node to the base station (sink)
  ii. CH sends *notification messages* to its previous& next neighbor node (to recover faulty node.)
  iii. CH select new path
f) END IF
g) ELSE IF heartbeat message did not sense in CH
  i. $N_{Fault} = N_{Fault} + N$
  ii. Broadcast the information about faulty node to the cluster nodes
  iii. CH removes that node from topology.
  iv. CH update topology
h) BREAK IF
5. END WHILE

END FOR
RETURN

RESULTS AND DISCUSSION

Experimental Settings

This section verifies the performance accuracy of the proposed SHM methodology, with respect to: a) detection of hardware failures in the sensor nodes, b) diagnosis of that failure, and c) recovery of malfunction nodes. The simulation scenario is applied to a cluster composed of five sensor nodes, which are structured and deployed in distributed methodology randomly, and related to cluster head node. The nodes within cluster are connected through a synchronizer to transfer numerical data set, for detection of fault, diagnosis of fault and then recovery of declared malfunction nodes.

To evaluate the proposed SHM methodology, system was implemented using five nodes, which are snooping on their neighbors' packets to compare them with their own readings, synchronously send heartbeat message to the cluster head during timing period (t =300 μs.), and announce itself as alive node. Node is responsible for diagnosing status of sensor, receiver circuits and battery of every node in the class. The cluster head is responsible for diagnosing status of transmitter and microcontroller circuit of every node within cluster and recover faulty node. For the development of proposed system, the following assumptions are taken into consideration:

1. Heartbeat message is issued from each node (as "10101010") to cluster head via estimated time which is declared in time unit class.
2. Each node senses its own data via sensor and receives neighboring nodes data; and then it performs comparison operation between both data to identify the status of its sensor circuits. If data entry of node is equal or slightly different from neighbor data, the node will transfer that data to the cluster head. If a difference occurring is large rate, the node will transfer ERROR message to cluster head class. Then, the cluster head will declare sensor circuits of that node are faulty.
3. Each sensor node in the simulation system must have an initial energy level identified as 0.5 Joule**.** The sensor node estimates its status of battery, by calculating amount of energy losses during transferring operations (transmission/receiving), with a certain threshold as 0.03 of battery level. If current energy ≤ threshold, the node will transfer SLEEP message to cluster head. In this situation, the cluster head will assign to previous or next node, for handling with tasks of the sleeping node.

Data Discussion

To evaluate the methodology of SHM methodology, the system detection ability of disconnected nodes in the cluster, diagnosis errors of sensor, receiver, and microcontroller as well as capability of SHM in recovery of the malfunction nodes were tested. The simulation results are discussed in the next sections.

- **The Cluster Connection**

The precision of the communication is based on consecutive data packets via the transmission time (t = 300 µs) and heartbeat messages which are transferring to cluster head class for detecting operational status of node whether connect or disconnect. Hence, its own transmission time at each node was changed to: 200, 250, 300, 400 and500 µs, and cluster head responses were recorded the status of transmitter circuit in each node present within node class. For testing the precision of the cluster connection, the actual number of disconnected nodes, which have introduced to it, transmission time exceeding estimated transmission time (300µs.) in the cluster during iteration was compared with the monitored number of disconnected nodes, which detected in the same iteration by simulation. The authors performed five iterations with changing transmission time at each node, for finding out the detection accuracy of transmitter faults in each of iterations; then the ratio of detection accuracy in five iterations was detected, which was amounted to 75.7 %.

- **Diagnosis of Errors**

The proposed system is enough to diagnose hardware errors. Therefore, it could be argued that integrity of hardware parameters is related to a certain time duration (t = 300 µs). The results revealed are displayed in Table 2.

Table 2: Data readings respect to detection accuracy of SHM. The first column shows some data readings of five nodes.

Data readings	Actual error	Detected error	Detection Accuracy (DA)	Consumed Time (µs)
(4,12,22,8,6)	1	1	1.0	171
(30,30,40,21,9)	3	2	0.6667	203
(0,7,34,30,22)	3	3	1.0	171
(37,34,36,45,49)	2	1	0.5	031
(34,2,28,7,40)	3	3	1.0	156
(2,27,44,8,6)	3	2	0.6667	296
(17,17,19,38,3)	3	3	1.0	281
(17,9,33,44,35)	3	2	0.6667	281
(9,20,44,35,45)	3	2	0.6667	046
(10,10,22,11,0)	2	1	0.5	265
(8,8,2,26,7)	2	1	0.5	296
(30,30,3,13,8)	3	3	1.0	214
(11,11,42,37,6)	3	2	0.6667	0
(18,18,13,14,31)	1	1	1.0	0
(40,40,1,19,23)	3	3	1.0	218
(27,27,18,1,45)	3	2	0.6667	296
(17,17,49,23,29)	3	2	0.6667	203
(21,21,40,12,26)	2	2	1.0	234
(8,8,12,48,33)	2	2	1.0	171
(2,2,42,39,30)	3	3	1.0	296
- Readings - Sensor nodes	20 5		- Total - Average - Detection Accuracy	16.1669 0.808345 80.83 %

In order to evaluate the performance of the proposed SHM system, two traditional metrics have been considered:

1. Detection Accuracy (DA): is the ratio of number of faulty sensor nodes that detected compared to number of faulty sensor nodes introduced to the network during estimated transmission time, Table (2). Figure 2 shows an evaluation of DA at different percentages of sensor circuit faults.

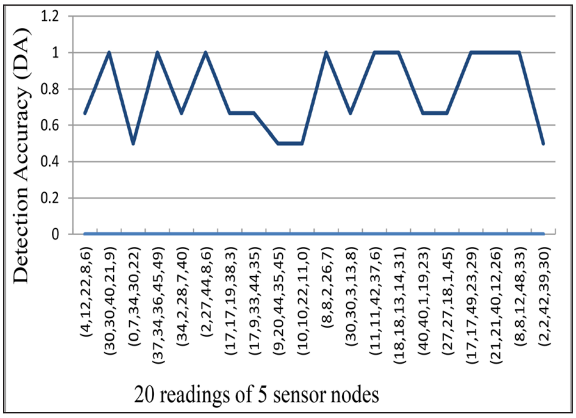

Figure 2: The illustration of DA at diffeent percentage of sensor circuits fault.

2. Consumed Time (CT): is the duration between receiving sensor data and detecting faults in the network, Table (2). Figure 3 illustrates the consumed time in the detection accuracy.

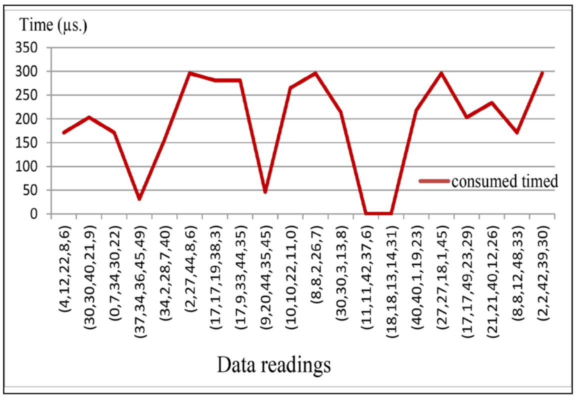

Figure 3: The illustration of consumed time in the detection accuracy.

In the proposed methodology, sensor nodes in the network are sensing, sending and receiving data packets at the same estimated period (t =300). Also, they are distributed and closely deployed in the sensing region. The sensed information of the neighboring node differs in small amount. Therefore, a receiver circuit was practically added to microcontroller and sensor circuits' compound in all of the five nodes within node class.

Practically, data entry was sensed via sensor circuits, in the proposed simulation system, and broadcasted between the cluster nodes. Then, they were checked to verify the integrity of receiver circuits by calculating the difference between sensing information and neighboring information. This is to detect the behavior of cluster nodes and determine which of them is traffic, sense or healthy nodes. As the difference vector between sensing information and neighboring information must not exceed 15°, the node would considered as faulty if this difference increased than this value and one was added into fault counter, as shown in Table 3A. The experimental results were recorded throughout five iterations for discovering receiver faults in detection phase, which are demonstrated in Table 3B.

Table 3A: Experimental results recorded via simulation for detecting the receiver circuits' faults of the cluster nodes at three iterations.

No. of cluster nodes	Sensed data	Received data	Detected fault	Detected status	Actual fault	Detection Accuracy
5	N1(29)	(8,3,26,27)	2	Sense	2	1.0
	N2(8)	(29,3,26,27)	4	Traffic	3	0.66
	N3(3)	(29,8,26,27)	3	Traffic	3	1.0
	N4(26)	(29,8,3,27)	2	Sense	2	1.0
	N5(27)	(29,8,3,26)	2	Sense	2	1.0
Total						4.66
Iteration 1	Detection accuracy in iteration 1= average of detection accuracy of the cluster nodes					DA_1= 0.932
10	N1(42)	(10,46,32,30,13,38,15,10,49)	4	Sense	4	1.0
	N2(10)	(42,46,32,30,13,38,15,10,49)	6	Traffic	6	1.0
	N3(46)	(42,10,32,30,13,38,15,10,49)	4	Sense	4	1.0
	N4(32)	(42,10,46,30,13,38,15,10,49)	1	Sense	5	0.2
	N5(30)	(42,10,46,32,13,38,15,10,49)	4	Sense	4	1.0
	N6(13)	(42,10,46,32,30,38,15,10,49)	6	Traffic	6	1.0
	N7(38)	(42,10,46,32,30,13,15,10,49)	4	Sense	4	1.0
	N8(15)	(42,10,46,32,30,13,38,10,49)	6	Traffic	6	1.0
	N9(10)	(42,10,46,32,30,13,38,15,49)	6	Traffic	6	1.0
	N10(49)	(42,10,46,32,30,13,38,15,10)	4	Sense	4	1.0
Total						9.2
Iteration 2	Detection accuracy in iteration 2 = average of detection accuracy of the cluster nodes					DA_2 = 0.92
15	N1(0)	(8,7,25,39,46,33,41,9,40,31,48,28,48,42)	11	Traffic	11	1.0
	N2(8)	(0,7,25,39,46,33,41,9,40,31,48,28,48,42)	11	Traffic	11	1.0
	N3(7)	(0,8,25,39,46,33,41,9,40,31,48,28,48,42)	11	Traffic	11	1.0
	N4(25)	(0,8,7,39,46,33,41,9,40,31,48,28,48,42)	8	Traffic	9	0.9
	N5(39)	(0,8,7,25,46,33,41,9,40,31,48,28,48,42)	4	Sense	4	1.0
	N6(46)	(0,8,7,25,39,33,41,9,40,31,48,28,48,42)	5	Sense	6	0.84
	N7(33)	(0,8,7,25,39,46,41,9,40,31,48,28,48,42)	7	Sense	4	0.25
	N8(41)	(0,8,7,25,39,46,33,9,40,31,48,28,48,42)	4	Sense	5	0.8
	N9(9)	(0,8,7,25,39,46,33,41,40,31,48,28,48,42)	11	Traffic	11	1.0
	N10(40)	0,8,7,25,39,46,33,41,9,31,48,28,48,42)	4	Sense	4	1.0
	N11(31)	(0,8,7,25,39,46,33,41,9,40,48,28,48,42)	7	Sense	6	0.84
	N12(48)	(0,8,7,25,39,46,33,41,9,40,31,28,48,42)	5	Sense	7	0.72
	N13(28)	(0,8,7,25,39,46,33,41,9,40,31,48,48,42)	8	Traffic	7	0.86
	N14(48)	(0,8,7,25,39,46,33,41,9,40,31,48,28,42)	5	Sense	7	0.72
	N15(42)	(0,8,7,25,39,46,33,41,9,40,31,48,28,48)	4	Sense	5	0.80
Total						12.73
Iteration 3	Detection accuracy in iteration 3 = average of detection accuracy of the cluster nodes.					DA_3 = 0.848

Table 3B: The detection accuracy of schema in detecting receiver faults through five iterations.

Iteration	No. of cluster nodes	Inputs faults	Total of detection accuracy values	Detection efficiency in each iteration
Iter. 1	5	2,3	4.66	0.932
Iter. 2	10	4,5,6	9.2	0.92
Iter. 3	15	4,5,6,7,9,11	12.73	0.8486
Iter. 4	20	6,8,11,15	11.53	0.5765
Iter. 5	25	11,12,15,17	8.03	0.3212
- Total			3.5983	
- Average			0.71966	
- Totally Detection Accuracy (DA)			71.96 %	

- **Capability of SHM in Recovery**

When the remaining energy (E_{RR}) becomes lower than a predefined threshold size which equals 3% of battery level (E_{thr}= 0.03), the node in the system will declare itself as sleeping node, and will seize sensing data, and then send a sleep notification to cluster head. Cluster head then modifies the topology to isolate this node. In this simulation, the ability of the system is based on detection of battery status (active or sleep) and arrival of notification messages, which are transferring from the faulty node to cluster head class for recover that node. Inthis simulation, when the head class receives covered flag from the faulty node (or called sleeping node), it removes the faulty node and updates the topology as shown in Table (4). For testing the ability of the system in detection and coverage the sleeping nodes, varying energy values for each node ranging from 0.5, to 2 Joules were given, and also the value of threshold to higher value as (threshold =0.5) was changed. An example of the nodes class response in round (3) is recorded in Table (4).

Table 4: Demonstration results revealed from the recovery of malfunction node.

Sensor node	Energy level (J)	Remaining energy (J)	Status	The cluster head action
Node1	2	1.7	active	pass
Node 2	0.5	0.2	sleep	Remove &update topology
Node 3	0.5	0.2	sleep	Remove &update topology
Node4	1	0.7	active	pass
Node 5	1.5	1.2	active	pass

Figure 4 shows evaluation of SHM coverage during different rounds for recovering malfunction nodes. SHM is fit to find a cover of malfunction nodes by locating and isolating the malfunctioning node. The system maintains nodes status using notification message. The proposed methodology achieved an improvement in coverage and improved network lifetime, because it becomes primary metric for evaluating the performance of a sensor network. Hence, the (PC) precision of coverage of faulty nodes is calculated by:

PC = (total of number of the covered nodes in each round / number of the cluster nodes) * number of rounds

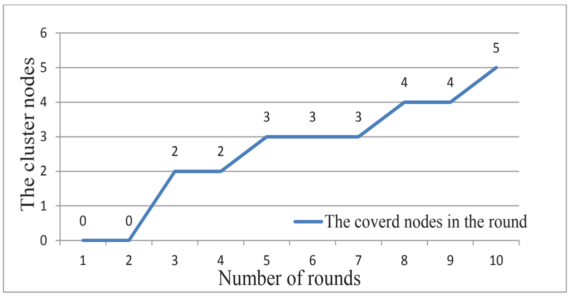

Figure 4: Evaluation of SHM in coverage for the battery faults.

SHM methodology can recover up to 52 % of battery faults compared to EFDR technique that can detect 40% of battery faults. Therefore, SHM methodology increases the network lifetime up to 62.6 %. In addition, the proposed SHM methodology handles the transmitter faults in recovery section. Where node's heartbeat message is transferred to cluster head for illustration its operation status whether connected or disconnected node. When the node sends its node's heartbeat message to the cluster head, the cluster head will send join message to the sender. In case of not arrival of heartbeat message to cluster head from any node in the cluster during a certain period (t=300µs.), it declares this node as disconnected node (as faulty). Therefore, the cluster head follows same procedure of battery recovery for recovery of transmitter faults, so the head node removes this node from topology, and asks previous node and next node to hold tasks of this node, and then selects new path. The transmission time was changed within each node in the cluster to: 150, 550, 300, 400 and 200µs, and cluster head response was recorded in the Table 5, for illustrating the precision of its recovery in recover disconnected nodes.

Table 5: Demonstration status of the cluster nodes needed to recovery in the cluster

IP address	Heartbeat message	Time	Status	Cluster head response
192.168.16.1	"10101010"	150	Connected	Join message
192.168.16.2		550	Disconnected	Remove & select new path
192.168.16.3	"10101010"	300	Connected	Join message
192.168.16.4		400	Disconnected	Remove& select new path
192.168.16.5	"10101010"	200	Connected	Join message

To evaluate the proposed methodology, we created a simulation by generating random sensor placements in the sensing field. The experiments were repeated 10 times and the average performance was reported. The results of this proposed method were compared with the results of EFDR method (Banerjee *et al*., 2014). The results of EFDR method was got by running it using the same programming language and the same parameters. In this simulation, sensor nodes are randomly deployed and randomly distributed in a square area of size 128 $m \times 128$ m. The values of different parameters used in this simulation are recorded throughout five iterations and given in Table (6). During simulation, the cluster head checks up all nodes in the cluster by heartbeat message. SHM is fast in finding a cover of malfunction nodes by locating and isolating them. The system maintains nodes status using heartbeat message. The proposed methodology achieved an improvement in faults detection and improved network lifetime, because it becomes primary metric for evaluating the performance of a sensor network.

Table 6: The experimental results revealed from applying SHM methodology

Parameter	Value
Initial energy	0.5 J
Time between consecutive packets	300 µs.
Detection Accuracy (DA)of sensor faults	76.9 %
Detection Accuracy (DA)of receiver faults	71.96 %
Detection Accuracy (DA)of transmitter faults	75.7 %
Diagnosis Efficiency (DE)	60 %
Precision of coverage (PC)	52 %
Ratio of loss energy cross transmitting phase	50 % of battery size
Ratio of loss energy cross receiving phase	40 % of battery size
Dissipated energy by transceiver circuits	10 % of battery size

To evaluate the proposed methodology, the efficiency of SHM methodology in fault detection and diagnosis was compared with that of EFDR schema (Banerjee *et al*., 2014). The comparison results clarified that the proposed methodology could tolerate up to 67.3% of different hardware faults (sensor circuit, microcontroller/transmitter, battery and receiver) compared to EFDR schema that reported 60% of hardware faults. At node level, SHM realized a detection accuracy of sensor circuit fault tolerate up to 76.9 % , 52 % of battery fault and 71.96 % of receiver faults compared to EFDR schema that reported 40 % of receiver faults. At head class level, 75.7% of transmitter fault and 60% of microcontroller circuit fault are realized. Therefore, SHM results have proved the improvement of network lifetime tolerate up to 62.6%. Figure (5) shows different percentages of hardware faults detection which detected by SHM compared to EFDR scheme.

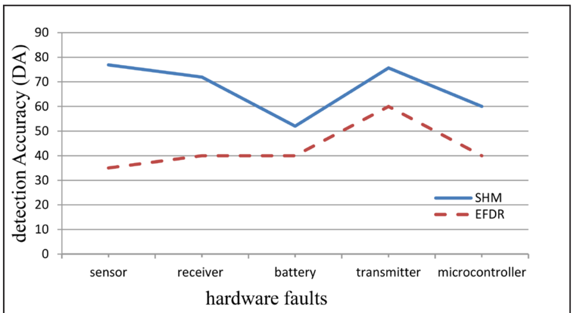

Figure 5: Comparative results of SHM and EFDR for detecting hardware faults.

CONCLUSION

WSN is a self-organized network that consists of thousands of inexpensive and low powered devices, called sensor nodes. These nodes are deployed at harsh and hostile environment that may cause sensor node failure. Therefore, self-healing methodology is one of the critical issues in WSNs. Self-healing methodology includes fault detection, diagnosis and healing. The self-healing in the majority of WSN environments is performed centrally at cluster head level.

In this paper, a distributed self-healing methodology for WSNs called `SHM' was proposed to detect, diagnose, and recover the hardware components faults. Detection and recovery is performed at both cluster head level and node level. The performance efficiency of the proposed methodology was evaluated. The results showed that SHM methodology is very efficient compared with the EFDR schema. The comparison results clarified that SHM could tolerate up to 67.3 % of different hardware faults (sensor circuit, microcontroller/transmitter, battery and receiver) compared to EFDR schema that reported 60 % of hardware faults. At node level, SHM realized a detection accuracy of sensor circuit fault tolerate up to 76.9 % , 52 % of battery fault and 71.96 % of receiver faults compared to EFDR schema that reported 40 % of receiver faults. At head class level, 75.7 % of transmitter fault and 60 % of microcontroller circuit fault are realized. Therefore, SHM results have proved the improvement of network lifetime tolerate up to 62.6 %.

Future proposed plans will investigate the software faults and self-configuration options for sensor nodes. In addition, the experiments will test different scenarios for the stability of the proposed SHM.

REFERENCES

Ajofoyinbo, A.M. 2013. Energy efficient packet-duration-value based MAC protocol for wireless sensor networks. Wireless Sensor Network. 5(10): 194-202.

Banerjee, I., Chanak, P., Rahaman, H., and Samanta, T. 2014. Effective fault detection and routing scheme for wireless sensor networks. Computers and Electrical Engineering. 40(2): 291–306.

Dhumale, R., Lokhande, S., Thombare, N., and Ghatule, M. 2015. Fault detection and diagnosis of high-speed switching devices in power inverter. International Journal of Research in Engineering and Technology (IJRET). 4(2): 253-257.

Diongue, D., and Thiare, O. 2015. An energy efficient self-healing mechanism for long life wireless sensor networks. pp. 599-605. In: Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering. Volume - 313 of the book series Lecture Notes in Electrical Engineering (LNEE). Springer, Cham.

Elhoseny, M., Yuan, X., Yu, Z., Mao, C., El-Minir, H., and Riad, A. 2015. Balancing energy consumption in heterogeneous wireless sensor networks using genetic algorithm. IEEE Communication Letters. 19(12): 2194 - 2197.

Jewel, D., Brundha, P., Joy Winnie Wise, D. C., and Aravind Swaminathan, G. 2016. Improved hole detection healing and replacing algorithm for optimal coverage in wireless sensor networks. International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET). 2(2): 2395-1990.

Krishnamachari, B., and Iyengar, S. 2004. Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. IEEE Transactions on Computers. 53(3): 241-250.

Lau, B. C.P., Ma, E. W.M., and Chow, T.W.S. 2014. Probabilistic fault detector for wireless sensor network. An International Journal of Expert Systems with Applications. 41(8): 3703–3711.

Liu, H., Nayak, A., and Stojmenović, I. 2009. Fault tolerant algorithms/protocols in wireless sensor networks. pp. 261-291. *In:* Guide to Wireless Sensor Networks. Computer Communications Networks. DOI 10.1007/978-1-84882-218-4_10

Mamta, and Singh, S. 2014. Fault detection and correction in using shortest path in wireless sensor networks. International Journal of Research (IJR). 1(7): 514-523.

Rodrigues, A., Camilo, T., and SáSilva, J. 2013. Diagnostic tools for wireless sensor networks. Journal of Network and Systems Management. 21(3): 408–452.

Saihi, M., Zouinkhi, A., Boumedyen, B., and Abdlkrim, M. N. 2014. A real time centralized fault detection application in wireless sensor networks. pp. 95-101. *In:* International Conference on Automation, Control, Engineering, and Computer Science. Proceedings - IPCO. Sousse, Tunisia.

Tahir, M., Khan, F., Jan, S. R., and Azim, N., Khan, I. A., Ullah, F. 2016. EEC: Evaluation of energy consumption in wireless sensor networks. International Journal of Engineering Trends and Applications (IJETA). 3(2): 2393 – 9516.

Thangaraj, M., and Anuradh, S. 2016. Energy conscious deterministic self-healing new generation wireless sensor network: Smart WSN using the Aatral framework. pp. 1-18. Wireless Networks. Springer. DOI: 10.1007/s11276-016-1214-2

Tapas Bapu, B.R. and Siddanna Gowd, L.C. 2016. A novel FPGA implementation for a self-healing reconfigurable system in wireless sensor network. East Journal of Scientific Research. 24(5): 1878-1886.

Watfa, M.K., and Abu Assi, R. 2011. A distributed algorithm for isolating malfunctioning nodes in wireless sensor networks. pp. 523-530. *In:* International Conference on Advances in Education and Management. Proceedings - ISAEBD 2011. Communications in Computer and Information Science - Springer. Berlin, Heidelberg.

تحسين أداء الشبكة في شبكة الاستشعار الاسلكية باستخدام نموذج تجميع ذاتي الصيانة

ولاء محمد السيد[1] و سحر فوزى سبح[1] و علاء محمد رياض[2]

(1) قسم نظم المعلومات، كلية الحاسبات والمعلومات، جامعة بنها، مصر
(2) قسم نظم المعلومات، كلية الحاسبات والمعلومات، جامعة المنصورة، مصر

استلام 21 ديسمبر 2016م - قبول 28 مارس 2017م

الملخص

يعد تصحيح الخطأ في عقدة الاستشعار مسألة شاقة في شبكات الاستشعار اللاسلكية (WSNs)؛ حيث تنشر عقد الاستشعار عادة في بيئات غير مؤهلة. وتعد الذاكرة المحدودة، وقوة المعالجة، ونطاق الاتصال لعقد الاستشعار من العوامل التي تجعل مخططات تصويب الخطأ التقليدية غير ممكنة في شبكات الاستشعار اللاسلكية.

في هذا البحث، تم تقديم منهجية مقسمة للشفاء الذاتي تجري فيها عمليات الكشف والتشخيص والعلاج على مستوى كلٍّ من العقدة ورئيس المجموعة. على مستوى العقدة، تم تشخيص أخطاء كل من البطارية، وأجهزة الاستشعار والاستقبال. في حين تم تشخيص كل من أجهزة الإرسال وكافة العقد التي تعاني من فشل وظيفي على مستوى رئيس المجموعة، وتغطيتها. وتبين نتائج المحاكاة أن المنهجية المقترحة هي مخططات دقيقة لتحديد مكان العقد التي تعاني من خلل وظيفي، وسريعة في إيجاد غطاء لمثل هذه العقد.

الكلمات المفتاحية: شبكات الاستشعار اللاسلكية، الشفاء الذاتي، عطل العقدة، النموذج العنقودي.