Hierarchical Fault Detection and Recovery Framework For Self-Healing WSN

R. Anitha (anitha02phd@gmail.com)
SA Engineering College
Tapas Bapu B R
SA Engineering College
V. Nagaraju
Rajalakshmi Institute of Technology
Pradeep. S
SA Engineering College

Research Article

Keywords: WNS, RTP, HDFR, Self-healing, PSO, Fault Detection

Posted Date: July 20th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-699566/v1

Hierarchical Fault Detection and Recovery Framework For Self-Healing WSN

[1]Ms. R.Anitha, [2]Dr. Tapas Bapu B R, [3]Dr. V.Nagaraju and [4]Mr. Pradeep.S

[1] Assistant Professor, MCA Dept, S.A.Engineering College

Research Scholar, Anna University, Chennai

[1]anitha02phd@gmail.com

[2]Professor, ECE, S.A.Engineering College

[2]tapasbabuphd@gmail.com

[3]Professor and Head, ECE, Rajalakshmi Institute of Technology

[3]vankadarinagaraju@ritchennai.edu.in

[4]Associate Professor, ECE, S.A.Engineering College

[4]pradeep.shanmugam4@gmail.com

Abstract: Wireless Sensor Network (WSN) contains several sensor nodules that are linked to each other wirelessly. Errors in WSN may perhaps be because of several causes which bring about hardware damage, power thwarts, incorrect sensor impression, faulty communication, sensor deficiencies, etc. This damages the network process. In this paper, we propose to develop a Hierarchical Fault Detection and Recovery Framework (HDFR) for Self-Healing WSN. This framework consists of three modules: Fault detection, fault confirmation and fault recovery. In fault detection module, Particle Swarm Optimization (PSO) algorithm is applied for estimating the discrete Round Trip Paths (RTPs). Along the established RTPs, round trip delay (RTD) time values are estimated. Then based on the RTD, the suspected nodes are identified. In fault confirmation module, the nodes are confirmed to be either in FAULTY or ACTIVE state. In fault recovery module, the primary controller (PC) will establish an alternate route via the secondary controllers (SC) by excluding the faulty nodes. Then, it will resend the stored packets to the sink via the newly established route. By experimental results, it is shown that the HDFR framework achieves better detection accuracy and packet delivery ratio.

Keywords: **WNS; RTP; HDFR; Self-healing; PSO; Fault Detection**

1. Introduction

1.1 Wireless Sensor Network (WSN)

In a sensor system, there occur abundant sensor nodules and little disreputable locations. The sensor nodules make, function and transmit the data packages to the disreputable locations via the transitional nodules [1]. In the WSN, the system is functioned by the sensor nodules, which connect the whole system and transfer via multi hops by carrying out tasks like getting data packages, handling it and supplying it at the relevant terminuses [2]. In WSN, the system topology varies comprehensively. WSN may have a star topology, ring topology, mesh topology, etc. The sensor nodules may transmit data packages from the basis to terminus either by steering or by engulfing. In WSN, the entire sensor nodules have the capacity to pose themselves and familiarize to the network necessities. The nodules strongly amend to the changing network topology and necessitate, and also decrease the connection expenditures when matched with the conservative system expenditures [3].

In the WSN, the sensor nodules practice the system strength. The sensor nodules are categorized into several modules like detecting nodules, advancing nodules and drop nodules. The detecting nodules are in charge for detecting the indicated feature, generate data packages and transmit it to the terminus. The advancing nodules supply the data package to the indicated locality by advancing the data packages after choosing the finest adjacent nodules. The drop nodule performs as a connection amid the base station and the system nodules, and is linked to the base station over the Universal Serial Bus (USB) link or through wireless link [4].

1.2 Fault Detection and Recovery in WSN

Errors in the WSN may be because of several causes such as tough situation situations which bring about hardware damage, power thwarts, incorrect sensor impression, faulty communication, sensor deficiencies, etc. Depending on the system level being infected the maximum, the errors on every stage is categorized as Network stage, Nodule stage, or Sensor stage. There are numerous sorts of network errors. Certain errors are link damage, error in steering route, jamming in the route, etc. These errors make difficulty at the time of data transmission amid nodules and therefore are measured as connection letdowns. When the sensor nodules do not work correctly because of the problem in any of its constituents such as radio, CPU, battery, memory, etc, then it is measured as nodule error. It effects in irregular rearranging, improper detected data, data broadcasts of little eminence, etc and is measured as data disaster. Error in sensor nodules disturbs only the identified data and therefore taken into consideration as data letdowns [5].

When error acceptance methods are active in the network, it will utilise huge quantity of vigor for identifying the error and then to overwhelm the error, else it will need additional hardware along with software source to identify the error. The error controlling method engaged in the WSN is superior when related to the ones utilised in the conservative networks. To manage numerous sorts of errors, various appliances have been intended which considers few of the problems found in WSN. A noble error controlling method will be considered as many as error as likely [6].

2. Related Works

Ivana Tomic et al [8] have offered Antilizer, a frivolous, fully-distributed answer to allow WSNs to sense and recuperate from communal network stage bout situations. In Antilizer every sensor nodule forms a self-referenced faith exemplary of its community by means of network earwigging. The nodule utilises the faith exemplary to separately familiarize its communication resolutions. In the circumstance of a network bout, a nodule is able to create adjacent cooperation steering resolutions to evade pretentious areas of the network. Mobile proxies additionally assure the harm produced by bouts. These mediators allow a humble announcement system which spreads cooperative resolutions from the nodules to the base station. A sifting appliance at the base station additionally authenticates the genuineness of the evidence divided by mobile mediators.

Abolfazl Akbari et al [9] have planned a cluster-based retrieval procedure, which is energy-efficient and receptive to network topology difference which is the consequence of sensor nodule letdowns. By this method, the group connectivity can be recovered in less significant period when likened with the period occupied by the error-tolerant grouping method and it is also fast in recuperating from errors. The comeback period essential by this method is less important with no needless intrusion in its functionalities. The network era is prolonged with the capable utilization of dynamism in the network.

Ravindra Navanath Duche et al [10] have suggested a Sensor Nodule Letdown Discovery Based on Round Trip Delay and Tracks in WSNs. The discovery of the nodules with errors is achieved based on the round trip delay occurred while travelling the round trip path. This technique is ensured depending on the execution and analysis done on the hardware and software. The discovery on the base of the round trip delay is ample and the scalability is definite on dissimilar WSN.

Khalid Mahmood et al [11] have hosted an intellectual on-demand connectivity refurbishment method for wireless sensor networks to discourse the connectivity refurbishment difficulty, where nodules use their broadcast variety to make sure the connectivity and the standby of unsuccessful nodules with their jobless nodules. The planned method

aids us to retain path of network topology and can reply to nodule letdowns efficiently. Thus their scheme can well manage the problem of nodule letdown by presenting fewer overhead on sensor nodule, highly effectual dynamic use, healthier analysis, and connectivity deprived of touching the sensor nodules.

Salim Ghanemi et al [12] have offered a dual self-healing method to strengthen MANET survivability. Initially, an error-tolerant IDS is intended by duplication of distinct mediators inside MASID to make sure the unceasing management of the network. Yet, as not the entire interruptions are expectable, there might have certain severe results on the network previously being sensed and entirely detached. Even for that, if the insinuations of interruptions could be diminished by the interruption discovery scheme MASID, still the requirement for the retrieval of transformed or removed data is a vigorous stage to guarantee the precise working of the network. For that, a recovery-oriented method for a self-healing MANET is also accessible. It is centered on the capacity of MASID-R to evaluate the harm produced by the noticed interruptions and intended at allowing the overseen network to reconcile itself of those errors and harms.

Stefano Galzarano et al [13] have engrossed on data errors, by initially learning the influence of ruined data, distressing identified data by diverse sorts of data-fault prototypes, on the exactness of a human doings appreciation system. Then, they define how the SPINE-* structure can augment the WBSN scheme by accumulating contributory autonomic components giving the essential selfhealing processes. They have seen that the usage of autonomic components creates the scheme more effectual and unfailing appreciations to its enhanced patience to data errors, as proven by investigational outcomes.

Shenfang Yuan et al [14] have planned a technique to improve a wireless sensor nodule with self-healing capacity centered on reconfigurable hardware. Two self-healing WSN nodule apprehension hypotheses centered on reconfigurable hardware are accessible, comprising a redundancy-based self-healing prototype and a complete FPAA/FPGA centered self-healing prototype. The nodules intended with the self-healing capability id able to vigorously alter their nodule structures to heal the nodules' hardware letdowns. To validate these two models, an anxiety sensor nodule is accepted as an image to display the ideas. Two anxiety WSN sensor nodules with self-healing capability are established correspondingly based on the suggested self-healing models.

Sani Abba et al [15] have offered an independent self-aware and adaptive error-tolerant routing method (ASAART) for WSN. They discourse the confines of self-healing routing (SHR) and self-selective routing (SSR) methods for steering sensor data. They also inspect the addition of autonomic self-aware and adaptive error discovery and flexibility methods for path creation and path renovation to offer flexibility to faults and letdowns.

3. Hierarchical Fault Detection and Recovery Framework (HFDR) for Self-Healing WSN

In this paper, a HFDR framework is designed for self-healing WSN. The system model and brief overview of the framework are presented in section 3.1 and 3.2, respectively.

3.1 System Model

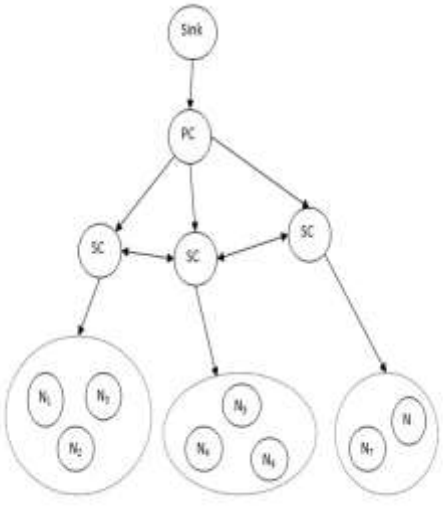

Figure 1 System Model

In this framework, primary (PC) and secondary (SC) controllers are deployed in the network along with the sensor nodes. Figure 1 shows the system model of the proposed framework. Here, Sink represents the sink node. PC and SC represent the primary and secondary controllers. N_1-N_8 represents the sensor nodes.

The SCs starts moving when requested by the PC. It is assumed that the recovery agent (RA) resides at the PC. PC will store the copy of every sent packet during every active detection interval of time. Each is SC is connected to a group of sensor nodes as well as with each other. The PC contains the accurate location information of all sensor nodes and SCs at the time of deployment. When a SC is moving, it will update its network topology information.

3.2 Brief overview of HFDR framework

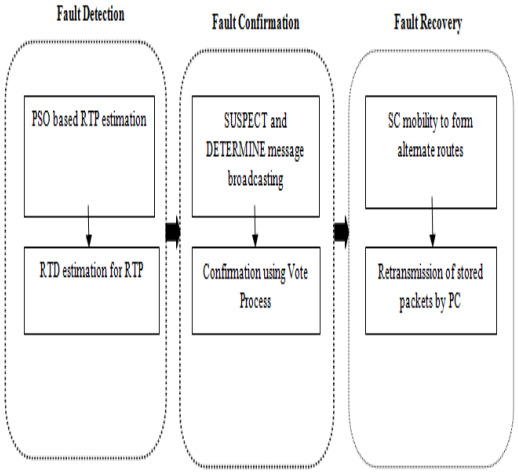

Figure 2 Block Diagram of HFDR framework

In this framework, each sensor node will be in either FAULTY or ACTIVE state. In the fault detection module, PSO algorithm is applied for estimating the discrete RTPs. Along the established RTPs, round trip delay (RTD) time values are estimated. Then based on the RTD, the suspected nodes are identified. In the fault confirmation module, Then these suspected nodes broadcast messages to its neigbhors. When the neighbors receive these messages, they reply based on the information recorded in its neighbor table. On the basis of the received message, the nodes are confirmed to be either in FAULTY or ACTIVE state. In the fault recovery module, the RA at the primary controller (PC) will establish an alternate route via the secondary controllers (SC) by excluding the faulty nodes. Then, it will resend the stored packets to the sink via the newly established route. Figure 2 shows the block diagram of the HFDR framework.

3.3 Fault Detection Module

3.3.1 Particle Swarm Optimization (PSO)

In the PSO method, the network produces a precise group of software mediators called elements which appears for the faultless resolution for a specific problem. PSO is a superior sort of horde intellect system in which the elements located in the exact area verifies the suitability purpose on the source of its position. These elements are skillful of stirring about the precise area centered on the earlier evidence acquired from the nearby nodules [7].

Particles self-updates by tracking the following two "extreme":

P_{best}(i) is got from the element itself and called the distinct extreme

G_{best}(i) is got from the present populace and called the universal optimal.

After finding the two optimal values, new velocity λ_i and new location σ_i of the particle are updated as per the following equation:

$$\lambda_i \text{ (t+1)} = \Omega \times \lambda_{id}(t) + L_1 \times rand() \times [P_{best}(t) - \sigma_{id}(t)] + L_2 \times rand() \times [G_{best}(t) - \sigma_{id}(t)] \quad (7)$$

where

$$\sigma_{id}(t+1) = \sigma_{id}(t) + \lambda_{id}(t+1) \quad (1)$$

$1 \le i \le D, D$ = number of initializes particle swarm.

$1 \le d \le V, V$ = dimension of searching space

$1 \le t \le D_{\max}, D_{\max}$ = desired iteration of particle swarm

Ω = inertia weight

L_1, L_2 = learning factor

rand( ) = random number in the range {0, 1}

In order to update the individual historical optimal position and optimal location of the particles, an objective fitness function is used and a new individual and global optimal value is obtained as follows

$$P_{best}(i)\ (t+1) = \begin{cases} P_{best(i)}(t), if\ \ fitness(\sigma_{id}(t+1) \ge fitness(P_{best(i)}(t)) \\ \sigma_{id}(t+1), if\ \ fitness(\sigma_{id}(t+1) < fitness(P_{best(i)}(t)) \end{cases} \quad (2)$$

3.3.2 Fault Detection using PSO

The faulty nodes in the network are determined based on the observed RTD values, along the the RTP. Initially, RTP in the network is determined using PSO. When the RTP is detected, the number of nodes involved is also determined. The distance between any two nodes in the network is assumed to be equal. After the paths are determined, then the RTD involved with each path is determined. Based on the estimated value of RTD, the nodes are detected to be faulty or not.

Table 1 presents the notations and their definitions used in the PSO based fault detection algorithm.

Notations	Definition
RTP	*Round Trip Path*
S	*Search space*
p_i	*Particles i=1,2….k*
X_i	*Position of particle p_i in S*
V_i	*Velocity of particle p_i in S*
LB	*Local best position*
GB	*Global best position*
RTD	*Round Trip Delay*
RTD_{est}	*Estimated Round Trip Delay*
n	*number of nodes*
T	*Time required to travel between two nodes*
RTD_{LThres}	*Lower bound threshold for RTD*
RTD_{UThres}	*Upper bound threshold for RTD*
$ACTIVE$	*Message indicating node to be active*
SUSPECT	*Message indicating node as suspected*
DETERMINE	*Message indicating node as highly suspected*
E_{res}	*Residual energy*
E_i	*Initial battery power of the node*
E_{tx}	*Transmitting energy*
E_{rx}	*Receiving energy*
TL_0	*Traffic Unit*
x	*Number of old load samples*
T	*Time interval*
$\alpha_i(t)$	*Counting function over T*
DU_s	*Delivery utility initialized to 0*
DU_i	*Delivery utility initialized to 1 if it is a sink node*
β	*Predefined constant with value [0,1]*
$\lambda_1, \lambda_2 \text{ and } \lambda_3$	*Weight values*
F_i	*Fitness Function*

Table 1 Notations used in the PSO based Fault detection algorithm

The software agents called as swarm particles are created by the network,

The swarm particles estimate residual energy at every node according to the equation given below:

$$E_{res} = [E_i - (E_{tx} + E_{rx})] \quad (3)$$

The traffic unit and delivery utility at each node is estimated according to Eq. (4) and (5)

$$TL_0 = \frac{1}{x}\frac{1}{T}\sum_{i=1}^{x}\sum_{t=1}^{T}\alpha_i(t) \quad \textbf{(4)}$$

$$DU_s = \beta \times DU_s^{'} + (1-\beta) \times DU_i \quad (5)$$

Based on the monitored parameters, fitness function (F_i) of each particle is estimated based on below

$$F_i = (\lambda_1 * DU_s) * (\lambda_2 * E_{res}) / (\lambda_3 * TL) \quad (6)$$

After the determination of all the available RTPs, the RTD of each path is estimated according to the equation given below.

$$RTD_{est} = n\ T \quad (7)$$

The PSO based fault detection algorithm is given below

Algorithm: PSO based Fault Detection

---

1. For each P_j
2. Do
3. p_i are initialized in S at position x_i
4. X_i represents the adjacency matrix towards R_j
5. p_i computes F_i using Eq. (6)
6. If F_{xi1} > LB (x_{i1}) then
7. LB (x_{i1}) = F_{xi1}
8. Else
9. LB(x_{i1}) is not modified
10. End if
11. If F_{xi1} > GB (x_{i1}) then
12. GB(x_{i1}) = F_{xi1}
13. Else
14. GB(x_{i1}) is not modified
15. End if
16. V_i is updated using Eq.(1)
17. X_i is updated using Eq.(2)
18. Move p_i to a X_{i+1}
19. Repeat Until (LB=GB)
20. For each rtp
21. Compute RTD using (7)
22. If $RTD_{est} < RTD_{LThres}$, then
23. Generate SUSPECT message
24. Else if $RTD_{est} > RTD_{UThres}$, then
25. Generate DETERMINE message
26. Else If $RTD_{LThres} < RTD_{est} < RTD_{UThres}$, then

27. Node is ACTIVE
28. End if
29. End For
30. Stop

In this algorithm, based on the fitness function of the consecutive node, the path followed during data transmission is discovered since the path followed for transmitting a specific data packet +will have similar F_i. Thus, the swarm particles traverse the route and determine the round trip path. On traversing the round trip path, the particles update the path and all the nodes involved in the path. In this way, all the RTP present in the network are determined. After the determination of all the available RTP, the RTD of each path is estimated. The estimated RTD is compared against a threshold value set $\{RTD_{LThres}, RTD_{UThres}\}$. If estimated RTD is less than RTD L_{Thresh}, then a SUSPECT message is generated. If it is more than $RTD_{UThresh}$, then a DETERMINE message is generated. On the other hand, if the estimated RTD is between RTD $_{Lthresh}$ and $RTD_{Uthresh}$, then the node is considered as an ACTIVE and valid sensor node.

The ACTIVE nodes are authenticated and are considered secure for network operation. But, the nodes for which the SUSPECT or DETERMINE message is generated i.e., nodes which are detected to be faulty, for those nodes, fault confirmation has to be performed.

3.4 Fault Confirmation Module

Once a node is detected to be faulty, then it has to be confirmed [2]. Every node in the network maintains a neighbor table which includes all the information related to its neighbors. For making decision about the suspected node, every node uses the information recorded in the neighbor table.. This process is described in algorithm 2.

Algorithm: Fault Confirmation

Notations	**Definition**
N_i	Initiator which generates SUSPECT or DETERMINE message
$\{Ne_i\}$	Set of Neighbors of N_i
CONFIRMED	Message confirming node to be faulty
$VOTE_{FALSE}$	Vote message indicating node status suspected to be false
$VOTE_{TRUE}$	Vote message indicating node status suspected to be true
$VOTE_{Threshold}$	Threshold for VOTE messages

---

1. For each Ni
2. Initiate SUSPECT timer
3. Broadcast message to $\{Ne_i\}$
4. For each node $N_{ij} \in \{Ne_i\}$
5. If received message is SUSPECT, then
6. If ID(suspected) in Neighbor table, then
7. If status is malicious , then
8. N_{ij} send $VOTE_{TRUE}$ to N_i
9. Else
10. N_{ij} send $VOTE_{FALSE}$ to N_i

11. End if
12. Else
13. N_{ij} send $VOTE_{FALSE}$ to N_i
14. End if
15. If SUSPECT timer expired, then
16. If $VOTE_{TRUE} > VOTE_{Threshold}$, then
17. N_i broadcasts DETERMINE to 3-hop neighbors
18. End if
19. End if
20. End if
21. If received message is DETERMINE, then
22. If ID(N_i) is in Neighbor table, then
23. N_{ij} send CONFIRMED to Ni
24. End if
25. End if
26. End For
27. End For

When the SUPECT message generated by the initiator, it, is broadcasted to the neighbor nodes after setting the SUSPECT timer on. When a node receives a SUSPECT message, the node will check whether suspected node's ID is in its neighbor table. If the suspected node ID is present in its neighbor table, then, its status is checked. If the status indicates that it is malicious, then it will send the $VOTE_{TRUE}$ message back to the initiator. If the status indicates that it is not malicious, then the node will send back a $VOTE_{FALSE}$ message. When the SUSPECT timer expires, the initiator node checks the number of vote messages received. If the number of $VOTE_{TRUE}$ is greater than the $VOTE_{Threshold}$, then the initiator node will three-hop broadcast a DETERMINE message. When the nodes receive the DETERMINE message, then it will check if the initiator node ID is present in its neighbor table. If present, then the node will send back the CONFIRMED message indicating that the node is confirmed to be faulty.

As soon as the initiator received the CONFIRM message, it forwards it to the PC, which in turn will invoke the fault recovery module.

3.5 Fault Recovery Module

After the node is confirmed to be faulty, the network recovers itself from the faulty nodes and protects the other valid nodes.

The steps involved in the fault recovery module are presented below:

1. If PC receives CONFIRM message from the initiators, then RA at PC, broadcasts Fault Recovery Request message (FR_REQ) to SCs that includes: PC ID, faulty node id and detection time.

2. Each SC upon receiving FR_REQ, will check its routing table for the faulty node id. If it exists, then it responds with the fault recovery response message FR_RES that includes: PC ID, SC ID, and its route information towards sink.

3. On receiving the FR_REP message from all SCs, PC will try to establish a new route excluding the faulty node to replace the damaged route.

4. If no such route can be formed, then PC will send MOBILITY information packet to the corresponding SCs such that a new route towards the sink can be formed.

5. On receiving the MOBILITY information packet, the corresponding SC move towards the position as specified by the PC.

6. PC will then resend the stored packets to the sink via the newly established route.

4. Experimental Results

4.1 Experimental Settings

The proposed HFDR framework is simulated in NS2 and its performance is compared with autonomous self-aware and adaptive fault-tolerant routing technique (ASAART) [15] and RTD [10] protocols. The performance is measures in terms of packet delivery ratio (PDR), average packet drop, average residual energy and fault detection accuracy. The experimental settings are tabulated in Table 1.

Number of nodes	20 to 100
Topology size	500m X 500m
MAC Protocol	IEEE 802.11b
Traffic type	Constant Bit Rate
Traffic rate	100Kb
Propagation model	Two Ray Ground
Antenna model	Omni Antenna
Initial Energy	12 Joules
Transmission Power	0.660 watts
Receiving Power	0.395 watts

Table 1: Simulation parameters

4.2 Results & Discussion

The simulation results are presented in the next section.

A. Varying the Nodes

In this section, the results of varying the number of nodes from 20 to 100 are presented.

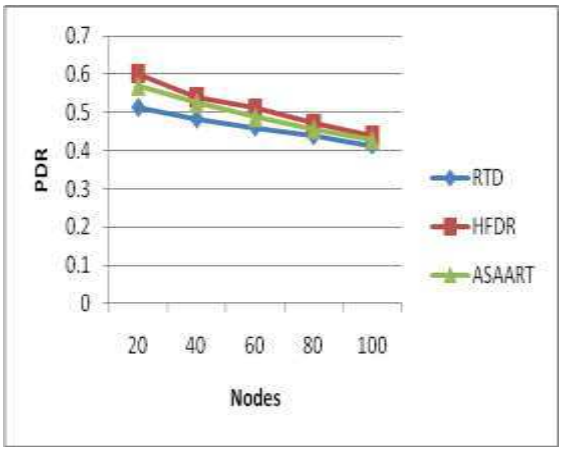

Figure 3 PDR for varying the nodes

The graph showing the results of PDR for varying the nodes is shown in Figure 3. The figure depicts that the PDR of HDFR ranges from 0.60 to 0.44 and PDR of RTD ranges from 0.51 to 0.41 and the PDR of ASAART ranges from 0.56 to 0.49. Ultimately, the PDR of HDFR is 8% high when compared to RTD and 6% of high when compared with ASAART.

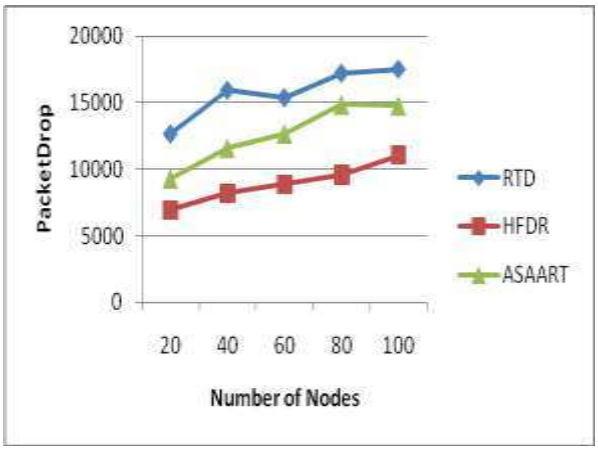

Figure 4 Packet Drop for varying the nodes

The graph showing the results of packet drop for varying the nodes is shown in Figure 4. The figure depicts that the packet drop of HDFR ranges from 1949 to 11067 and packet drop of RTD ranges from 6108 to 17539 and the packet drop of ASAART ranges from 4258 to 14796. Ultimately, the packet drop of HDFR is 51% less when compared to RTD and 21% of less when compared with ASAART.

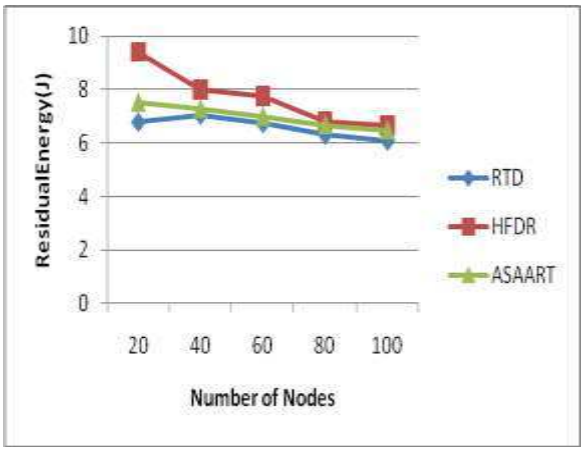

Figure 5 Average Residual Energy for varying the nodes

The graph showing the results of residual energy for varying the nodes is shown in Figure 5. The figure depicts that the residual energy of HDFR ranges from 9.4 to 6.6 joules and residual energy of RTD ranges from 6.7 to 6.0 joules and the residual energy of ASAART ranges from 7.5 to 6.4. Ultimately, the residual energy of HDFR is 14% high when compared to RTD and 6% of high when compared with ASAART.

B. Varying the Faults

In this section, the results of varying the number of faults from 1 to 5 are presented.

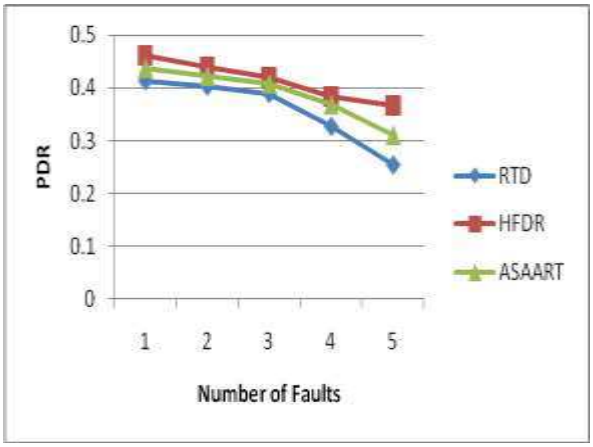

Figure 6 PDR for varying the faults

The graph showing the results of PDR for varying the failures is shown in Figure 6. The figure depicts that the PDR of HDFR ranges from 0.46 to 0.36 and PDR of RTD ranges from 0.41 to 0.25 and the PDR of ASAART ranges from 0.43 to 0.31. Ultimately, the PDR of HDFR is 14% high when compared to RTD and 9% of high when compared with ASAART.

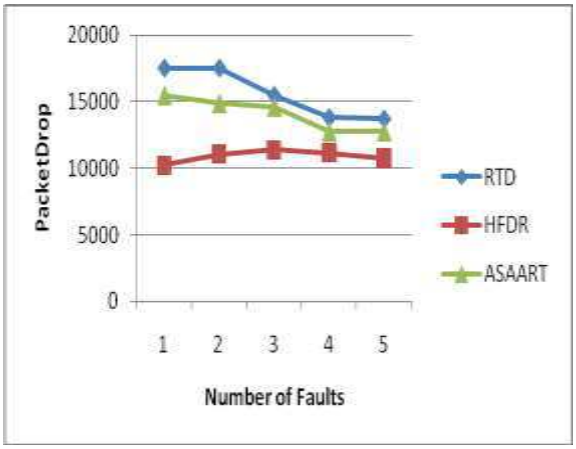

Figure 7 Packet Drop for varying the faults

The graph showing the results of packet drop for varying the failures is shown in Figure 7. The figure depicts that the packet drop of HDFR ranges from 10228 to 10771 and packet drop of RTD ranges from 17539 to 13699 and the packet drop of ASAART ranges from 15478 to 12785. Ultimately, the packet drop of HDFR is 29% less when compared to RTD and 9% of less when compared with ASAART.

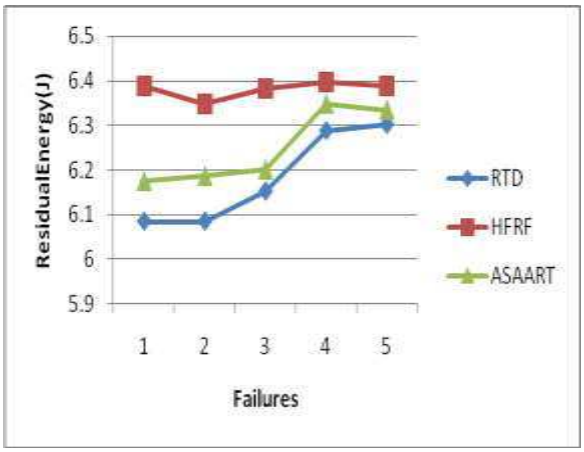

Figure 8 Residual Energy for varying the faults

The graph showing the results of residual energy for varying the failures is shown in Figure 8. The figure depicts that the residual energy of HDFR ranges from 6.4 to 6.3 joules and residual energy of RTD ranges from 6.0 to 6.3 joules and the residual energy of ASAART ranges from 6.1 to 6.3 joules. Ultimately, the residual energy of HDFR is 3% high when compared to RTD and 1% of high when compared with ASAART.

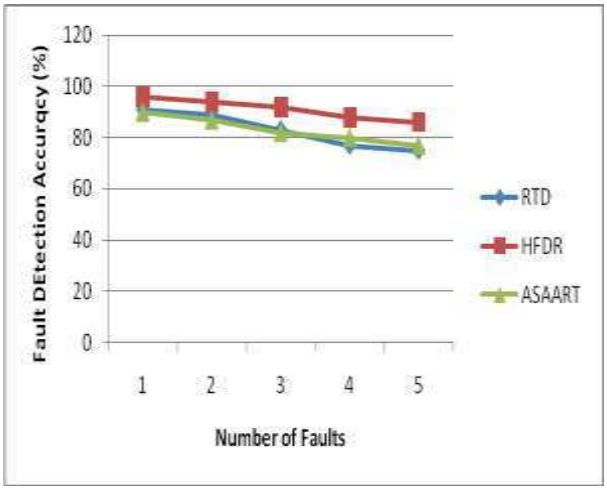

Figure 9 Fault detection accuracy for varying the faults

The graph showing the results of fault detection accuracy (%) is shown in Figure 9. It shows that HFDR has 9% higher accuracy than RTD and 8% higher accuracy than ASAART.

5. Conclusion

In this paper, HDFR framework for Self-Healing WSN has been designed. This framework consists of three modules: Fault detection, fault confirmation and fault recovery. Initially, the faulty nodes are detected based on the RTD during RTP. The nodes detected to be faulty broadcast message to its neighbors. Thus, the nodes detected to be faulty are checked with respect to the response obtained from the neighboring nodes. Based on the response received from the neighboring nodes, the detected faulty node is either confirmed to be faulty or not. Then the faulty nodes are isolated from the other valid network nodes until it recovers from the fault. By experimental results, it is shown that the HDFR framework achieves better detection accuracy and packet delivery ratio.

Declaration:

This manuscript was originally prepared by me and my co-authors. This is not submitted to any-where for Publication.

Ethical Approval - NA

Consent to Publish - NA

Consent to Participate – NA

Authors Contribution – All the authors are equally contrbuted

Funding – Not Applicable

Competing Interests – The authors don’t have any Competing Interests

Data Availability - The authors didn’t use any third party data on their manuscript.

References

1. Maulin Patel, R. Chandrasekaran and S.Venkatesan,(2005), “Energy Efficient Sensor, Relay and Base Station Placements for Coverage, Connectivity and Routing”, 24th IEEE International Performance, Computing, and Communications Conference, USA.

2. Feifei Li, Xi Wang, and Tingrong Xu,(2011), “Energy-aware Data Gathering and Routing Protocol Based on Double Cluster-heads”, Communications in Information Science and Management Engineering, CISME Vol.1 No.4,PP.24-29.

3. Amlan Kumar Nayak and Bibhas Mishra,(2012), “Fault Detection in Wireless Sensor Network Using Distributed Approach”, Department of Computer Science & Engineering, National Institute of Technology Rourkela, Rourkela, Orissa, 769 008, India.

4. A. T. I. Fayeez, V. R. Gannapathy, A. S. Baharom, Ida S. Md Isa, M. K. Nor and N. L. Azyze,(2015), “Real Time load distribution via Particle Swarm Optimization (PSO) for Wireless Sensor Network (WSN)”, ARPN Journal of Engineering and Applied Sciences, ©2006-2015 Asian Research Publishing Network (ARPN). ISSN 1819-6608. VOL. 10, NO. 3.

5. A. De Paola, G. Lo Re, F. Milazzo, M. Ortolani,(2013), “QoS-aware Fault Detection in Wireless Sensor Networks”, In International Journal of Distributed Sensor Networks, vol. 2013,
Article ID 165732, 12 pages.

6. A.Manikandan and S.Rathinagowri,(2014), “An Efficient Detection and Recovery of Fault node in Wireless Sensor Networks”, International Journal of Innovative Research in Science, Engineering and Technology, Volume 3, Special Issue 1.

7. Valeria Loscrí, Enrico Natalizio, Francesca Guerriero and Gianluca Aloi,(2012), “Particle Swarm Optimization Schemes Based on Consensus for Wireless Sensor Networks”, MSWiM’12, October 21–25, 2012, Paphos.

8. Ivana Tomić, Po-Yu Chen, Michael J. Breza and Julie A. McCann,(2018), "Antilizer: Run Time Self-Healing Security for Wireless Sensor Networks",arXiv:1809.09426.

9. Abolfazl Akbari , Arash Dana, Ahmad Khademzadeh and Neda Beikmahdavi,(2011), “Fault Detection and Recovery in Wireless Sensor Network Using Clustering”, International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 1.

10. Ravindra Navanath Duche and Nisha P. Sarwade,(2014), “Sensor Node Failure Detection Based on Round Trip Delay and Paths in WSNs”, IEEE SENSORS JOURNAL, VOL. 14, NO. 2.

11. Khalid Mahmood ,Muhammad Amir Khan ,Mahmood ul Hassan ,Ansar Munir Shah,Shahzad Ali and Muhammad Kashif Saeed,(2018), "Intelligent On-Demand Connectivity Restoration for Wireless Sensor Networks",Wireless Communications and Mobile Computing,Volume 2018, Article ID 9702650, 10 pages.

12. LeilaMechtri,FatihaTolba,SalimGhanemi,DamienMagoni,(2017), "A Twofold Self-Healing Approach for MANET Survivability Reinforcement", International Journal of Intelligent Engineering Informatics,Vol-5,No-4.

13. Stefano Galzarano, Giancarlo Fortino and Antonio Liotta,(2012), "Embedded self-healing layer for detecting and recovering sensor faults in body sensor networks", IEEE International Conference on Systems, Man, and Cybernetics,Korea.

14. Shenfang Yuan, Lei Qiu, Shang Gao, Yao Tong and Weiwei Yang,(2012), "Providing Self-Healing Ability for Wireless Sensor Node by Using Reconfigurable Hardware",Sensors,doi:10.3390/s121114570.

15. Sani Abba and Jeong-A Lee,(2015), "An Autonomous Self-Aware and Adaptive Fault Tolerant Routing Technique for Wireless Sensor Networks",Sensors,doi:10.3390/s150820316.