Distributed Fault Tolerant Architecture for Wireless Sensor Network

Siba Mitra and Ajanta Das
Department of Computer Science and Engineering, Birla Institute of Technology
Mesra, Kolkata Campus, India
E-mail: sibamitra@bitmesra.ac.in, ajantadas@bitmesra.ac.in

Keywords: fault detection, fault recovery, fault tolerance, reliability, wireless sensor network

Received: December 25, 2016

Smart applications use wireless sensor network for surveillance of any physical property of that area, to realize the vision of ambient intelligence. Since wireless sensor network is resource constrained and for unattended deployment scenario faults are quite trivial. Reliability and dependability of the network depends on its fault detection, diagnosis and recovery techniques. Detecting faults in wireless sensor network is challenging and recovery of faulty nodes is very crucial task. In this research article, a distributed fault tolerant architecture is proposed. This paper also proposes fault recovery algorithms. Recovery actions are initiated based on fault diagnosis notification. The novelty of this paper is to perform recovery actions using data checkpoints and state checkpoints of the node, in a distributed manner. Data checkpoint helps to recover the old data and the state checkpoint tells the previous trust degree of the node. Moreover, the result section explains, that after replacement of a faulty node, the topology and connectivity between rests of the nodes are maintained in WSN.

Povzetek: Opisana je arhitektura brezžičnega senzorskega omrežja.

1 Introduction

The use of wireless sensor network (WSN) nowadays has seen a huge growth in the field of ambience intelligence. WSN is resource constrained in nature but can be integrated with any system by using Dynamic Adaptive System Infrastructure (DAiSI) proposed by Klus and Niebuhr (2009) in [11]. Component reconfiguration and dynamic integration can be done with the help of this. Another interesting application that uses WSN to detect and track presence of human and human motion in an environment is presented in the research of Graham et al. (2011) in [7]. It is also shown in the work that appropriate device placement scheme can improve network performance.

The unit of WSN is a tiny sensor node, which communicates to other sensor nodes through radio transmission. Small sensor nodes constituting of sensing unit, tiny memory, a microcontroller, a transceiver and an omni-directional antenna are deployed in the target area. Sensor nodes send relevant data to the nearest base station (BS), which is used for some meaningful decision making. Fault in WSN is trivial because of its resource constraints and unattended deployment scenario. Therefore to make the WSN reliable and dependable, fault tolerance must be implemented in it.

Various types of node faults are classified in the Figure 1. Faults in WSN can be permanent, transient or intermittent in nature. Fault management is the process to monitor the nodes, detect and diagnose fault and perform necessary recovery tasks to make WSN fault tolerant. Permanent failures generally have no option for recovery but for transient and intermittent faults the recovery actions should prevail. Proper recovery schedules should be there for occurred faults to make it fault tolerant and help application to make correct decision, even in presence of faults. Some of the critical factors in recovery process are the available residual energy of the sensor node, the network traffic scenario, connectivity issues and current topological structure of the WSN.

A distributed adaptive fault detection scheme for WSN is proposed in [28], where each node detect any unnatural event by fetching neighbor sensor nodes' reading with queries. A three-bit control packet exchange is done during the fault detection phase in order to reduce communication overhead. Here moving average filter was employed for implementing fault tolerance in WSN. The article claims to have reached high detection accuracy and low false alarm rate.

WSN may comprise of static or mobile sensor nodes. Borawake-Satao and Prasad (2017) [4] presents a study of effects of sensor node mobility on various performance parameters of WSN. A proposal of mobile sink with mobile agent mobility model for WSN is also presented. In [12] Kumar and Nagarajan (2013) proposed Incorporated Network Topological control and Key management (INTK) for relay nodes of WSN, for privacy and security measures in the network. The proposed scheme includes hierarchical routing architecture in WSN for better performance and security. Another novel research proposal by Mukherjee et al. (2016) [22] presents a model for disaster aware mobile Unmanned Aerial Vehicle (UAV) for flying Ad-hoc network. The nodes can perform collaborative job by relaying useful message in a post-disaster situation of any ecosystem.

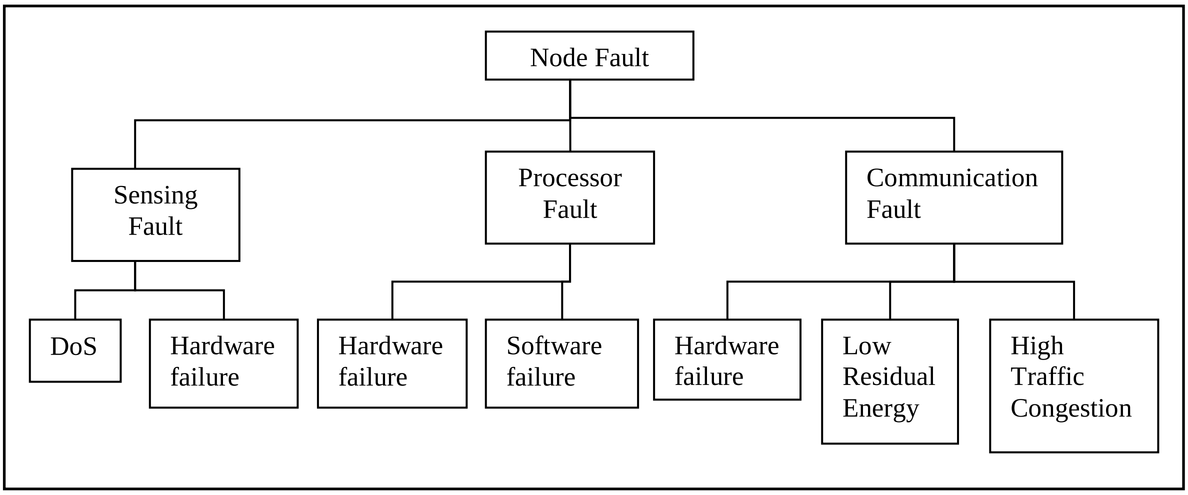

Figure 1: Node Fault Classification.

The analytical comparison presented in [3] by Bathla & Jindal 2016, where two distributed self-healing recovery techniques, Recovery by In-ward Motion (RIM) and Least Disruptive Topology Repair (LeDiR) are analyzed and compared with respect to their efficiency in various applications. Both the approaches are distributed in nature.

The RIM method aims to replace a failed node by a healthy node, by moving the latter towards the former's location. Here all nodes must have a 1-hop neighbor list and should be aware of their neighbor's locality and proximity. Now the goal of LeDiR is to restore the connectivity among the sensor nodes. But it also takes care that after the recovery action the shortest path length among the nodes is not extended compared to the pre-failure topology.

A fault recovery algorithm for WSN is proposed in [13] by Lakamana et al. 2015, which enhances the routing efficiency in the WSN. Battery depletion is a major issue and that is taken care of over here by reducing the number of node replacements and by reusing the historic routing paths. According to the authors' claim the network longevity is increased over here.

In WSN, now sensor node(s) when gets disconnected from the network due to some reason may generate partitions or isolations in the network, which is not good for reliability and dependability of the network. Moreover, it is crucial to maintain the connectivity throughout its longevity. So the objective of this research is to design a distributed fault tolerant architecture for WSN, which includes fault detection, diagnosis and recovery. However the architecture proposed here is an improved version of a fault tolerant framework already proposed in our previous research work available in [18] by Mitra & De Sarkar (2014). Moreover this article also proposes a novel fault recovery model, which is integrated with the proposed architecture. This research also proposes some algorithms for connectivity maintenance, and recovery tasks to be performed. The novelty of the proposed recovery technique is to initiate the recovery actions after proper diagnosis of the detected fault. Recovery tasks are done once after it gets notification from the diagnosis layer about the fault-type. The recovery model has two phases; the first one being set action and start recovery.

The remainder part of the article is sub-divided into sections namely, related work done in the current field, followed by proposed Distributed Fault Tolerant Architecture and supporting fault recovery architecture and algorithms; and then the results and discussions section is presented. Finally, the conclusion and the references to the article are presented.

2 Related work

This section mainly presents some of the valuable researches carried out by many scholars in the field of WSN. Many existing fault management techniques are available, which are used for fault tolerance in WSN. A review work of the same is presented in our previous research work, Mitra, De Sarkar and Roy (2012) in ref. [20] and a few of them are mentioned here also. Moreover in this article a study on some of the existing recovery schemes is presented. Data communication is important factor in WSN hence routing decision is significant. Leskovec, et al. (2005) [14] proposed a novel link quality estimation model for sensor network, which uses link quality map to estimate a link in sensor network. This work also optimizes power consumption of radio transmission signal, while scheduling the communication task and taking routing decision.

An analytical study and comparison of various recovery techniques are presented in our recent research work, in [19] (Mitra, Das & Mazumdar 2016). Some of those recovery schemes are also discussed briefly here. Among them CRAFT (Checkpoint/Recovery-based scheme for Fault Tolerance) [26] for WSN, proposed by Saleh, Eltoweissy and Agbaria (2007) is studied; another scheme proposed by Ma, Lin, Lv & Wang (2009) [16] called ABSR, which recovers some compromised sensor nodes in a heterogeneous sensor network. Various types of sensor nodes each playing specific role are used here. Reghunath, Kumar & Babu (2014) proposed Fault Node Recovery (FNR) algorithm, which is a combination of Genetic algorithm with Grade Diffusion algorithm. A rank based replacement strategy for the sensor nodes is presented in [25].

In [6] Chen, Kher & Somani (2006) proposed DLFS (distributed localized fault sensors) detection algorithm, for locating and identifying faulty nodes in WSN. Each node can be either in good health or can be faulty depending upon the node behavior. The technique here uses probabilistic approach. The implemention of the algorithm claim that execution complexity of the same is much low and detection accuracy is high. Haboush, Mohanty, Pattanayak and Al-Tarazi (2014) [8] have proposed a faulty node replacement algorithm for hybrid WSN. Mobile sensor nodes are considered over here. Any node having low residual energy may seek a replacement; after replacement maintenance of the topology etc. are taken care of. Redundancy is used to avoid faulty results and also adaptive threshold policy is employed for rectification of the faults and optimizing the network lifetime. The research in [2] Akbari et al. (2010) presents a survey of faults in WSN due to energy crunch and the role of cellular architecture and clustering for network sustain purpose. The cluster-based fault detection and recovery techniques was observed to be quite efficient, robust and fast for WSN sustain and longevity. Another cluster maintenance technique is designed by them for nodes having energy crunch as mentioned in [1] (Akbari, Dana, Khademzadeh & Beikmahdavi 2011). First of all, nodes with highest residual energy are selected as primary cluster head, and nodes second in residual energy becomes the secondary cluster head. So the technique is energy aware in nature and consequentially selects the cluster head as per the nodes' residual energy.

An FNR algorithm is proposed by Brahme, Gadadare, Kulkarni, Surana & Marathe (2014) in [5], for fault recovery in WSN to enhance network lifetime. Researchers employed genetic algorithm and grade diffusion algorithm for designing the scheme. Moreover researchers, Mishal, Narke, Shinde, Zaware & Salve (2015) in [17] have worked upon FNR and improved it performing lesser number of node replacement for fault recovery, and basically old routing paths are reused; however better result is claimed over here. A proposal on a distributed fault detection algorithm for detecting coverage holes in the WSN is presented in Kang et al. 2013 [9]. The research do not maintain any node coordinates. The critical information of a node can be collected from the neighbors and that can be used for detection and recovery purpose for WSN. On demand checkpoint based recovery technique for WSN is proposed in [23] by Nithilan & Renold (2015). In this scheme checkpoint coordination and non-blocking checkpoint is used for consistency and some backup nodes maintains and checks the health of a node by monitoring the checkpoints. A localized tree based method for fault detection is proposed by Wan, Wu & Xu (2008) [27]. The recovery scheme uses elected new parent technique for avoiding isolation of children node of the tree. This technique enhances the network lifetime.

The main objective of this research work is to design a distributed fault tolerant architecture for WSN, with intrinsic parts for fault detection, diagnosis and recovery. In this research we mainly concentrate to propose a distributed fault recovery model for WSN with a set of algorithms, which are employed for performing node, data and network recovery. For fault detection, existing detection algorithm proposed in our previous work in [18] is used. Thereafter the proposed recovery technique is employed to maintain the fault tolerance. The major job is to increase the reliability and dependability of the WSN for correct decision making. The novelty of this research work is to perform recovery actions, using data checkpoints and state checkpoints of the node, in a distributed manner. Also topology maintenance is being performed by each node during the recovery process.

3 Proposed distributed fault tolerant architecture for WSN

This section details on a proposal of a fault tolerant framework for WSN. Event detection is important for implementing fault tolerance in WSN, where the event can be presence of hole in the network. A distributed, lightweight, hole detection algorithm proposed by Nguyen et al. (2016) in [24] monitors and reports about any hole in the network. The present proposal is an improvisation of an already proposed framework mentioned in Mitra et al (2014). The architecture as mentioned in Figure 2 can be embedded in each sensor node of WSN and the node can independently perform fault management in a distributed way.

In a centralized system fault management scheme there is a central manager, who monitors and controls the network. So each node has to report the central manager with relevant data for fault tolerance in the WSN. Therefore too much of communication will result and in the WSN huge overhead will be incurred in terms of energy and bandwidth, which may affect network performance. In centralized approach the traffic flow is towards a single central manager creating overheads and resulting in bottlenecking. However this is not desirable in WSN since it is resource constrained and infrastructure less. This critical bottleneck problem can be avoided in the distributed architecture of fault management scheme. In distributed fault management, the network is partitioned and self fault management is implemented. Moreover in comparison with the centralized system the communication cost is less in distributed system. Therefore this research work mainly aims for distributed architecture. The proposed architecture has three main phases viz. *Fault Detection*, *Fault Diagnosis* and *Fault Recovery* respectively.

3.1 Fault detection

The detection phase has three significant tasks namely *Node and Link Monitoring*, *Fault Isolation* and *Fault*

Prediction. Fault detection algorithm is already proposed in our previous work presented in Mitra et al. (2014). A brief discussion is presented hereafter. All the tasks are computed in an energy aware mode. In the monitoring stage the sensor node listener carefully monitors and examines the health of the sensor node and detects if any unnatural event occurs; and then it scans the attributes of the event; it also evaluates some useful parameter-value required for detecting faults in the node. First of all a neighbor table for each node is created and after that node performs self-checking. Sensor nodes evaluates own tendency by comparing the average of neighbors' reading with the own read value. Again the nodes do a similar comparison with its own previous read value and current value. The tendency of the node quantizes whether it is trustworthy or not. If a node is not trustworthy then the trust degree (TD) value is zero and if it is trustworthy then the TD value is one. So the TD value isolates rather detects the fault in the WSN.

In the *Prediction* module the residual energy analysis of the node is carried out and any fault-to-be are forecasted. The forecast is done on the basis of some comparative study of the fault evaluation parameters namely residual energy of the node. If the residual energy of the node goes below a threshold then the built-in fault predictor invokes two actions; firstly it broadcast the information of its low energy state. Secondly some query packets are broadcasted asking for a node with high residual energy for offloading its own responsibility. Finally the node is sent to sleep mode.

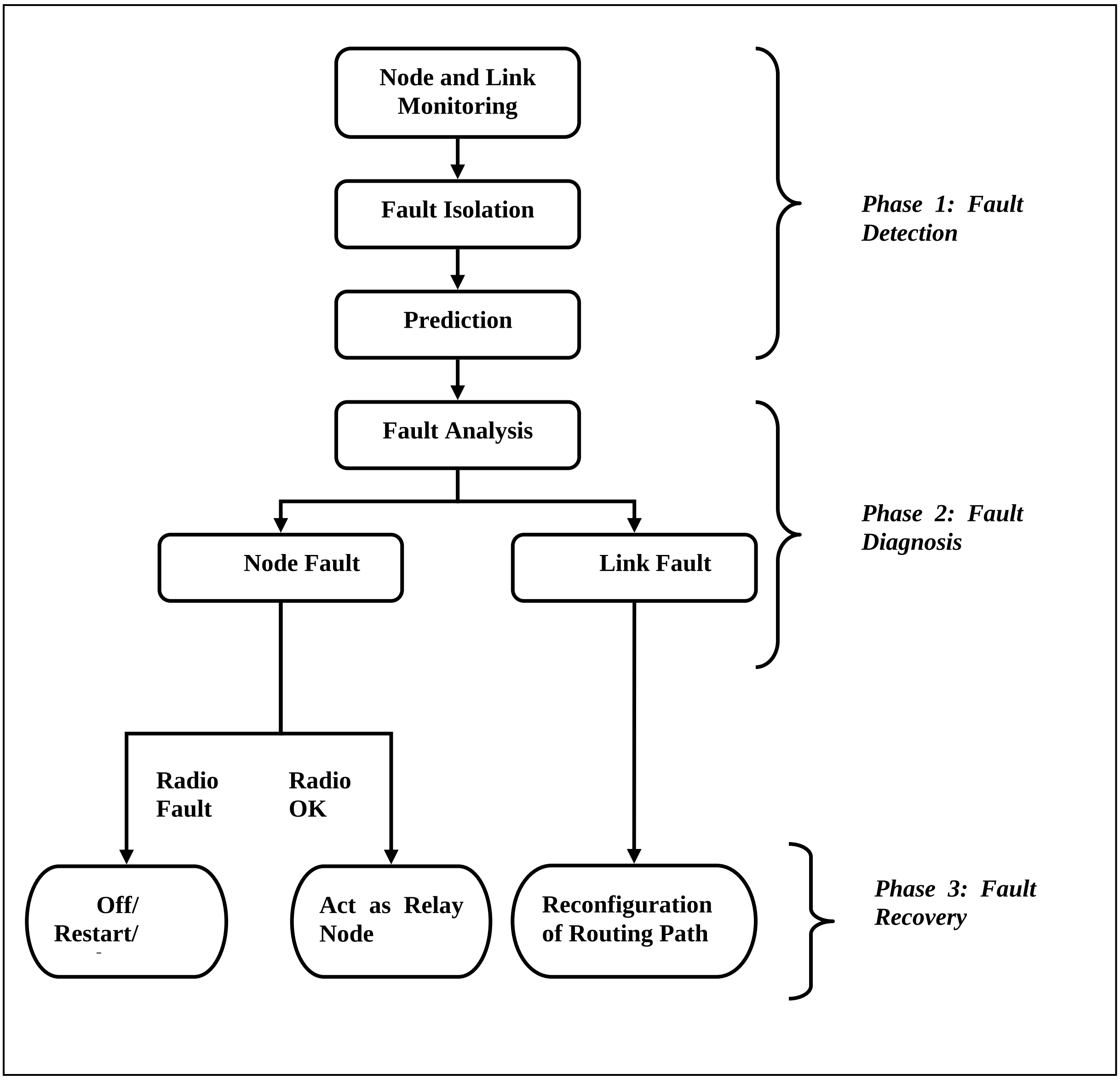

Figure 2: Distributed Fault Tolerant Architecture for WSN.

3.2 Fault diagnosis

The second phase of the fault tolerant architecture is *fault diagnosis* and it is done after the analysis of the occurred event. Fault analysis is a reactive process, and the fault category in WSN can be either a node fault or communication fault. For diagnosing the node fault, the assigned TD value is taken into consideration as available in [18]. Evaluation of TD value of a node is computed on the basis of self analysis and neighbor analysis for a fixed number of iterates. And depending on the iteration count the decision of node fault is finalized. Now for communication fault diagnosis, two critical parameters received signal strength (RSS) and link utilization of the sensor nodes are taken into account. The average RSS of all the neighbors of the sensor node are computed for communication fault analysis. Moreover the sensor node also computes the average link utilization parameter to check self performance. Once self-fault diagnosis is completed then a notification is forwarded to the next phase i.e. *Fault Recovery.*

4 Proposed fault recovery scheme

This section presents the fault recovery scheme for WSN. This scheme can be integrated in each sensor node such that distributed fault recovery is possible. The recovery process is invoked when a sensor node is suffering of some fault. The next sub-sections present the network model and the fault recovery model.

4.1 Network model

The WSN model in this research work can be represented as a graph structure, $G(S, E)$, where S is a set of sensor nodes $S = \{S_1, S_2, \ldots, S_n\}$, which is deployed in random or planned way in the target area. The area can be represented as a two-dimensional plane whose origin is (x_0, y_0). Now $E = \{E_1, E_2, \ldots, E_n\}$ is a set of communication links in between a pair of nodes S_i and S_j, which transmits within its communication range but has a sensing range lesser than communication range, in [15] given by $R_S < R_C$ where R_S and R_C, are sensing range and communication range respectively. The necessary condition for a node S_i to transmit signal to a node S_j is the Euclidean distance between the two nodes should conform Equation 1. Each node maintains a list of neighbors, which may dynamically change with time as per availability of the node in the communication process. It is very general to perform low power transmission in WSN, where node's transmission power is directly proportional to the distance. To send data with good quality signal strength a node may have to adjust its transmission power. For the current problem scope if $P_{i,j}$ is the power of transmission for communication of S_i and S_j. It is quite obvious that Equation 2 will satisfy if and only if Equation 3 is true. Moreover the maximum value of transmission power is also limited. The assumption is that any node S_i will perform low power transmission for nodes within R_C and may sometimes, as required, perform high power communication with S_j if and only if Equation 4 is true.

$$\|S_i S_j\| \le R_C \qquad \textit{Equation (1)}$$

$$P_{i,j} \le P_{k,r} \qquad \textit{Equation (2)}$$

$$\|S_i S_j\| \le \|S_k S_r\| \qquad \textit{Equation (3)}$$

$$R_S < \|S_i S_j\| < R_C \qquad \textit{Equation (4)}$$

4.2 Connectivity issue

Now not all the nodes can directly transmit data to the sink or BS; so any node unable to do the same will employ some intermediary forwarding parent nodes to send the data to the BS. At the run time one or more sensor nodes may not work properly due to faults and then the recovery actions of the nodes may be initiated to recover the node, data or network. Any recovering node may need to stop its scheduled tasks for self-recovery. In that case other affected neighbor nodes may have to update their own neighbor list and exclude the recovering node from any current activity. This scenario is explained in Figure 3.

It is well understandable from the figure that the normal nodes need to maintain the connectivity even in absence of the faulty nodes marked as black nodes. Hence the normal nodes have to find alternate suitable nodes within its communication range for forwarding data. But if it is unable to find one then it should perform a high power transmission to the nodes, which are in its communication range, rather than getting isolated. In the figure the dotted arrows demarcate the unstable or sometimes unavailable links. To transmit in high power the node should increase its transmission power by a multiplicative factor, somewhat proportionate to the increment in the distance given by $|D_{i,j} - D_{i,k}|$, where $D_{i,j}$ and $D_{i,k}$ are explained in Equation 5 and 6. In the equations i-th node transmits to k-th node in the place of recovering j-th node.

$$D_{i,j} = \|S_i S_j\| \qquad \textit{Equation (5)}$$

$$D_{i,k} = \|S_i S_k\| \qquad \textit{Equation (6)}$$

4.3 Fault recovery model

The proposed fault recovery model is depicted in Figure 4, where there is a *Fault Recovery Process*, which has two main phases viz. *Set Action* and *Start Recovery*. The *Fault Recovery Process* gets notification from the fault diagnosis layer along with the information on the fault type; depending upon that the *Set Action* decides on what kind of recovery activity has to be invoked. Again *Start Recovery* actually begins the specific recovery task. Faults can be due to hardware or software failure, bad link quality or power depletion of the node.

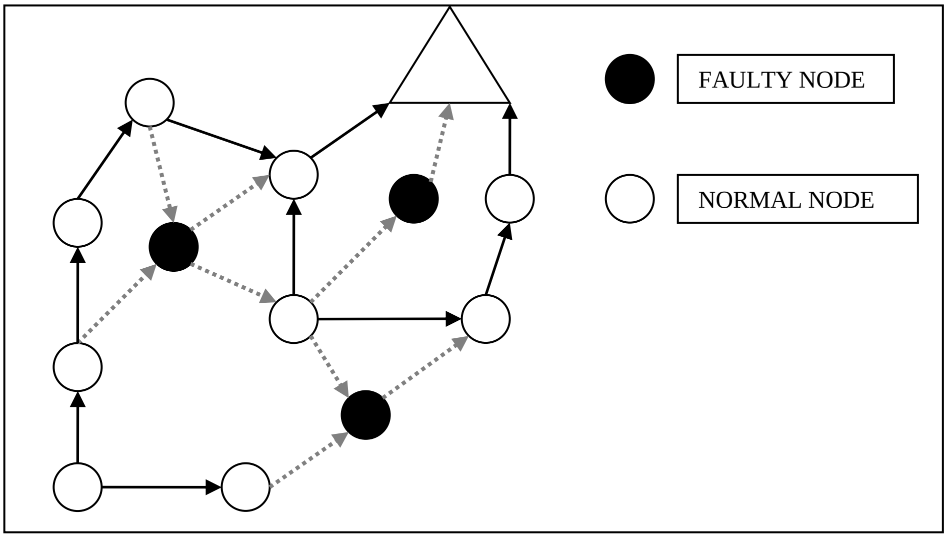

Figure 3: Connectivity Issue in WSN.

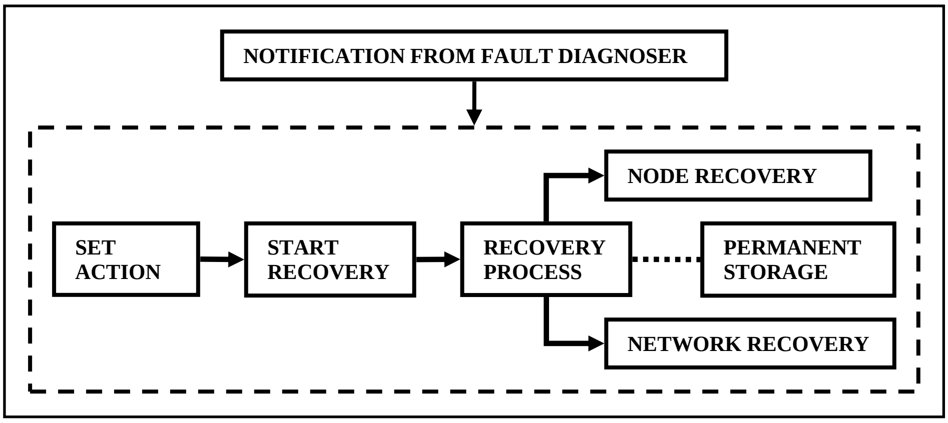

Figure 4: Fault Recovery Model for WSN.

The Recovery Process performs and maintains node recovery and network recovery and it also communicates with the permanent storage for any kind of query checking. The permanent storage contains node status and data checkpoint before occurrence of the fault. The Recovery Process fetches the necessary information to perform the recovery tasks smoothly. It also performs various types of communication before the node gets reinitialized. Node recovery means data recovery from the node and also checking the node state and performing activities to preserve the node functionality. Network recovery deals with reconfiguring the network by performing the path quality estimation already proposed by Mitra, Roy & Das (2015) in [21]. The recovery jobs are vividly explained later in Algorithm 3.

5 Proposed algorithms for fault recovery

The proposed fault recovery algorithm consists of various parts, where each node carry out some self-checking task and some of them are already mentioned in [18]. Since the total process is being carried out in a distributed atmosphere so the nodes perform self-evaluation and self-recovery. All the symbols and notations used in the algorithm are mentioned in Table 1. The proposed fault recovery scheme uses some sort of check pointing for performing the data recovery task. Each node maintains a data checkpoint and a state checkpoint by using two variables TD (Trust Degree) and DCkpt (Data Checkpoint) respectively, in the permanent storage of the node i.e. even if the node is restarted the data remains intact for future reference as mentioned in Algorithm 1 and presented in Figure 5. TD is already proposed, explained and used in our prior research mentioned in [18]. This previous work also presented a novel fault detection scheme, which is used over here to detect and diagnose faults. Now HF means sensing unit or hardware failure, where the node is unable to sense any ambient signal, or it may be transceiver fault that occurs when transmitter or receiver is not in working mode and microcontroller fault means when a node cannot perform its computations at par.

Each node has a delivered packet counter as DPC, which keeps the count of the delivered packets. Whenever a node delivers 200 packets it stores TD, as state checkpoint, in the permanent storage and stores the current read value of the node in DCkpt, as data checkpoint, in permanent memory. After completion of these steps the DPC is reinitialized to zero so that it can again count the next set of 100 and 200 delivered packets respectively. The checkpoint creation process continues for each node until it goes to recovery state. When a node gets a notification from the fault diagnosis layer, that a fault has occurred, it fetches the fault type and performs some internal necessary actions that is mentioned in Algorithm 2 and presented in Figure 6. As mentioned the fault type can be either hardware fault (HF) or software fault (SF). Depending upon fault-type the recovery process is initiated as mentioned in Algorithm 3 and presented in Figure 7.

Notation	Meaning
S_i	i-th node
S_r	Node at Recovery mode
S_r.NBR	Neighbor list of S_r
S_i.CURR-VAL	Current reading of S_i
DCkpt	Data Checkpoint
DPC	Delivered packet count
TD	Trust degree
CR	Communication range
$P_{j,k}$	Power to transmit data from S_j to S_k
PRR	Packet reception ratio
PDR	Packet delivery ratio

Table 1: Symbols and Notations.

In all these cases the node needs a third party or human intervention to get the problem fixed. Software failure refers to logical or runtime faults in the software, which is again needs third party intervention. If the packet reception ratio (PRR) and packet delivery ratio (PDR) are much low then there must be some disturbances in data transmission and receiving; hence a communication failure may occur in near future. So the nodes goes for a self-recovery process. Finally the node has to be shut down if the residual energy is much less than the threshold value, which may be specified as per application requirement. The node recovery module for any faulty node starts with low-power transmission of probe packets to the neighbors, which again go for topology maintenance as mentioned in Figure 8 as Algorithm 4. The recovery activity takes place by reinitializing the sensor nodes so that it releases all its resources and take a fresh start. The last state checkpoint and data checkpoint is recovered from the permanent memory. Data checkpoint helps to recover the old data and the state checkpoint tells the previous trust degree of the node.

```
Create Checkpoint ( )
{
    For each node $S_i$ do this
    {
        Initialize DPC=0;
        For each packet delivery
        {
            DPC++;
            If (DPC = 200)
            {
                Store TD in permanent storage;
                Store $S_i$.CURR-VAL in DCkpt;
                Set DPC=0;
            }
        }
    }
}
```

Figure 5: Algorithm 1.

```
For each node $S_r$ with detected fault
{
      Get Notification (fault-type)
      If (fault-type=HF OR fault-type=SF)
      {
          Third party assistance needed
          Initiate Node Recovery ( )
      }
      If (PRR very low OR PDR very low)
          Initiate Node Recovery ( )
    If (Residual energy << Threshold)
          Shut down Sensor Node
}
```

Figure 6: Algorithm 2.

```
Initiate Node Recovery ( )
{
        Send probe packets to all of $S_r$.NBR
        For each node $S_j \in S_r$.NBR
        {
          Maintain Topology ( )
        }
        Start recovery action ($S_r$)
        {
            Reinitialize the sensor node;
            Fetch the last data from $S_r$.DCkpt;
            Get $S_r$.TD;
            Perform LQE;
        }
}
```

Figure 7: Algorithm 3.

Finally the node performs link quality estimation given by LQE, which is again proposed in our work mentioned in [21]. In the node recovery algorithm any node which is in recovery mode sends some probe packets to its neighbor, stating its unavailability for some instance of time. The neighbors in turn, update their own neighbor tables and get prepared for running the topology maintenance schedule.

```
Maintain Topology ( )
{
    Get parent list of $S_r$
    For each parent $S_k$ of $S_r$
    {
      If $\|S_j S_k \le CR\|$
      {
          Update neighbor list
          Update routing table
          Transmit through $S_k$
      }
      Else
      {
          Estimate transmission power $P_{j,k}$
          If  $P_{j,k} < P_{j,k-1}$
              Store current $P_{j,k}$
          Else
              Keep the previous $P_{j,k-1}$
        }
    }
    Select $S_k$ with minimum $P_{j,k}$
    Update neighbor list and routing table
    Transmit through $S_k$
}
```

Figure 8: Algorithm 4.

6 Results and discussions

In this section the results are displayed and corresponding discussion is presented. For simulation purpose and preparing the results MATLAB version 7.11.0.584 (R2010b) and Microsoft Office Excel 2003 was used. The sensor node specifications considered and the simulation environment is mentioned in the Tables 2 and Table 3 respectively. Table 4 next shows the computed energy consumption for each task done by each node. Initially the nodes are deployed randomly and then they are initialized and they start to do their normal task. In an area of 100×100 m^2 thirty sensor nodes were randomly deployed considering a uniform communication range of 25 meters.

6.1 Fault detection and diagnosis

The nodes were deployed randomly and then gradually nodes become faulty in the WSN. The faults are detected and consequentially the recovery is carried out by the nodes.

Just after the nodes are deployed, the scenario is presented in the first quadrant of the Figure 9, which is followed by the edge development of the nodes, depending upon the transmission radius of the sensor nodes and presented in second quadrant of Figure 9. Now for the detection of faults the fault detection algorithm proposed by Mitra and De Sarkar (2014) is used and the nodes demarcated by red colors in the third and fourth quadrant of Figure 9. It was observed that five out thirty nodes were detected to be faulty. Moreover in the Figure 10 especially the faulty nodes with the affected links are represented.

Parameter	Value
Frequency Range	2.4 – 2.48 GHz
Data Rate	250 Kbps
Current Draw	16 mA @ Receive mode 17 mA @ Transmit mode 8 mA @ Active mode 8 µA @ Sleep mode

Table 2: Sensor Node Specifications.

Parameter	Value
No. of Nodes Deployed	30
Area Covered	100×100 m^2
Communication Range	25 meter
Node Density (ρ)	0.003nodes/ m^2

Table 3: Simulation Environment.

Task Performed	Energy Consumed (in mJ)
Data Sensing	0.0018
Data Processing	0.0513
Data Transmission	0.1864152
Data Receiving	0.0627456
Self Evaluation	0.12

Table 4: Energy Consumption for various tasks performed by Sensor Nodes. [18]

6.2 Fault recovery

After the faults are detected then the recovery activities are started. Faulty nodes go for recovery and here they are named as recovering node (RN) and the affected nodes (AN) are their neighbors. Now as in Figure 10 the red nodes are RN and red links are affected links, which will get defunct later on. A list of susceptible parents for each set of ANs is mentioned in Table 5. However the ANs have to select a suitable node to maintain the connectivity even in absence of the corresponding RN.

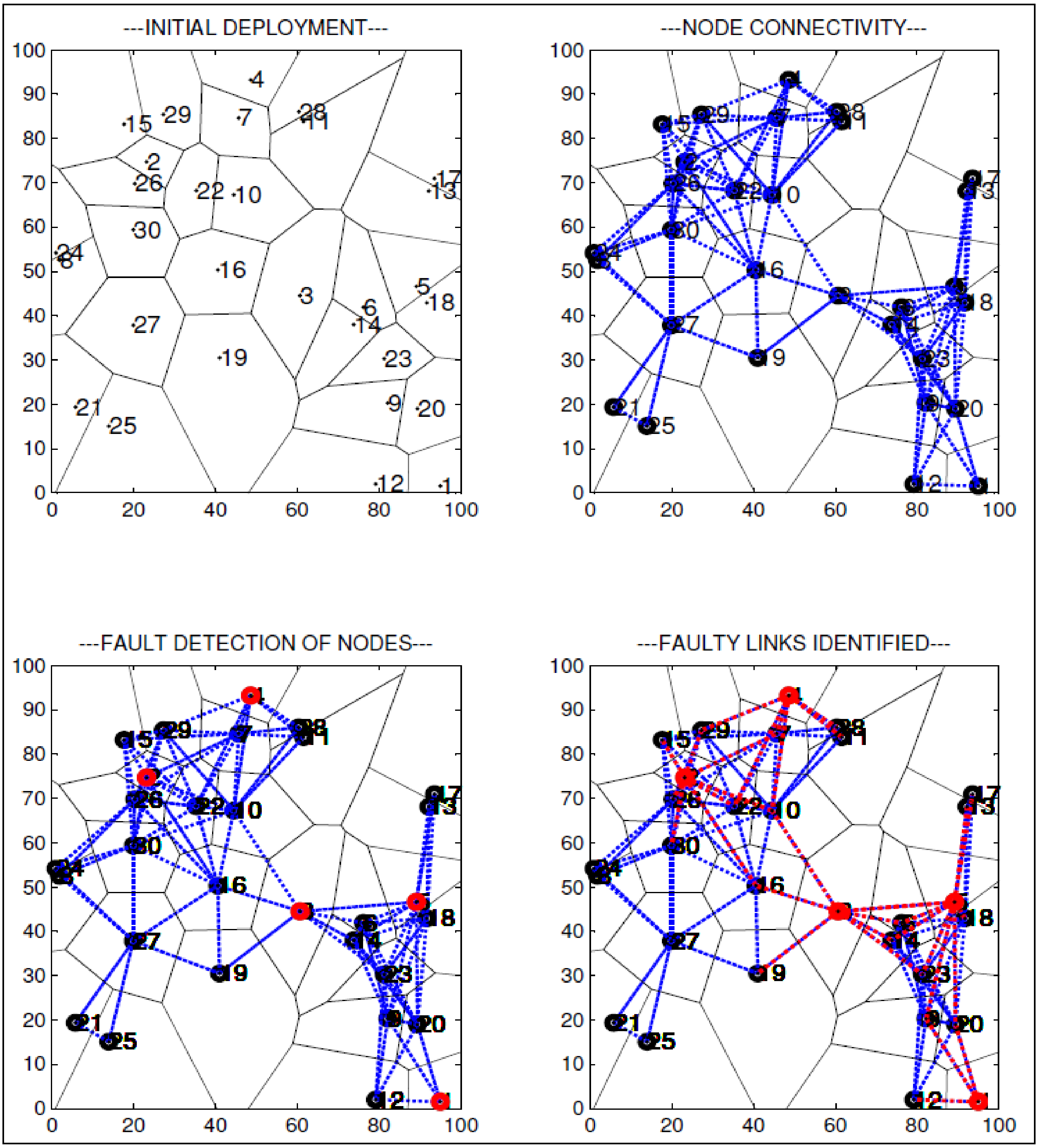

Figure 9: Node Deployments and Fault Scenario.

Case 1 (Recovery for Node 1): Here the node ID 1 is RN after getting faulty goes to recovery mode; now this node sends probe packets to its neighboring nodes, which are actually in its communication range. The IDs of the ANs are 9, 12 and 20. Now these nodes will update their neighbor list and each of them will try to find out a node, which will act as their new parent. There are multiple susceptible parents for nodes 9, 12 and 20 and they select node IDs 18, 23 and 6 respectively, as their new parent. Node ID 9 have 5 susceptible parents and out of which it selects node ID 18 since the transmission power factor is minimum among all other possible parents (as presented in Table 5). Similarly node ID 12 and 20 has 5 and 6 susceptible parents respectively but node IDs 23 and 6 are selected as actual parents because of low transmission factor. So in all the 3 situation the tasks are carried out to minimize the power consumption. The necessary results to support new parent selection, from the each node's susceptible parent list, is elaborately presented in Table 5.

Case 2 (Recovery for Node 4): In the second case node ID 4 is RN and its ANs are 7, 11, 28 and 29. Just similarly like case 1 here all ANs find a suitable parent for transmitting data. In this case there are multiple susceptible parents out of which nodes 7, 11, 28 and 29 (mentioned in Table 5) but each node selects some specific node as their new parent. Moreover they update their neighbor list also.

Node ID 7 and 11 have 6 susceptible parents and out of which node ID 7 selects node ID 10 as its immediate parent and node 11 selects 22 as its new parent. This selection is done on the basis of the minimum power factor for these nodes. Lower power factor means lower power consumption for transmission. Similarly node ID 28 and 29 selects node ID 2 and 30 as their new parent respectively. So in all the 4 situations specific selections are made to keep the power consumption of the node low, in comparison to others. The necessary results to support new parent selection, from the each node's susceptible parent list, is elaborately presented in Table 5.

7 Discussion

Moreover in Table 6 all the ANs are mentioned along with their transmission power and distance from the current parent. After simulation it is inferred that the ANs select those nodes as their new parent, in absence of the RN, through which they can forward data towards BS.

Node 28 has to raise its multiplication factor as high as 2.82, in order to avoid isolation. As in the cases of nodes 11, 28 and 29 the distance with the new parent is greater than their distance with node 4 so they have to raise their transmission power.

Here the activities for two of the nodes with ID 1 and 4 are shown the same activities are carried out for other RNs. All RNs send the probe packets to their neighbors, and the ANs in turn perform the topology maintenance tasks and then the RNs are reinitialized and the values from system variable DCkpt and TD are fetched, since they contain the last data checkpoint and state checkpoint of the node. After that the link quality estimation is done to check the vicinity traffic situation and finally the node comes back to the network.

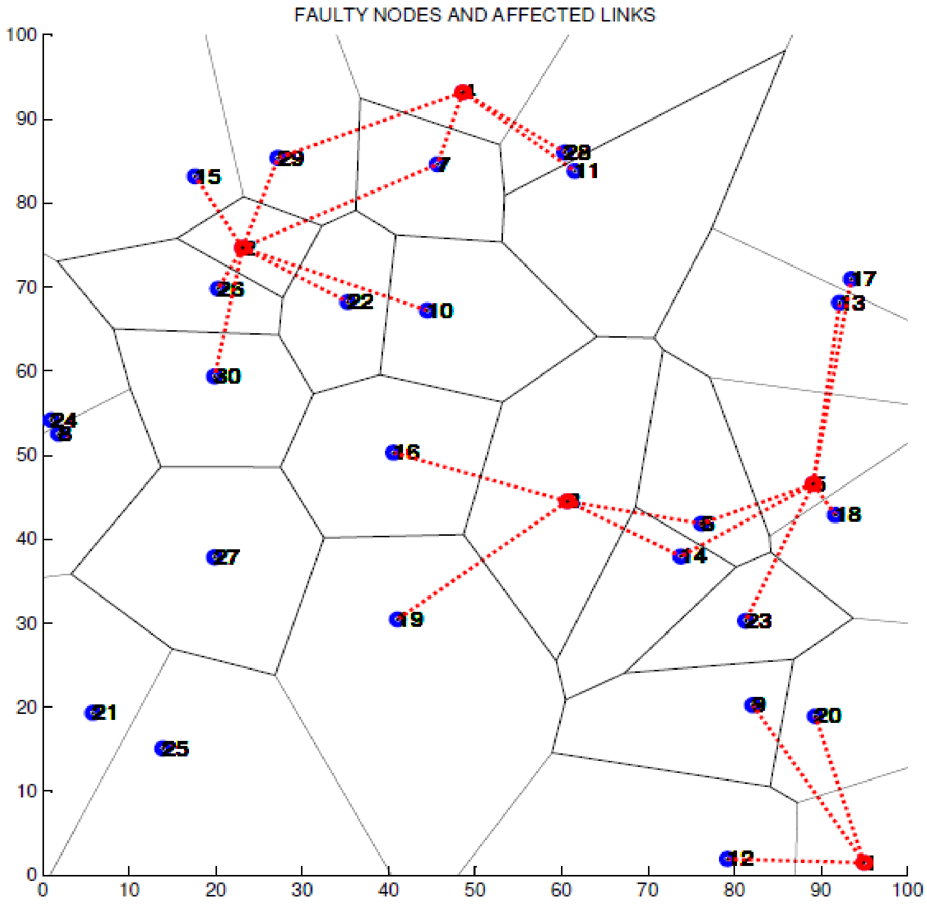

Figure 10: Affected Links due to Node Fault.

AN ID	**Susceptible Parent IDs**	**Power Factor For Each Parent**
9	3, 5, 13, **18**, 19	1.42, 1.20, 2.15, **1.08**, 1.86
12	5, 6, 14, 18, **23**	2.89, 2.53, 2.30, 2.30, **1.79**
20	3, 5, **6**, 13, 17,19	2.10, 1.51, **1.44**, 2.69, 2.85, 2.71
7	2, **10**, 15, 16, 26, 30	2.72, **1.93**, 3.10, 3.83, 3.25, 3.99
11	2, 3, 13, 17, **22**, 29	2.47, 2.46, 2.15, 2.16, **1.91**, 2.15
28	**2**, 3, 15, 16, 26	**2.82**, 3.01, 3.11, 2.96, 3.14
29	11, 16, 28, **30**	1.51, 1.65, 1.46, **1.19**

Table 5: Susceptible Parent List for Affected Nodes.

RN ID	**AN ID**	**New Parent Node ID**	**Power Factor**	**Distance of AN with new Parent (in meters)**
1	9	18	1.08	24.56
	12	23	1.79	28.39
	20	6	1.44	26.41
4	7	10	1.93	17.45
	11	22	1.91	30.54
	28	2	2.82	38.95
	29	30	1.19	27.02

Table 6: New Parent Selections.

8 Conclusion

WSN is used widely nowadays for various field surveillance and distributed fault tolerance in necessary in the same for reliability and dependability of WSN. Novel fault recovery architecture is designed and proposed in this paper; the recovery architecture is destined to be integrated with a fault tolerant framework for wireless sensor network. This paper also presented proposed algorithms for fault recovery and connectivity maintenance in WSN. This algorithm explains details of recovery tasks are carried out. A brief discussion is presented to identify the detection of faults and then different cases for recovery are done.

This proposed recovery technique takes care of recovery actions related to the faults due to hardware or software failure. It also improves link quality or connectivity among the nodes during recovery phases. However, the noise-related measurement or error due to presence of noise is not scope of this paper. This research will enhance the recovery scheme with self-organization and noise-related measurement based recovery in future. As a future work this research will also present a result-interpretation based comparative study of recovery schemes.

9 References

[1] Akbari A., Dana A., Khademzadeh A. & Beikmahdavi N. (2011) “Fault Detection and Recovery in Wireless Sensor Network using Clustering” in International in Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, Issue 1, 130-138

[2] Akbari A., Beikmahdavi N., Khosrozadeh A., Panah O., Yadollahi M. & Jalali S. V. (2010) “A Survey Cluster-Based and Cellular Approach to Fault Detection and Recovery in Wireless Sensor Networks” in World Applied Sciences Journal Vol. 8 Issue 1 76-85

[3] Bathla G. & Jindal S. (2016) “A Review of RIM and LeDiR recovery mechanism for node recovery in Wireless Sensor Actor Network” in International Journal of Engineering Development and Research Vol. 4, Issue 2, 2145-2147

[4] Borawake-Satao R. & Prasad R. S. (2017) “Mobile Sink with Mobile Agents: Effective Mobility Scheme for Wireless Sensor Network” published in International Journal of Rough Sets and Data Analysis Vol. 4 Issue 2 24-35

[5] Brahme C., Gadadare S., Kulkarni R., Surana P. & Marathe M.V. (2014) “Fault Node Recovery Algorithm for a Wireless Sensor Network” in International Journal of Emerging Engineering Research and Technology Vol. 2, Issue 9, 70-76 ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

[6] Chen J., Kher S. & Somani A. (2006) “Distributed Fault Detection of Wireless Sensor Networks” in Proc. of Workshop on Dependability issues in Wireless Ad hoc Networks and Sensor Networks (DIWANS), pp. 65-72 published by ACM, Los Angeles, CA, USA

[7] Graham B., Tachtatzis C., Franco F. D., Bykowski M., Tracey D. C., Timmons N. F. & Morrison J. (2011) “Analysis of the Effect of Human Presence on a Wireless Sensor Network” published in

International Journal of Ambient Computing and Intelligence (IJACI), Vol. 3 Issue 1, 1-13

[8] Haboush A., Mohanty M. N., Pattanayak B. K. & Al-Tarazi M. (2014) "A Framework for Wireless Sensor Network Fault Rectification" published in International Journal of Multimedia and Ubiquitous Engineering Vol. 9 Issue 1 133-142

[9] Kang Z., Yu H. & Xiong Q. (2013) Detection and Recovery of Coverage Holes in Wireless Sensor Networks in Journal Of Networks, Vol. 8, Issue 4, 822-828

[10] Karl H. & Willig A (2005) "Protocols and Architectures for Wireless Sensor Networks" West Sussex, England, John Wiley & Sons Ltd.

[11] Klus H. & Niebuhr D. (2009) "Integrating Sensor Nodes into a Middleware for Ambient Intelligence" published in International Journal of Ambient Computing and Intelligence (IJACI), IGI Global Vol. 1, Issue 4, 1-11

[12] Kumar S. & Nagarajan (2013) N. "Integrated Network Topological Control and Key Management for Securing Wireless Sensor Networks" published in International Journal of Ambient Computing and Intelligence (IJACI), Vol. 5 Issue 4, 12-24

[13] Lakamana V. S. S. K. & Rani S. J. (2015) "Fault Node Prediction Model in Wireless Sensor Networks Using Improved Generic Algorithm" in International Journal of Computer Science and Information Technologies, Vol. 6 Issue 4, 3501-3503

[14] Leskovec J., Sarkar P. & Carlos Guestrin (2005) "Modelling Link Qualities in a Sensor Network" published in Informatica Vol. 29 445–451

[15] Liu X. (2006) "Coverage with Connectivity in Wireless Sensor Networks" in Proc. Of Basenet 2006, in conjunction with BroadNets, San Jose, CA

[16] Ma C., Lin X., Lv H. & Wang H. (2009) "ABSR: An Agent based Self-Recovery Model for Wireless Sensor Network" in Proc. Of Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, pp. 400-404, Chengdu, China

[17] Mishal M.D., Narke V.A., Shinde S.P., Zaware G.B. & Salve S. (2015) "Fault Node Recovery For A Wireless Sensor Network" in Multidisciplinary Journal of Research in Engineering and Technology, Vol. 2, Issue 2, 476-479

[18] Mitra S. & Sarkar A. D. (2014) "Energy Aware Fault Tolerent Framework in Wireless Sensor Network" in Proc. Of AIMoC 2014 pp. 139-145 published by IEEE, Kolkata, India

[19] Mitra S., Das A. & Mazumdar S. (2016) "Comparative Study of Fault Recovery Techniques in Wireless Sensor Network" in Proc. Of WIECON-ECE 2016 pp. 130-133, published by IEEE, AISSMS, Pune, India

[20] Mitra S., Sarkar A. D. & Roy S. (2012) "A Review of Fault Management System in Wireless Sensor Network" in Proc. of International Information Technology Conference, CUBE pp. 144-148 published by ACM, Pune India

[21] Mitra S., Roy S. & Das A. (2015) "Parent Selection Based on Link Quality Estimation in WSN" in Advances in Intelligent Systems and Computing (AISC) Vol. 379, Proc. of IC3T, pp. 629-637, published by Springer, Hyderbad, India

[22] Mukherjee A., Dey N., Kausar N., Ashour A. S., Taiar R. & Hassanien A. E. (2016) "A Disaster Management Specific Mobility Model for Flying Ad-hoc Network" published in International Journal of Rough Sets and Data Analysis Vol. 3 Issue 3 72-103

[23] Nithilan N. & Renold A. P. (2015) "On-Demand Checkpoint And Recovery Scheme For Wireless Sensor Networks" in ICTACT Journal On Microelectronics, Vol. 1, Issue 1, 35-40, ISSN online (2395-1680)

[24] Nguyen K. V., Nguyen P. L., Phan H. & Nguyen T. D. (2016) "A Distributed Algorithm for Monitoring an Expanding Hole in Wireless Sensor Networks" published in Informatica Vol. 40 181–195

[25] Reghunath E. V., Kumar P. & Babu A. (2014) "Enhancing the Life Time of a Wireless Sensor Network by Ranking and Recovering the Fault Nodes" published in International Journal of Engineering Trends and Technology (IJETT) Vol. 15 Issue 8, 410-413

[26] Saleh I., Eltoweissy M., Agbaria A. & El-Sayed H. (2007) "A Fault Tolerance Management Framework for Wireless Sensor Networks" published in Journal of Communications, Vol. 2, Issue 4, 38-48

[27] Wan J., Wu J. & Xu X. (2008) "A Novel Fault Detection and Recovery Mechanism for Zigbee Sensor Networks" in Proc. Of Second International Conference on Future Generation Communication and Networking, pp. 270-274, Hainan Island, China published by IEEE

[28] Yim S. J., & Choi Y.H. (2010) "An Adaptive Fault-Tolerant Event Detection Scheme for Wireless Sensor Networks" published in Sensors Journal, Vol. 10 Issue 3 2332-2347