Recognition and Revival of Failure Nodes based on RDE in WSN

M. Senthil[1], K. Sugashini[2], M. Abirami[3], N. Vaigai[4]
Department of Computer Science and Engineering
Christ College of Engineering & Technology
Puducherry, India.
senthilmresearch@gmail.com[1], success.sugashini@gmail.com[2], abiamarnath@gmail.com[3], vaigainataraj20@gmail.com[4]

Abstract - Wireless sensor networks are used to monitor physical and environmental conditions in various areas in order to reduce the manpower. The deployment of sensor nodes becomes increased in WSN since it provides the high quality of service (QoS).But such network can be affected by failures such as hardware, battery failure or some environmental factors. This affects the QoS and leads to failure or malfunctioning of sensor nodes and so increases the usage of sensor node. To avoid such problems, identifying the repaired nodes and recovering the malfunctioning node is essential. This goal is achieved by framing WSN with distributed hash table (DHT) which provides an efficient recovery method known as Checkpoint recovery and the topology is constructed using this table in a decentralised approach. The randomly directed exploration (RDE) of this approach uses the Adhoc on Demand distance Vector (AODV) routing protocol to find the repaired node directly. The proposed method finds the repaired node directly and recovers malfunctioning node, consumption of time and energy is reduced thereby increases the performance level.

Keywords: WSN, QoS, AODV, RDE, DHT, checkpoint.

I.INTRODUCTION

A Wireless sensor network is a collection of sensor node organized into a cooperative network. Sensor network consist of multiple detection stations called sensor nodes, each of which is small, less weight and portable. Due to the advancement in electronic fabrication technology, the usage of sensor nodes has been increased. The major requirement used in wireless sensor network is scalability, security, production cost etc.Some of the unique characteristics of WSN are application specific, scale, density and deployment.

The application of WSN is military, environmental monitoring, industrial monitoring, health monitoring, and home monitoring[1-2] and the advantages are it avoids a lot of wiring, it can accommodate new devices at any time, it is flexible to go through physical partitions, and it can be accessed through a centralized monitor.

The sensor nodes in WSN can become inefficient at some stage due to battery failure, inhospitable environment and unattended deployment. In order to make the WSN to work properly, the repaired nodes need to be removed. For that, we frame the Distributed Hash Table (DHT), uses a decentralized approach to catch faulty nodes.

The protocol‟s performance on memory consumption and a critical security metric are theoretically deducted through a model often called probability, supported by the simulations. In the analysis, the simulation results show that the DHT-based protocol can detect node faulty with high security level in WSN.

II. PROBLEM MEASURES IN NODE FAILURE DETECTION

The topology used in this paper is star and mesh in order to connect „n‟ number of sensor nodes in the network. In [14] the method is combined with time delay based DOA estimators and tested with simulations. In RTD algorithm, it have only static sensor to sense the neighbour nodes energy level and also used for backup the data in particular time which reduces the data transmission speed. In [12] they have been using ring and mesh combination topology, there only limited number of sensor nodes are allowed also the system is based upon only time. The proposed method depends on both time and distance. Similarly the malfunction [4] nodes are recovered using checkpoint recovery by DHT algorithm.

The checkpoint recovery is mainly used to recover malfunction node. So that the performance is also gets increased. We find both faulty and malfunction nodes by means of the parameter time and distance using routing protocol called AODV [3]. The main advantage of this protocol is having routes established on demand and that destination sequence numbers are applied to find the latest route to the destination.

III. MECHANISMS FOR FINDING FAILED NODES

The failure of a sensor node can be find by using two approaches[8], called self- detection(passive), and active detection. These two approaches are the basis for finding the faults in the wireless sensor network.

In self-detection, sensor nodes need to be monitored within a stipulated time to check their energy level. In active detection cell manager asks its cell members to regularly send their update message which consists of their id, location information.

If any node fails to update itself, it will be examined by the cell manager with an instant message. If the cell manager not received the acknowledgement for instant message, it will be declared repaired one to its neighbours. If any node sends improper updates or exceeds the threshold time, it may be a malfunctioning one.

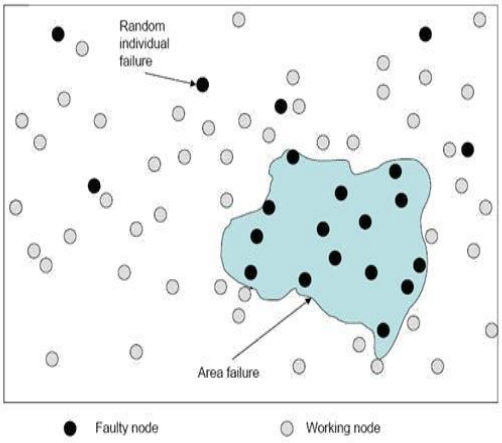

Fig 1.Network model and Fault model

IV.FACTORS INVOLVED IN ANALYZING THE NODE

A. Evaluation of Traverse Time for the RTP

The time taken to traverse the path depends on number of sensor node present in the round trip path and the distance between them. Delay time can only be decreased by reducing the sensor nodes in RTP Selecting minimum numbers of sensor nodes in the RTP will reduce the RTD time. But the RTD of RTP needs to have continuity in the topology. This causes the accuracy to lag in its level. To overcome this issue, we modify the topology structure as well as go for AODV protocol [3].

B.DHT Implementation

The address space is a parameter of DHT. DHTs use 128-bit or 160-bit key space. Some real-time applications uses DHTs hash functions other than SHA-1.Some DHTs may also publish objects of different types. For example, key „k" could be the node ID and associated data could describe how to contact this node. In simple, ID is just a random number that is directly used as key k.

In some DHTs, publishing of nodes IDs is also used to reduce DHT operations. Redundancy is used to improve reliability. The (k, data) key pair can be stored in more than one node corresponding to the key.

C. DHT Interfaces

There are two angles from which the functionality of Distributed Hash Tables can be viewed: they can be represented as routing systems or as storage systems. The first one focuses on, the delivery of packets to nodes in a DHT based on a destination ID and in second thing, a Distributed Hash Table projected as a storage system similar to a hash table.

Routing Interface

Routing in a DHT is done in the logical address space which is divided among the participating nodes. Any identifier from the address space can act as a destination address for a message. The feature given by the DHT is to forward a message for an ID to the node which is responsible for this identifier.

Storage Interface

As a storage system, a DHT implements an interface for storing and retrieving data in a distributed way. Each node is provided with two primitives by the application interface. The put primitive takes a (key, value) pair and stores it on the node responsible for the identifier key and the get primitive accepts an identifier and returns the value associated with the specified identifier.

D. DHT Requirements

Distributed Hash Tables provide a layer for routing and managing data in distributed systems. By broadcasting routing information and data across multiple nodes, the scalability of centralized systems are avoided while data retrieval is more efficient than in unstructured WSN. Also, Distributed Hash Tables supports a wide spectrum of applications and uses. This is shown in their properties, such as scalability, routing latency, fault tolerance, and adaptability.

V. NODE FAILURE DETECTION USING RDE

Node failure is detected using the distributed detection protocol, Randomly Directed Exploration [14] .Each node has the neighbour list and the detection is initiated by sending claiming messages to other its neighbour.

Algorithm is executed in two phases, the First phase is used to decide the threshold value of delay time of the nodes and fault is detected in the second phase. The highest value of delay time found during the execution of first phase is selected as the threshold time for all paths in WSNs.

In the second phase, instantaneous delay time of discrete paths is compared with the threshold time. The path to which the delay time is found to be greater than threshold time is then analyzed further. This particular discrete path is examined in some stages to locate the exact position of fault.

Let S*X* be the source node with sequence of sensor nodes such as S*X*–S*X*+1–S*X*+2. Faulty sensor node in the WSNs can be present at position S*X* or S*X*+1 or S*X*+2 in path [12]. Hence path formed by these sensor nodes have to be checked to identify the fault. Detected faulty sensor node, which can be either failed or malfunctioning, is verified by comparing the delay times of respective Paths with threshold time.

A particular sensor node in WSNs can be declared as faulty in order to test and verify the suggested method. Faulty sensor node can be either failed or malfunctioning so two cases have to be considered and implemented in order to justify the proposed method.

TABLE1. FAULT DETECTION ALGORITHM

Step 0: At each node V_i, create a neighbour table NT_i and $M=[m_{jk}]$, and set m_{jk} to 1 and F_i to 1 (faulty)

Step 1: (threshold test at each V_i) For k=1 to q V_i tests $V_j \in R(V_i)$

If $|X_i^K - X_j^K| \leq \partial$,then set m_{jk} to 0

If $\sum_{K=1}^{q} m_{jk} \leq (q - \theta_2)$, then set C_{ij} to 0

If $|C_i| \geq \theta_1$, then set F_i to 0 and send F_i to neighbours

Step2: **For** the remaining undetermined nodes

Do the following in parallel for l cycles

For each V_k with $C_{ik} = 0$,$C_{ki} = 0$, $F_i = 0$ and $F_k = 1$ Set F_k to 0 and send F_k to neighbors

Failed (dead) sensor node detection is done by declaring the particular node as dead in tcl script. Same as, malfunctioning behaviour is detected by adding certain delay in the Paths of particular sensor node.

TABLE2. PATH ANALYSIS FOR DETECTING FAULT IN WSNS

1. Select any sensor node S_X from WSN with N sensor nodes,
   The values of X=1, 2, 3.....N ($S_1 \leq S_X \leq S_N$).
2. RTP_X formed has sensor sequence as S_X-S_{X-1}-S_{X-2}.
3. Call subroutine "RTD Time".

RTD time Subroutine

I. If $S_{X+1}=S_N$ then replace S_{X+2} by S_1
   Else if $S_{X+1}>S_N$ then replace S_{X+1} by S_1 and S_{X+2} by S_2 respectively.

II. Measure the round trip delay time of corresponding RTP. Initially it is RTP_X.

III. Return to main program.

4. If $\tau_{RTD}\_X=\tau_{THR}$ then Increment S_X by 3($S_X=S_{X+3}$)
   If $S_X+3>S_N$ then reset S_{X+3} to S_N and go to step 2
   Else go to step 2
   Else Call subroutine "RTD Time". Measure RTD time of RTP_(X+1) having sequence as S_{X+1}.S_X+2-S_{X+3}.
5. If τ_{RTD} (X+1)=τ_{THR} then go to step 7
   Else if τ_{RTD} X=∞ then S_X node is failed (dead).
   Otherwise S_X node is malfunctioning.
6. Go to step 4
7. Call Subroutine "RTD Time". Measure RTD time of RTP_(X+2) having sequence as S_{X+2}-S_{X+3}-S_{X+4}.
8. If τ_{RTD} (X+2)=τ_{THR} then go to step 10
   Else if τ_{RTD}_(X+1)=∞ then S_{X+1} node is failed(dead)
   Otherwise S_{X+1} node is malfunctioning
9. Go to step 4
10. If τ_{RTD}_(X+2)=∞ then S_{X+2} node is failed(dead)
    Otherwise S_{X+2} node is malfunctioning
11. If $S_{X+2}>S_N$ then go to step 4
12. Stop.

VI. RECOVERING THE COMMUNICATION PATH

The modules used in this paper starts with the setting up the sensor network. The sensor network is setup with the „n" no. of nodes.RDE algorithm uses the DHT to store the node details such as its id, name, and capacity.

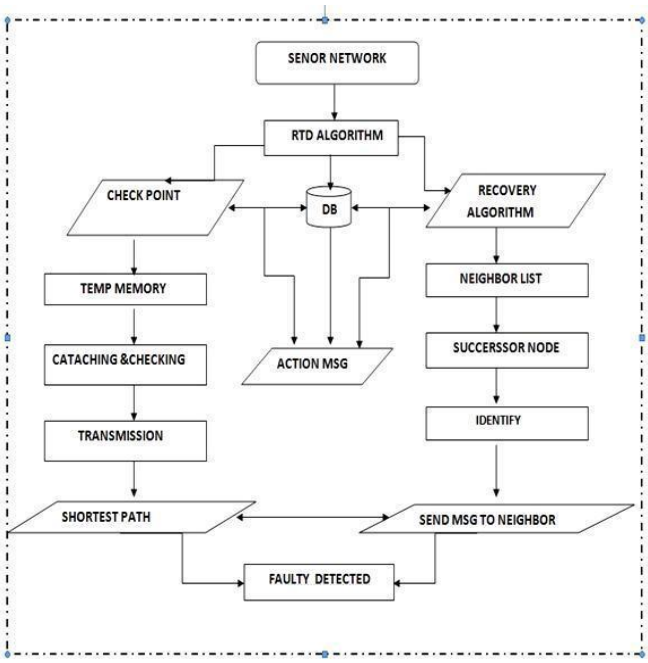

Fig2. Fault Detection Strategy

The action messages are sent between the nodes to share their updates. If any node fails to send the updates, it will be suspected by the source node, and then analyzed in detail.

For this process, the RDE algorithm checks for the path to find the delay time of a suspected node. If it exceeds the predefined threshold time which is set by the source node, it will be the malfunctioning node.

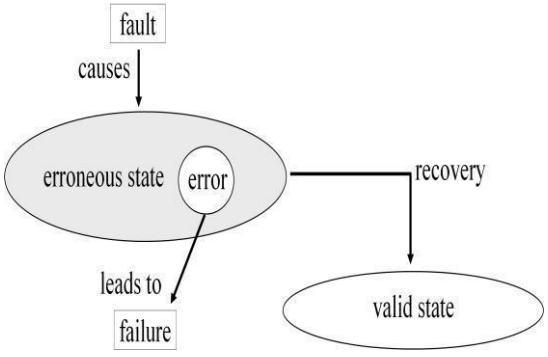

Fig3. Finding faulty node and recovering malfunctioning node

When the node could not function any more, it will be considered Failure (dead) and will be detached. After that the path will be reconstructed to continue the process.

The communication path can be recovered from the failure state using the decentralized approach. To achieve this; the approach provides the recovery algorithm called checkpoint recovery. This recovery algorithm works on DHT, which contains „n" no. of nodes dynamically often known as mesh type. This algorithm recovers the malfunctioning node by communicating with the recently interacted nodes of a faulty one.

VII. PERFORMANCE ANALYSIS

TABLE3. SIMULATION PARAMETERS

PARAMETERS	***SELECTED PARAMETERS VALUES***
Number of Sensor Nodes	6,10,20,30,40,50 and 100
Simulation Area	20×20meters
Simulation Time	2.2 sec for 100 nodes
Routing Protocol	AODV
Transmission Range	1 meter
Traffic Type	CBR
Initial Sensor Node Energy	250mJ
Packet Size	20 Bytes

In Analysis process, the following parameters are considered.

A. Packet Delivery Ratio

The Packet Delivery Ratio is for finding the ratio of packets which have been sent to the Destination. Then, the ratio is compared with the packets sent from the source node.

B.Residual Energy

The parameter Residual energy, using this Enhanced Network we have to generate a high energy when compared to the existing work such as RDE and RTP methods for finding faulty nodes [15]. The amount of energy that could be consumed can be reduced by this optimization technique as depicted in following graph. Using DHT method, after 10 seconds of simulation, average energy level decreased to 85mj from 250mj (Initial energy).

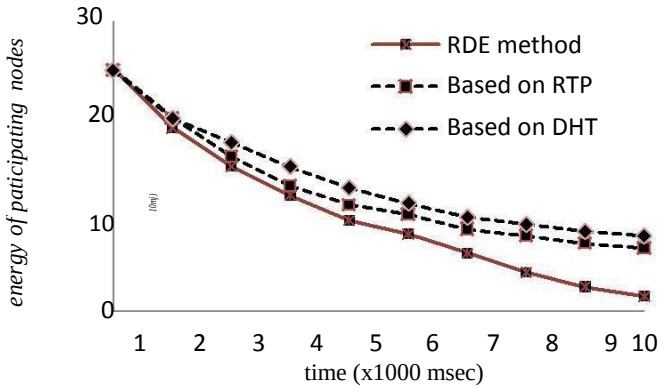

Fig4. Average Residual Energy of Individual nodes

C.Result Analysis

Method described to detect the fault is implemented and tested in NS2 simulation software. Since the implementation of sensor nodes in hardware is difficult process, the proposed method is being tested in NS2 software. The methods to find the repaired nodes are specified in the sections III and IV. This proves the effectiveness of the proposed concept in the software successfully.

VIII. CONCLUSION

The proposed method is successfully implemented, tested and verified using the NS2 software. The efficiency of this method has been increased due to the usage of the DHT which allows the recovery techniques to find the malfunctioning nodes. This leads to enhancement in the performance level and also raises the detection probability.

REFERENCES

[1] K. Sha, J. Gehlot, and R. Greve, "Multipath routing techniques in wireless sensor networks: A survey," Wireless Personal Commun., vol. 70, no. 2, pp. 807–829, 2013.

[2] M. Asim, H. Mokhtar, and M. Merabti, "A fault management architecture for wireless sensor network," in Proc. IWCMC, Aug. 2008, pp. 1–7.

[3] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," Ad Hoc Netw., vol. 6,no. 4, pp. 621–655, 2008.

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

[5] I. Chen, A. P. Speer, and M. Eltoweissy, "Adaptive fault tolerant QoS control algorithms for maximizing system lifetime of query-based wireless sensor networks," IEEE Trans. Dependable Secure

[6] Comput.,vol. 8, no. 2, pp. 1–35, Mar./Apr. 2011.

[7] A. A. Boudhir, B. Mohamed, and B. A. Mohamed, "New technique of wireless sensor networks localization based on energy consumption,"Int. J. Comput. Appl., vol. 9, no. 12, pp.25–28, Nov. 2010.

[8] M. Lee and Y. Choi, "Fault detection of wireless sensor networks Comput. Commun, vol. 31, pp. 3469–3475, Jun. 2008.

[9] Akbari, A. Dana, A. Khademzadeh, and N. Beikmahdavi, "Fault detection and recovery in wireless sensor network using clustering," IJWMN vol. 3, no. 1, pp. 130–138, Feb. 2011.

[10] C.-C. Song, C.-F. Feng, C.-H. Wang and D.-C. Liaw, "Simulation and experimental analysis of a ZigBee sensor network with fault detection and reconfiguration mechanism," in Proc. 8th *ASCC*, May 2011, pp. 659–664.

[11] A. Mojoodi, M. Mehrani, F. Forootan, and R.Farshidi, "Redundancy effect on fault tolerance in wireless sensor networks, "Global J. Comput. Sci. Technol., vol. 11, no. 6, pp. 35-40, Apr. 2011.

[12] S. S. Ahuja, R. Srinivasan, and M. Krunz, "Single-link failure detection in all-optical networks using monitoring cycles and paths," IEEE/ACM Trans. Netw., vol. 17, no. 4, pp. 1080–1093, Aug. 2009.

[13] R. N. Duche and N. P. Sarwade, "Sensor node failure or malfunctioning detection in wireless sensor network," ACEEE Int. J. Commun., vol. 3, no. 1, pp. 57–61, Mar. 2012.

[14] T. W. Pirinen, J. Yli-Hietanen, P. Pertil, and A. Visa, "Detection and compensation of sensor malfunction in time delay based direction of arrival estimation," IEEE Circuits Syst., vol. 4, no. 1, pp. 872–875, May 2004.

[15] Neenu Gorge and T.K Parani "Detection of Node Clones in Wireless Sensor Network Using Detection Protocols, IJETT" vol.8, no.6, Feb 2014.

[16] Ravindra Navanath Duche and Nisha P. Sarwade, "Sensor Node Failure Detection Based on Round Trip Delay and Paths in WSNs", IEEE Sensors Journal, Vol. 14, No.2, February 2014.