

## Discrete Path Selection and Entropy Based Sensor Node Failure Detection in Wireless Sensor Networks

*Spoorthi K.<sup>1</sup>, Snehanshu Saha<sup>2</sup>, Archana Mathur<sup>2</sup>*

<sup>1</sup>*PES Institute of Technology, BSC, Bangalore, India*

<sup>2</sup>*Department of Computer Science and Engineering, PES Institute of Technology, BSC, Bangalore, India*

*Emails: hebbbar.spoorthi@gmail.com snehanshusaha@pes.edu archanamathur@pes.edu*

**Abstract:** *Exertion of wireless sensor networks has been increasing in recent years, and it imprints in almost all the technologies such as machine industry, medical, military and civil applications. Due to rapid growth in electronic fabrication technology, low cost, efficient, multifunctional and accurate sensors can be produced and thus engineers tend to incorporate many sensors in the area of deployment. As the number of sensors in the field increases, the probability of failure committed by these sensors also increases. Hence, efficient algorithms to detect and recover the failure of sensors are paramount. The current work concentrates mainly on mechanisms to detect sensor node failures on the basis of the delay incurred in propagation and also the energy associated with sensors in the field of deployment. The simulation shows that the algorithm plays in the best possible way to detect the failure in sensors. Finally, the Boolean sensing model is considered to calculate the network coverage of the wireless sensor network for various numbers of nodes in the network.*

**Keywords:** *Wireless Sensor Network (WSN), sensor node failure detection, entropy, network coverage, energy model.*

### 1. Introduction

Wireless Sensor Networks have gained significant popularity in a current technological era. It has a broad range of applications in environmental changes, detection mechanisms, battlefield surveillance, industrial process monitoring, machine's and human health monitoring, etc. Wireless Sensor Networks (WSN) consist of small, portable, non-expensive devices called as sensors. This handle

collects the sensory information from the place of deployment and sends them to a machine, where processing of this information takes place internally. Sensors mainly consist of a microcontroller, external memory, transceiver, power source, analog to digital conversion and other components to capture variation in temperature, pressure, light, etc. A WSN normally will have little or no infrastructure. WSN consists of collaborative sensor nodes used to monitor the area and extract information about the environment. We can broadly classify the WSN as structured WSN and unstructured WSN. An unstructured WSN mostly contains of an enormous collection of sensor nodes, and it normally uses Ad hoc method of deployment. These nodes deployed in the network monitor the field. Huge numbers of sensor nodes make system maintenance difficult, and failure detection becomes a tedious task as well. But in a structured WSN, deployment of nodes is done in a predefined manner and a few nodes are used. Thus, the prime advantage of this methodology is to lower maintenance cost compared to an unstructured WSN. But sensor nodes must be placed very intelligently such that the performance of the WSN may not be affected. Present work considers a structured WSN.

Sensors have limited areas of coverage, prompting deployment of a vast number of those to capture accurate information for further processing. Another reason that might explain the tendency to incorporate many sensors is because of the low cost. As the number of nodes in the field increases, the probability of failure of nodes due to hardware inefficiency will also increase. Thus, detecting sensor node failure mechanism needs robust strategies. The mechanism has to be quite accurate and fast which will not disturb the actual intention of the deployment of the network in the field. Hence, there is a lot of scope for algorithmic design to detect sensor node failure.

Sensor node failure in the network may lead to unresponsive behavior of the node that might be caused by disconnection in radio antenna, crash in the system, drastic energy depletion, and anomalous reboot in the system. The current work simulates three algorithms that are proposed to enhance the performance by considering the drawbacks of current detection algorithms. Simulations compare the performance of the algorithms. The methodology considers the circular topology of nodes, since the distance between the neighboring nodes are almost equal, which in turn helps us in neglecting the overhead because of the distance between the nodes while transmitting the information.

The paper proposes three algorithms where the first one considers the Round Trip Propagation Delay (RTPD) between the concrete paths formed in the network; compares with the threshold value calculated *a priori*. The second algorithm endeavors to improve the performance of the first algorithm by reducing the number of paths in the network so that the time required to find the failed sensor node is reduced. Finally, the third algorithm consists of only one path and finds the failed node by considering propagation delay and the energy associated with the nodes.

In a nutshell, the algorithms deal mainly with the energy consumption of nodes and propagation delay incurred in sending packets to their neighboring nodes.

## 2. Related work

There are numerous algorithms for detection of sensor node failure. [1] Proposes one such algorithm which tackles fault detection in a distributed way, by comparing the neighboring nodes and disseminating the decisions made by each node. Nodes, identified as “malfunctioning”, will act in communication for routing and algorithm isolates them logically from the network. The involvement of malfunctioning nodes in communication process makes this algorithm inefficient; since these may not transmit the actual or complete data.

[2] proposes Cluster Head Failure (CHF) recovery algorithm which divides the network into clusters and for each cluster the head is selected which is responsible for finding the faulty node by storing the state-space information of other nodes. Two types of ranges are defined, namely transmission range and sensing range. The cluster head is selected based on its energy and transmission range. The method groups the Nodes under cluster head’s transmission range in one cluster. Since the cluster heads stores information regarding other in cluster head, if the cluster head fails, the information needs to be transferred to a secondary cluster head. Hence, there are possibilities of data loss in this mechanism.

[3] discusses a situation where large numbers of portable sensor nodes can be deployed to enhance the quality of service and the outcome of such exercise in triggering the probability of sensor node failures WSNs. QoS of wireless sensor network is affected by failure of sensor nodes, in particular node failure. Such failure is detected by measuring Round Trip Delay (RTD) time of round trip paths. The delay is compared with the threshold values. By reducing RTD time, the authors claim that the accuracy of fault detection technique is increased.

[4] Advocates Path (AP) redundancy technique to find the faulty sensor node where redundancy path is included in the network. If any of the nodes fail, the sensory information is sent using the redundant path to the base station. But this increases the energy consumption in the network since nodes store the redundant path’s information. Also, the excessive number of paths decreases the efficiency of the fault detection process.

[5] Propounds Link failure detection mechanism, where monitoring of cycles and paths is introduced. Monitoring location is the field that is considered for sensing. Failure of monitoring cycles and paths that is unique to one monitoring site results in link failure detection. Each monitoring location considers Three-edge connectivity. But the different wavelength is assigned to each link, which introduces redundancy in this algorithm.

[6] Elucidates Fault (EF) detection scheme for sensor network in a distributive manner, which uses local comparisons with sensed data and sensory data of neighboring nodes. It also considers a confidence factor calculated by all its neighboring nodes and makes decisions based on these parameters. However, this scheme is pretty complex where the exchange of sensory information has to take place twice, once for comparison and a second time for confidence factor computations, thus hindering the efficiency of the algorithm.

A sizable chunk of the task of failure detection is due to the fact that routing is a complex task in WSN owing a number of various constraints such as cross layer interference, multiple channels and radio frequencies [7-9]. Many algorithms for failure detection in wireless sensor networks have been proposed and implemented with their share of pros and cons [10-12]. Fault management architecture and smart detection techniques tend to address failure detection in WSN [13,14]. In our work, a method to detect failed sensor network by selecting paths efficiently, minimizing the time required for the detection process has been proposed. We have considered the circular topology of a network where the distance between the nodes is almost equal. Since propagation delay depends on distance and speed of light, distance can be ignored in the detection mechanism. Thus, detection algorithm becomes simpler and more efficient.

### 3. Paths and propagation delay

Fixed number of nodes forms the path in the network, which will propagate packets among themselves. The algorithm forms a specific number of paths and compares the sum of propagation delay with the predetermined threshold value. Round trip propagation delay is the sum of the propagation delays between all the nodes in the path. The fault node detection time depends on the number of paths in the network and the round trip propagation delay between the paths.

#### 3.1. Round trip propagation delay estimation

We calculate RTPD for each path in the network. Each path may consist of a varying number of nodes. RTPD depends mainly on the number of nodes in each path. The distance between the nodes will also have a greater impact on this calculation. As the number of nodes in the path increases, the time required to find the failed sensor node increases. Similarly, distance is directly proportional to propagation delay. Thus, the number of nodes in the path should be constant for all the paths and minimal. Also, the distance between the nodes in the path should be constant. Hence, the circular topology of the network places nodes at an equal distance (Fig. 1).

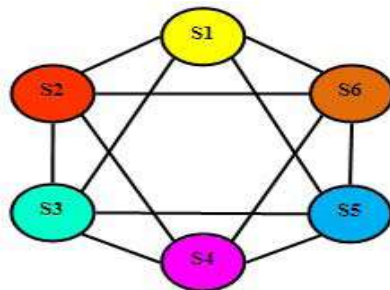


Fig. 1. Circular topology with six sensor nodes

We have considered a minimum possible number of nodes in each path so that the time required for the malfunctioning node detection mechanism can be reduced. Thus, three nodes in each path are considered. Thus, round trip propagation delay for one particular path will be the sum of the propagation delay between node 1 to node 2, node 2 to node 3 and node 3 back to node 1,

$$(1) \quad \tau_{\text{RTPD}} = \tau_1 + \tau_2 + \tau_3,$$

where  $\tau_1$ ,  $\tau_2$ ,  $\tau_3$  are propagation delays for sensor node pairs (1, 2), (2, 3) and (3, 1). The propagation delay is dependent on the distance between the nodes and is directly proportional to the distance between the nodes and will be almost equal between the nodes as the distance is constant. A round trip propagation delay for a path is given by

$$(2) \quad \tau_{\text{RTPD}} = 3\tau.$$

The time must be the minimum round trip propagation delay of a path in the network. The propagation delay of a particular sensor node that is decided by application of network and also the distance between the nodes determines the round trip propagation delay. The efficiency of the method can be improved only by reducing the number of paths in the network.

### 3.2. Evaluation of paths in the network

The algorithm mainly depends on the round trip propagation delay of each path in the network and is compared with a threshold RTPD to find the failed node in the network. Hence, all the possible paths in the network might be required for the detection process. The maximum number of paths that can be formed with  $m$  sensor nodes for each path in the network with  $N$  sensor nodes can be given as

$$(3) \quad P = N(N - m),$$

where  $N$  is the number of nodes in the network and  $m$  is the number of nodes in each path that is three in our case. The analysis time of the failed node detection process is the time required to measure the RTPD times of all the paths in the network. It is the addition of all RTPD times of paths. The equation for analysis time with a  $P$  number of paths is given by

$$(4) \quad \tau_{\text{ANL}} = \tau_{\text{RTPD1}} + \tau_{\text{RTPD2}} + \tau_{\text{RTPD3}} + \dots + \tau_{(\text{RTPD}-P)},$$

$$(5) \quad \tau_{\text{ANL}} = \sum_i^P \tau_{\text{RTPD}} - i.$$

The time required for fault detection process increases exponentially with the number of paths in the network as the number of paths depends on the number of nodes in the network. The efficiency of the failed node detection process requires optimization of the number of paths in the network.

### 3.3. Paths in detection of failed node: Our contribution

This paper is concerned with the theoretical and experimental study of how sensor node failures can be detected. Various research related to the topic is reviewed and the conclusion was that there is a need for efficient and a scalable detection mechanism for failure in WSN. The work concentrates mainly on the failure of sensor nodes due to battery depletion. As per our literature survey, the failure in

sensor node detection mechanisms specific to cell depletion has not been proposed so far. Hence our algorithms contribute to efficient and scalable methodology to detect and reconfigure the topology of WSN using propagation delays.

Three methods to find the paths in the network have been found out, and each of them is better than the previous one. Hence, these are called optimization methods. These methods mainly concentrate on finding the paths so as to improve the performance of the detection process. The first method enhances the performance by curtailing the number of paths in (2) to fewer paths which is equal to the number of nodes in the entire network. The second method further improves the detection process by selecting unique paths from the set formed by the first optimization method. The algorithm promotes the performance to a greater extent since redundancy is removed. We further tried to improve the performance and finally came up with a solution to include all the nodes in single path. And also we dynamically find the “failed “nodes by waiting for acknowledgment packet from the nodes; consequently check the nodes’ energy and conclude the failure of nodes in the network. We calculated the probability of failure of nodes in the network and found network coverage based on that likelihood. Also, Shannon’s entropy model has been applied to our algorithm. The resulting analysis shows that the proposed methodologies have proved to be efficient.

Equation (2) gives all possible paths in the network amounting to replication of paths and thus affecting the detection performance. Repetition of nodes in the paths must be avoided to obtain better performance.

### 3.3.1. Linear selection of paths in the network

Instead of including (Figs 2 and 3) all possible paths in the detection process, we select the nodes in a linear fashion such that the number of paths in the network can be reduced. Linear selection of paths gives the number of paths that are equal to the number of nodes in the network, sufficient for failed node detection process as it includes all the nodes in one or the other paths. So linear selection of paths in a wireless sensor network with  $N$  number of nodes is given by

$$(6) \quad P_L = N.$$

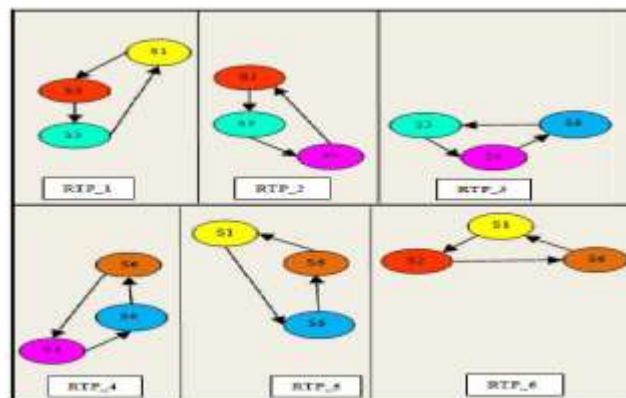


Fig. 2. Example of linear selection of paths in a network of six nodes

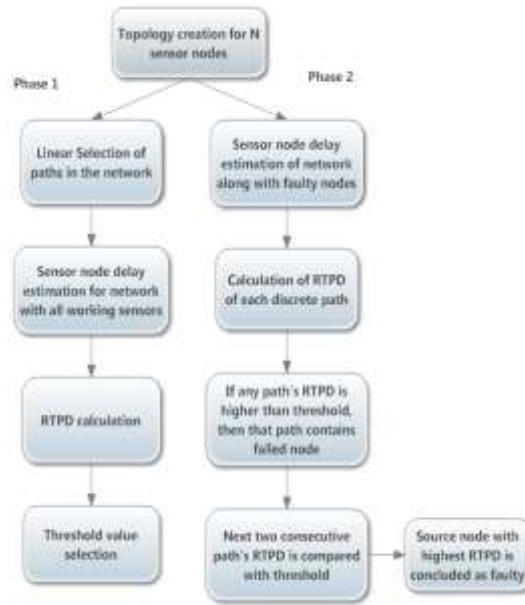


Fig. 3. Flowchart for working on linear selection of path algorithm

**Algorithm 1. Failed node detection algorithm using linear selection of paths**

**Step 1.** Select any node  $S_i$  from the network with  $N$  sensor nodes. The sensor nodes are  $S_1, S_2, S_3, \dots, S_N$ , where  $i = 1, 2, \dots, N$ .

**Step 2.**  $P_i$  can be formed by selecting three nodes in continuum as  $S_i, S_{i+1}, S_{i+2}$ .

**Step 3.** Transmit packet from  $S_i$  to  $S_{i+1}$ ,  $S_{i+1}$  to  $S_{i+2}$  and  $S_{i+2}$  to  $S_i$ .

**Step 4.** Record sending time at sender and receiving time at receiver and find transmission delay for each sensor node.

**Step 5.** Find round trip propagation delay by adding delay corresponding to nodes in each path. So the RTPD for  $P_i$  is  $\tau_i = \tau_1 + \tau_2 + \tau_3$ , where  $\tau_1$  corresponds to propagation delay of  $S_i$ ,  $\tau_2$  corresponds to propagation delay of  $S_{i+1}$ , and  $\tau_3$  corresponds to propagation delay of  $S_{i+2}$ .

**Step 6.** Compare  $\tau_i$  with threshold. If  $\tau_i$  is greater than threshold then, the path contains a node which is failed.

**Step 7.** Compare the RTPD of next path, i.e.,  $\tau_{i+1}$  with Threshold and  $\tau_i$ . If it is greater, then conclude  $S_{i+1}$  as failed.

**Step 8.** Find next path,  $P_{i+1}$  which consists of  $S_{i+1}, S_{i+2}$ , and  $S_{i+3}$ , if it is linear path. Repeat all the steps from Step 4 till all the paths are traversed in linear paths collection.

The analysis time with a linear selection of paths is given by,

$$(7) \quad \tau_{ANL} = N\tau_{RTPD},$$

where  $\tau_{\text{RTPD}}$  propagation delay time and  $N$  is a round trip is the number of sensor nodes in the network. RTPD for each path is considered to be almost equal since the distance between the nodes is assumed to be similar in a circular topology of the network.

But when the number of nodes in the network increases, the number of paths formed in order to find failed nodes also increases which hinders the performance of the algorithm. So to overcome the disadvantages of linear selection of paths, discrete selection of paths is proposed.

### 3.3.2. Discrete Selection of paths in the network

The second level of optimization is done to improve the performance of the detection process. With a linear selection of the path, the number of paths will be equal to the number of sensors in the network. With a vast sensor network, detection process becomes slower. So we need to curtail up the number of paths involved in the failure detection process. We tried to remove some of the redundant paths from the set of first level optimization.

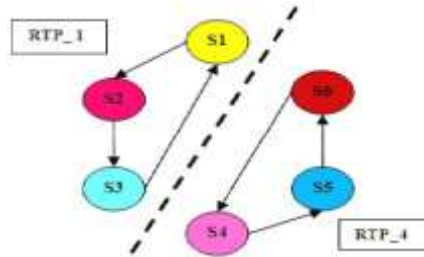


Fig. 4. Example of discrete selection of paths in network of six nodes

#### Algorithm 2. Failed node detection algorithm using discrete selection of paths

**Step 0.** Perform the operation till Step 5 to find the threshold value of RTPD.

**Step 1.** Select any node  $S_i$  from the network with  $N$  sensor nodes. The sensor nodes are  $S_1, S_2, S_3, \dots, S_N$ , where  $i = 1, 2, \dots, N$ .

**Step 2.** Form discrete paths where none of the nodes are repeatedly included in any of the paths. Discrete Path  $P_i$  can be formed by selecting three nodes in continuum as  $S_i, S_{i+1}, S_{i+2}$ .

**Step 3.** Transmit information packet from  $S_i$  to  $S_{i+1}$ ,  $S_{i+1}$  to  $S_{i+2}$  and  $S_{i+2}$  to  $S_i$ .

**Step 4.** Record sending time at sender and receiving time at receiver and find transmission delay for each sensor node and store it in the database.

**Step 5.** Find round trip propagation delay by adding delay corresponding to nodes in each path. So the RTPD for  $P_i$  is  $\tau_i = \tau_1 + \tau_2 + \tau_3$ , where  $\tau_1$  corresponds to propagation delay of  $S_i$ ,  $\tau_2$  corresponds to propagation delay of  $S_{i+1}$ , and  $\tau_3$  corresponds to propagation delay of  $S_{i+2}$ .

**Step 6.** Compare  $\tau_i$  with threshold. If  $\tau_i$  is greater than threshold then, the path contains a node which is failed.



**Step 7.** Find two more paths corresponding to next two nodes of source node in the path marked as failed. Calculate and compare the RTPD of those two paths, i.e.,  $RTPD_{i+1}$  and  $RTPD_{i+2}$  with threshold and  $\tau_i$ . If it is greater, then conclude source node of the particular path as failed node.

**Step 8.** Find next path,  $P_{i+1}$  which consists of  $S_{i+1}$ ,  $S_{i+2}$ , and  $S_{i+3}$ , if it is discrete path. Repeat all the steps from Step 4 till all the paths are traversed in discrete paths collection.

So this method of selection of paths includes unique paths by ignoring paths containing nodes repetitive from the set of paths in the first level of optimization. A path in the discrete selection in wireless sensor network is given by,

$$(8) \quad P_D = \frac{N}{m} \text{ if } N \text{ is divisible by } m,$$

$$(9) \quad P_D = \frac{N}{m} + 1 \text{ otherwise,}$$

where  $N$  is the number of nodes in the network and  $m$  is the number of nodes in each path. Here we have considered  $m$  as three since it is the least possible number of nodes in a path that can contain just to make delay incurred in the path minimal.

Analysis time required for the algorithm is given by

$$(10) \quad \tau_{ANL} = \frac{N}{m} \tau_{RTPD}.$$

By comparing (6) and (9), the analysis time of the method is reduced to some extent. But still, there is a scope for improvement. Hence, we tried to reduce the analysis time much more.

### 3.3.3. A single path with energy comparison

A single path with energy comparison is the third level of optimization, which includes all the nodes in a single path with the hope to reduce the time required for the detection process. In this method, a parent node is selected which has higher energy. The parent node initiates the algorithm to detect the failed sensor node. It transmits an information packet to the next node in the topology and waits for the acknowledgment. If an acknowledgment is not received within a particular period, then the energy of it is checked to determine whether it is failing or not. In this algorithm,  $P_S = 1$ . **Error! Reference source not found.** is used as mentioned in the title of the methodology. Analysis time of this method is given by

$$(11) \quad \tau_{ANL} = \tau_{RTPD},$$

which can be least possible time for the algorithm to detect a failure in nodes. When compared to (9) and (6), there is a lot of improvement in the analysis time.

### 3.4. Energy model

In our work, we are interested in energy lost during transmission and reception of data packets. Failed sensor node's energy will be depleted more than the standard node since battery failure case is considered. Energy drained while transmission of packets is given by

$$(12) \quad E_{TX(K,d)} = E_{elec}K + \varepsilon_{amp}Kd^2,$$

where  $K$  is the number of bits sent;  $\varepsilon_{amp}$  is energy consumed per bit for a particular area;  $E_{elec}$  is the energy required for processing the bits for transmission.

Energy drained while reception of a packet is given by

$$(13) \quad E_{RX(K,d)} = E_{elec}K,$$

where,  $K$  is the number of bits received,  $E_{elec}$  is the energy required for processing the bits after the reception.

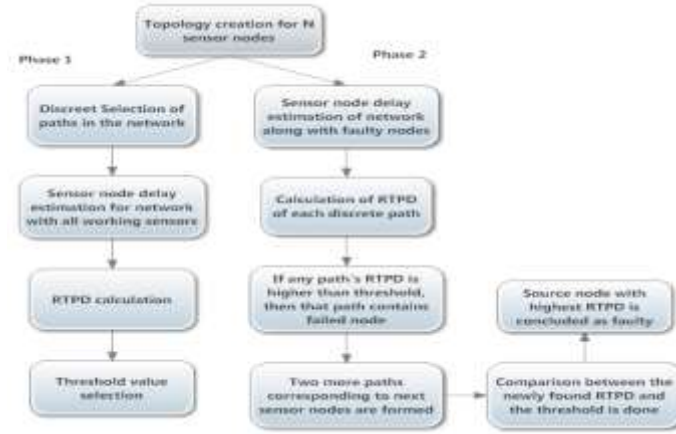


Fig. 5. Flowchart for working on a discrete selection of paths algorithm

Table 1. RTPD value of all discrete paths in the network of 30 nodes

Sl_no	Path number	Sequence of sensor nodes in path	Round trip propagation delay
1	P_0	$N_0 - N_1 - N_2$	0.02301580
2	P_3	$N_3 - N_4 - N_5$	0.00914563
3	P_6	$N_6 - N_7 - N_8$	0.00896455
4	P_9	$N_9 - N_{10} - N_{11}$	0.00938548
5	P_12	$N_{12} - N_{13} - N_{14}$	0.00994443
6	P_15	$N_{15} - N_{16} - N_{17}$	0.00938475
7	P_18	$N_{18} - N_{19} - N_{20}$	0.04818960
8	P_21	$N_{21} - N_{22} - N_{23}$	0.00956423
9	P_24	$N_{24} - N_{25} - N_{26}$	0.00950443
10	P_27	$N_{27} - N_{28} - N_{29}$	50.0088000

### Algorithm 3. Failed node detection algorithm with Single path and energy comparison

**Step 1.** Run the procedure for particular interval of time  $\alpha$ , till all the nodes are working correctly.

**Step 2.** Form a path  $P$ , which includes all the nodes,  $N_i$ ,  $i = 0, 1, 2, \dots, n$ .

**Step 3.** Select a node as parent node  $N_p$  based on its energy and  $N_p$  transmits the information packet to next consecutive node  $N_i$ .

**Step 4.**  $N_p$  waits for  $W_{ack}$  time to receive acknowledgement from  $N_i$ .

**Step 5.** Parent node,  $N_p$  checks for wait time, i.e., if  $W_{ack} < \text{Transmission to } N_i \text{ start time} + \text{current time}$ . If yes, then check the energy of the node ( $N_i$ ) to which it transmitted the information packet.

**Step 6.** If energy of  $N_i$  is drained beyond threshold value,  $\text{Thr}_{ene}$ , i.e., if  $N_{i_{ene}} < \text{Thr}_{ene}$ , then go to Step 7, else set  $N_i$  as parent node  $N_p$  and go to Step 8.

**Step 7.** Prune the node  $N_i$  from the network's database by recording it is failed node then remove it from network routing process.

**Step 8.** Now  $N_p$  sends the information packet to failed node's next node  $N_{i+1}$  and go to Step 5.

**Step 9.** Perform the same process till all the nodes are traversed.

**Step 10.** Declare that all nodes are working properly and Repeat the Step 1 after some fixed interval period of time. This procedure need to be executed continuously for efficient detection of failed sensor node.

Table 2. Comparison of paths for various methodologies

Paths	Number of nodes in the WSNs (N)					
	10	20	40	60	80	100
$P_M = N(N - m)$	70	340	1480	3420	6160	9700
$P_L = N$	10	20	40	60	80	100
$P_D = N/m$	4	7	14	20	28	34
$P_S = 1$	1	1	1	1	1	1

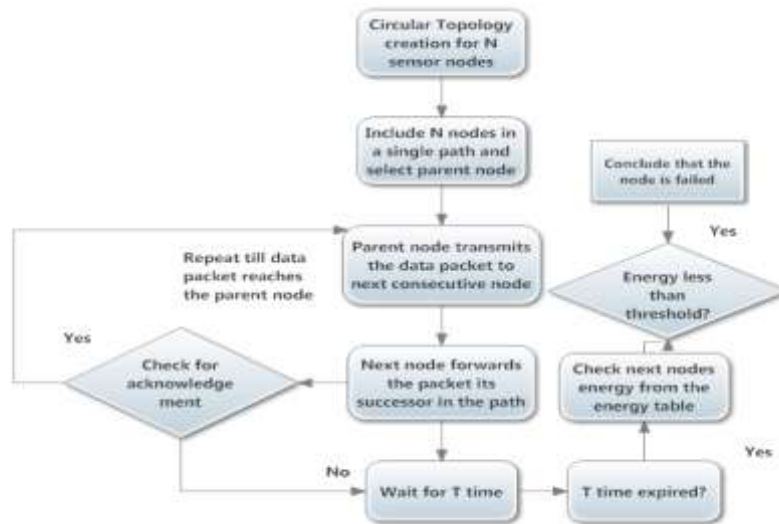


Fig. 6. Flowchart for working of single path with energy comparison algorithm

### 3.5. Entropy of nodes

Entropy has origin from thermodynamic parameter that represents the characterization of the state of the material. It is the measure of an amount of uncertainty in the system. We can relate Entropy theory to our work in the

uncertainty of sensor nodes getting failed. Currently, seven types of failure in sensor nodes are observed. They are bias, gain, drifting, precision degradation, complete failure, noise, and constant with noise. Since we are concentrating on the total failure of a node due to battery energy depletion, the probability of node getting failed would be  $(1/2) \times (1/7)$ . Using Shannon's entropy model, for a discrete random variable  $x$  whose value is  $x_i$ , the probability would be  $p(x_i)$ ,  $i = 1, 2, 3, \dots, n$ ,

$$(14) \quad H(x) = -\sum_{i=0}^n (p(x_i) \log(p(x_i))),$$

where  $H(x)$  is the entropy of nodes in the network.

### 3.6. Network coverage

Network coverage or coverage defines the ratio of the area covered by the network in the field of interest. Suppose the area of interest for deployment of sensor networks is  $A$ ,  $r_s$  is the sensing area of a node,  $N$  is the number of nodes that are deployed in the network. Then probability of node getting detected for its failure is given by,

$$(15) \quad p = \frac{\pi r_s^2}{A}.$$

The probability of an event to be detected by an arbitrary sensor node is given by  $p(1 - P_f)$ , where  $P_f$  is the probability of node failure. The probability that the event will not be detected by the absolute sensor is equal to  $1 - p(1 - P_f)$ . Then coverage fraction is

$$(16) \quad f_a = 1 - (1 - p(1 - P_f))^N.$$

Table 3. Network coverage versus number of nodes

Network coverage $F_a$				
Number of nodes	Probability of failure=0	Probability of failure=0.1	Probability of failure=0.5	Probability of failure=0.8
10	0.479053	0.442866	0.274387	0.119322
20	0.728614	0.689601	0.473486	0.224407
30	0.858622	0.827066	0.617955	0.316952
40	0.926349	0.903652	0.722783	0.398455
50	0.961632	0.946321	0.798848	0.470233
60	0.980012	0.970094	0.854041	0.533446
70	0.989575	0.983338	0.89409	0.589117
80	0.994575	0.990717	0.923151	0.638144
90	0.997174	0.994828	0.944237	0.681322
100	0.998527	0.997118	0.959537	0.719347

## 4. Experimental analysis

### 4.1. Implementation

The implementation is done using network simulator 2.35. It is simulated for 30 nodes and can be scaled to the finite number of nodes. The circular topology of the network has been considered here. NS2.35 has already incorporated AODV algorithm for routing the information packets in the network. We have revised this

algorithm to implement failed sensor nodes in the network. Failed sensor nodes behave unexpectedly and drop the routed packets. Also, their energy will be depleted more than any typical sensor node. Simulations are carried out by intentionally allowing few of the sensor nodes to fail and then algorithms are run to investigate the performance of methodologies.

#### 4.2. Experiments

The sensor node  $S_{12}$  is made to fail. First, let us consider the linear selection of paths algorithm. Here since we already have all paths with all nodes as a source node, let us consider  $P_{12}$ . The RTPD is calculated by adding the propagation delays of all the nodes in the path  $P_{12}$ . By considering all the nodes as working nodes, algorithm compares the RTPD with the threshold value are determined. A threshold value is the highest RTPD value of all the paths. If the RTPD of  $P_{12}$  is greater than the threshold value, then the path contains failed node. Further, it is compared with RTPD value of  $P_{13}$  and  $P_{14}$  and checked if it is greater than those value too. Finally, it is declared that  $S_{12}$  is failed.

Let us consider the same node with the discrete selection of paths algorithm. It is similar to the linear selection of paths, but all sets of paths are not calculated. So when any path's RTPD exceeds the value greater than the threshold, it is compared with threshold of two more consecutive paths thus formed. RTPD of these two newly formed paths is calculated and compared with the threshold. Finally failed sensor node is detected.

Both the algorithm implementations involve two phases. The first phase consists of threshold value calculation and second phase performs the actual failure detection process.

Now let us consider the third optimization of paths, i.e. single path with energy comparison. In this method, a single path incorporates all the nodes in the network, and this process is dynamic when compared to other two methodologies. Here node  $S_{11}$  sends the information packet and waits for a particular interval of time to get acknowledgment. If it does not, then the energy of the node  $S_{12}$  is checked. If it is less than energy threshold, then the node is said to be failed. This algorithm is proved to be efficient and scalable to any number of nodes.

### 5. Results

This section consists of experimental results obtained during simulation of the algorithms proposed. The throughput is calculated and compared with all the optimization algorithms after simulation. Fig. 7 shows the graph. Here it is clear that the performance of the third optimization algorithm is stable and is low only that the sensor node failure regions. When the failed sensor nodes have to send the information packet to other nodes, packets are dropped, and hence there is a fall in the graph.

Table 4. Comparison of the number of transmissions between all optimization task

Algorithm	$m$ (No of nodes in the path)	No of paths in every iteration	No of info packet transmissions
Multiple Selection	3	$N(N-m)$	$3N(N-m)$
Linear Selection	3	$N$	$3N$
Discrete Selection	3	$N/m$	$3N/m$
Single path selection	$N$	1	$N$

Table 4 shows the number of transmissions that can happen during a single iteration of discovery process algorithm execution. We observe that the optimization nominally affects the performance of the system.

Fig. 8 shows the average performance comparison between these three algorithms. In the figure, it shows that the mean throughput of linear selection has been less since; it has multiple transfers of information packets and droppings

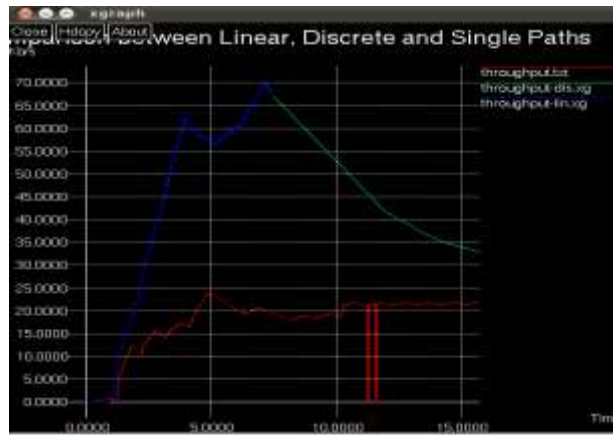


Fig. 7. Throughput comparison between all optimization algorithms

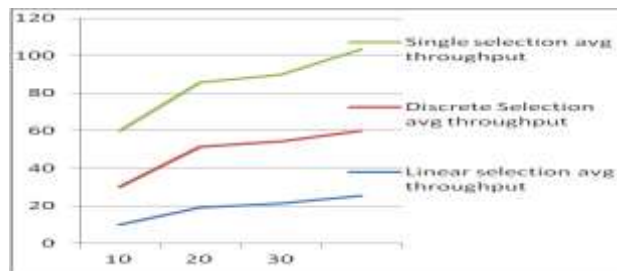


Fig. 8. Average throughput comparison between all the optimization algorithms

Fig. 9 shows the RTPD comparison between all three algorithms. For each path in the network, RTPD is calculated. So for first optimization algorithm, we can have  $N$  number of RTPD values, where  $N$  is the number of nodes in the network. For the second optimization algorithm, we can have  $N/3$  RTPD values. And for the third optimization algorithm, we can have just one RTPD value.

So for a network of 30 sensor nodes, we can have 30 RTPD values for linear paths and 10 RTPD values for discrete paths. And third optimization algorithm will always run a single time, irrespective of the number of nodes in the network. For

comparison, third algorithm simulation is done for three times to compare with other two algorithms.

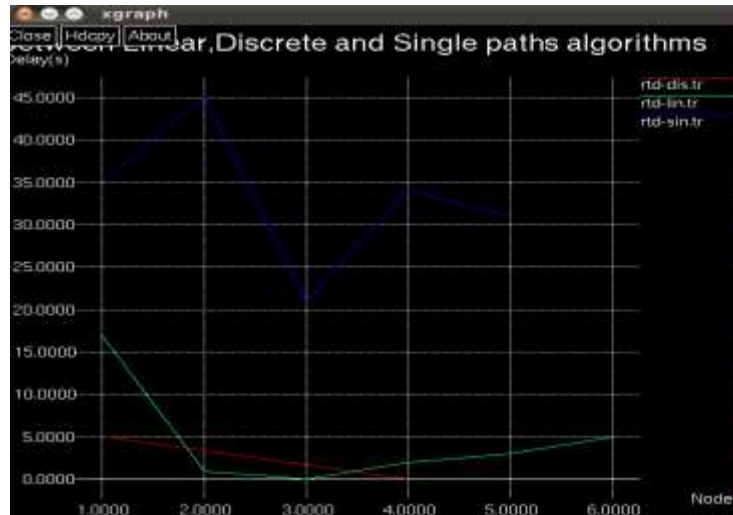


Fig. 9. RTPD comparison between the optimization algorithms

Fig. 10 shows the power of all the nodes pre-simulation of the algorithm. It clearly shows the few nodes have extremely less energy to transfer packets and are “failed” nodes. Thus, the energy of nodes plays a vital role in our detection process.

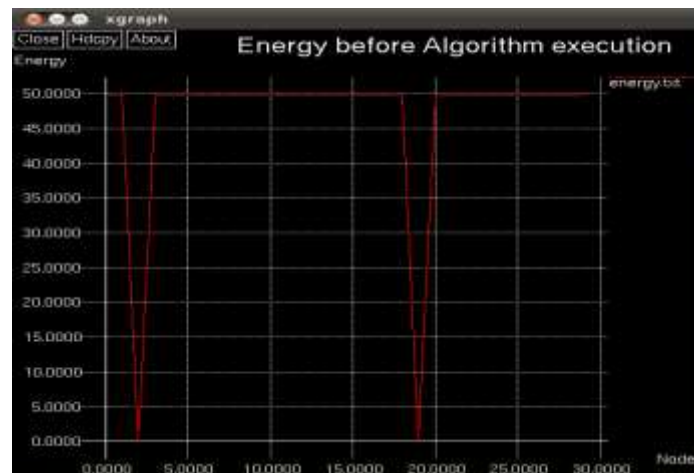


Fig. 10. Energy of all the nodes before transmission of information packets

Fig. 11 shows the energy of all the nodes post-simulation of the algorithm. Due to packet transmission, processing and reception the energy of all the nodes is drained to some extent. Even though the entire detection process is complete, none of the nodes energy is exhausted to the extent as in failed sensor nodes. Thus, it proves that the sensors whose energy is less than the threshold value, (in this experiment the threshold value of energy is considered as 5 kJ) are said to be failed.

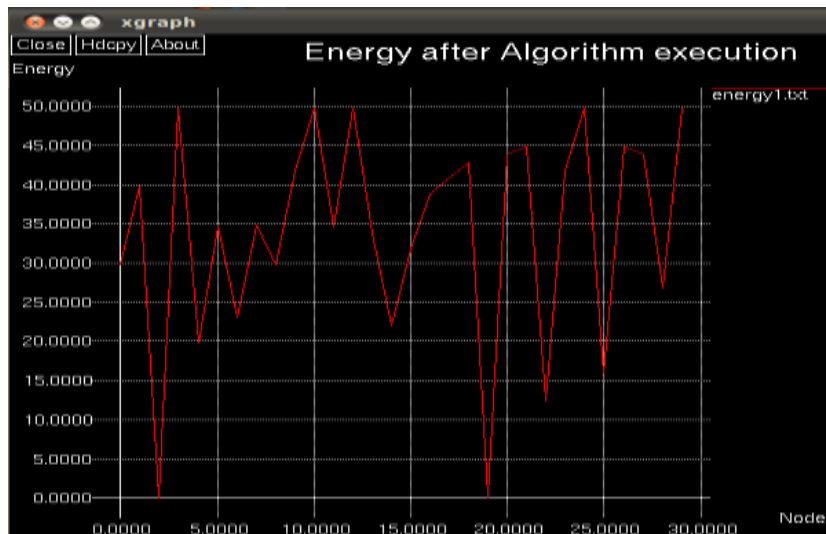


Fig. 11. Energy of all the nodes after transmission of information packets

## 6. Conclusion

Three methods of detecting the failed sensor node in the wireless sensor network are proposed, implemented and simulation results were analyzed. Simulations were done using NS2, which is a stable and efficient simulator, used in networking research work. Future work may include investigating other causes of failure in WSN and study of partial failure detection.

## 7. Future research work

This work could be scaled to mobile sensor nodes where the location of the nodes in the network changes randomly. The algorithms can be modified and extended to unstructured wireless sensor networks. Fast and effective mechanism to route the information of failure to the base station can also be considered in future work. Another approach which we would contemplate for future work is to consider a structured topology, like star or mesh, with dedicated optimal path to transmit data and another redundant path which connects all the nodes in the network, to ensure that sensors are alive. Each node drops probe message through the dedicated path to fortify other nodes that it is active.

## References

1. Lee, M. H., Y.-H. Cho. Fault Detection in Wireless Sensor Networks. – Elsevier Computer Communications, 2008, pp. 3469-3475.
2. Akbari, A., A. Dana, A. Khademzadeh, N. Beikmahdavi. Fault Detection and Recovery in Wireless Sensor Network Using Clustering. – IJWMN, Vol. 3, February 2011, No 1, pp. 130-138.



3. Song, C.-C., C.-F. Feng, C.-H. Wang, D.-C. Liaw. Simulation and Experimental Analysis of a ZigBee Sensor Network with Fault Detection and Reconfiguration Mechanism. – In: Proc. of 8th ASCC, May 2011, pp. 659-664
4. Mojoodi, A., M. Mehrani, F. Forootan, R. Farshidi. Redundancy Effect on Fault Tolerance in Wireless Sensor Networks. – Global J. Comput. Sci. Technol., Vol. **11**, April 2011, No 6, pp. 35-40.
5. Ahuja, S. S., R. Srinivasan, M. Krunz. Single-Link Failure Detection in All-Optical Networks Using Monitoring Cycles and Paths. – IEEE/ACM Trans. Netw., Vol. **17**, August 2009, No 4, pp. 1080-1093.
6. Duché, R. N., N. P. Sarwade. Sensor Node Failure or Malfunctioning Detection in Wireless Sensor Network. – ACEEE Int. J. Commun., Vol. **3**, March 2012, No 1, pp. 57-61.
7. Sarasvathi, V., Snehanshu Saha, N. Ch. S. N. Iyengar, Mahalaxmi Koti. Coefficient of Restitution Based Cross Layer Interference Aware Routing Protocol in Wireless Mesh Networks. – International Journal of Computer Networks and Information Security, Vol. **7**, No 3, pp. 177-186.
8. Sarasvathi, V., N. Ch. S. N. Iyengar, Snehanshu Saha. QoS Guaranteed Intelligent Routing Using Hybrid PSO-GA in Wireless Mesh Networks. – Cybernetics and Information Technologies, Vol. **15**, 2015, No 1, pp. 69-83.
9. Sarasvathi V, N. Ch. S. N. Iyengar, Snehanshu Saha. Interference Aware Channel Assignment Using Edge Coloring in Multi-Channel Multi-Radio Wireless Mesh Networks. – International Journal of Computer Networks and Information Security, Vol. **6**, 2014, No 1.
10. Akylidiz, I., W. Su, Y. Sankarasubramanian, E. Cayirci. A Survey on Sensor Networks. – IEEE Communication Mag., Vol. **40**, 2002, pp. 102-114.
11. Rajaravivarma, V., Y. Yang, T. Yang. An Overview of Wireless Sensor Network and Applications. – In: Proc. of 35th Southeastern Symposium on System Theory, Morgantown, WV, USA, 2003; pp. 432-436.
12. Sha, K., J. Gehlot, R. Greve. Multipath Routing Techniques in Wireless Sensor Networks: A Survey. – Wireless Personal Communication, Vol. **70**, 2013, No 2, pp. 807-829.
13. Asim, M., H. Mokhtar, M. Merabti. A Fault Management Architecture for Wireless Sensor Network. – In: Proc. of IWCMC, August 2008, pp. 1-7.
14. Younis, M., K. Akkaya. Strategies and Techniques for Node Placement in Wireless Sensor Networks: A Survey. – Ad Hoc Networks, Vol. **6**, 2008, No 4, pp. 621-655.
15. Jiang, P. A New Method for Node Fault Detection in Wireless Sensor Networks. – Sensors, Vol. **9**, 2009, No 2, pp. 1282-1294.
16. Chen, I., A. P. Speer, M. Eltoweissy. Adaptive Fault-Tolerant QoS Control Algorithms for Maximizing System Lifetime of Query-Based Wireless Sensor Networks. – IEEE Trans. Dependable Secure Computing., Vol. **8**, March/April 2011, No 2, pp. 1-35.
17. Boudhir, A. A., B. Mohamed, B. A. Mohamed. New Technique of Wireless Sensor Networks Localization Based on Energy Consumption. – Int. J. Computer Applications, Vol. **9**, November 2010, No 12, pp. 25-28.