# 39. A Method of Fault Tolerance and Mitigation in Wireless Sensor Networks

*[1]Dr. Arun Kumar Marandi, [12]Sweta Kumari Barnwal, [3]Dr. S. N. Singh*
*[1]Dept. of computer Science & IT ARKA JAIN University, India Jamshedpur-831014,*
*[3]NIT Jamshedpur NIT Jamshedpur, India Jamshedpur-831014, India*
*[1]dr.arun@arkajainuniversity.ac.*in , *[2]kumar.sweta85@gmail.com*, *[3]snsingh.ece@nitjsr.ac.in*

## ABSTRACT

*For a system fault tolerance is the ability to do the work without any interference. In present era wireless sensor network has wide role or a variety of applications with unlimited future scopes. In Wireless Sensor Networks there are several types of failures just because of various environmental hazards, such as interference, internal factures(battery failure, hardware failure, link errors, DoS attack, processor failure, transceiver failure etc., which affects the whole transmission and therefore the deployment of Wireless Sensor Network is not effective. In this manner fault tolerance is most critical issue in Wireless Sensor Networks. This paper presents, how the faulty nodes of mobile communication can be detect and mitigate. It is very tough to monitor the WSNs continuously by manual operator, so there is a requirement of a system which is in capable of overcoming the failures and can send the data in proper manner. The WSN should be designed in such a way that it should be able to find out the faulty nodes, try to resolve and then transmit the sensed data if any fault occurs and make the fault free network by improving fault tolerance capability. In this paper we have proposed an algorithm for fault detection, which will rectify the fault and make the network fault free, ultimately maximizes the efficiency of the cluster and improves the network performance.*

**Index Terms**— *fault detection, wireless sensor networks (WSNs), fault tolerance, mobile node management.*

## INTRODUCTION

Wireless Sensor Networks (WSNs) have received significant attention in recent years due to their potential application in military sensing, wildlife tracking, traffic surveillance, health care, environment monitoring, building structures monitoring, etc. WSNs can be treated as a special family of wireless ad hoc network [1]. Each sensor node is equipped with a sensing unit, which is used to capture events of interest, and a wireless transceiver, which is used to transform the captured events back to the base station called sink node .Sensor nodes collaborate with each other to perform tasks of data sensing, data communication, and data processing [2]. Nodes in WSNs are prone to failure due to energy depletion, hardware failure, communication link errors, malicious attack, and soon. Unlike the cellular networks and ad hoc networks where energy has no limits in base stations or batteries can be replaced as needed, nodes in sensor networks have very limited energy and their batteries cannot usually be recharged or replaced due to hostile or hazardous environments. So, one important characteristic of sensor networks is the stringent power budget of wireless sensor nodes. Two components of a sensor node, sensing unit and wireless transceiver, usually directly interact with the environment, which is subject to Variety of physical, chemical, and biological factors. Fault tolerant computing can be defined as “The ability to execute specified algorithms correctly regardless of hardware failure and software errors” [2-3]. Five level of fault tolerance are physical layer, hardware layer, system software layer, middleware layer and application layer. Basically the technology of fault tolerant computing encompasses theory and techniques of fault and error detection and correction, modelling, analysis, synthesis, and architecture of fault- tolerant system and their evaluation. The complexity of this subject area can be viewed in a different way. It need computer that would sustain essential operation even under multiple hardware failures and software errors [4]. This implies the requirement of “self-repairing” and highly reconfigurable computer. Moreover we address five categories of applications: node placement, topology control, target and event detection, data gathering and aggregation, and sensor surveillance.

## FAULT TOLERANCE AT DIFFERENT LEVELS

On the basis of study it classify fault tolerance in WSNs into four levels i.e. hardware layer, software layer, network communication layer, and application layer [3-4].

### A. LAYER OF HARDWARE

Faults here can be caused by malfunction of any hardware component of a sensor node, such as memory, battery, and microprocessor. Three main reasons causing hardware failure are: sensor node will not always use the highest quality components, strict energy constraints restrict long and reliable performance of sensor nodes, sensor networks are often deployed in harsh and hazardous environment which affect normal operation of sensor nodes [2].

### B. LAYER OF SOFTWARE

This consist of two components system software (such as operating system) and middleware (such as communication, routing).Software bugs are a common source of error in WSNs. Since it is difficult to provide fault tolerance in economic way in hardware level of sensor node, it is expected at the middleware level [1-3].

### C. LAYER FOR NETWORK COMMUNICATION

Faults in this layer are the faults on wireless communication links which can be caused by radio interference of sensor nodes[11]. The standard way to enhance the performance of wireless communication is to use aggressive error correction schemes and retransmission. These two methods may cause promote delay of operation [2-3].

### D. APPLICATION LAYER

Fault tolerance can be addressed also at the application layer. An approach for fault tolerance cannot be directly applied to other applications. It requires proper addressing of fault tolerance in different applications, on a case by case basis. A WSN is a self-organised network that consists of a large number of low- costs and low –powered sensor devices, called sensor nodes .which can be deployed on the ground, in the air, in vehicles, on bodies, under water, and inside buildings.

## FAULT DETECTION AND RECOVERY

To tackle faults in a WSN, the system should fellow two main steps. The first step is fault detection. It is to detect that a specific functionality is faculty, and to predict it will continue to function properly in the near future [5]. After the system detects a fault, fault recovery is the second step to enable the system to recover from the faults. Basically, there are two types of detection techniques: self –diagnosis and cooperative diagnosis [6].Some fault that can be determined by a sensor node itself can adopt self diagnosis detection.

For example, a sensor node itself can detect fault caused by depletion of battery. The remaining battery of the sensor node can be predicted by measuring current battery voltage. Another example is the detection of failure links. A sensor node may detect that some link to one of its neighbours is faulty if the node does not receive any message from the neighbour within a predetermined interval. However, there are some kinds of fault that require cooperative diagnosis among a set of sensor nodes [5-6]. Recovery is defined as the continuation of system functions after the incidence of an error with data integrity. In a total system environment it is a problem requiring both hardware and software aids. An essential requirement is that error propagation must be minimized and any damaged data must be reconstructed before restarting. The most common used technique for fault recovery is replication or redundancy of components that are prone to be failure [4].

For example, WSNs are usually used to periodically monitor a region and forward sensed data to a base station. When some nodes fail to provide data, the base station still gets sufficient data if redundant sensor nodes are deployed in the region. Multiple path routing is another example. In the case of providing single route, a requested call cannot be set up or be maintained if some nodes/links along the route fail. Keeping a set of candidate routes provides high reliability of the routes for routing. It requires K-connectivity of the network if it is able to tolerate failure of K-1 nodes.

## NODE PLACEMENTS IN TWO-TIERED WIRELESS SENSOR NETWORK

Sensor nodes are prone to failure, one approach to improving reliability and prolonging lifetime of WSNs is the introduction of two-tiered network architecture [7]. The architecture employs some powerful relay nodes whose main function is to gather information from sensor nodes and relay the information to the sink. Relay nodes serve as a backbone of the network. They are more powerful than sensor nodes in terms of energy storage, computing, and communication capabilities. The network is partitioned into a set of clusters and the relay nodes act as cluster heads and they are connected with each other to perform the data forwarding task [5-6]. Each cluster has only one cluster head and each sensor belongs to at least one cluster, such that sensor nodes can switch to backup cluster heads when current cluster head is not available [7]. In each cluster, sensor nodes collect raw data and report to the cluster head. The cluster head analyzes the raw data, extracts useful information, and then generates outgoing packets with much smaller size to the sink via multichip paths. A fault in transmitter can cause the relay nodes to stop transmitting tasks to the sensors as well as relaying the data to the sink. Data sent by the sensors will be lost if the receiver of a relay node fails. So, a communication link fault on a sensor requires the sensor to be reallocated to other cluster heads within communication range [4-5-7]. If faults occur in inter-cluster heads, another multichip path should reconnect the two corresponding cluster heads. Thus to handle general communication faults, there should be at least two node-disjoint paths between each pair of relay nodes in the network [4].

An intuitive objective of relay node placement in two-tiered WSNs is to place the minimum number of relay nodes so that some degree of fault tolerance can be achieved. A lot of work has been done on the minimum placement of relay nodes for fault tolerance in two-tiered WSNs. It does not employ relay nodes and two-tiered architecture but it can be reduced to the same placement problem in two-tiered architecture by setting uniform communication ranges for both sensor nodes and relay nodes [8]. So in this paper we focus on relay node placement problem in two-tiered networks in this section. There are variant dentitions on the problem of minimum placement of relay nodes. The problem can be described as follows. If given a set of sensor nodes that are randomly distributed in a region and their location, some relay nodes are needed to be placed on the region for forwarding data to the sink, such that each sensor node is covered by at least one relay node. The objective is to minimize the number of relay nodes that make the network k-connected. It is assumed that the original sensor network is 2-connected and sensor nodes also participate in forwarding of the data. The objective is to guarantee that at least two relay nodes cover each sensor node and the network of relay nodes is 2-connect. Since sensor nodes usually have limited computing and communication capability, and especially very limited energy resource, it restricts application of the algorithm.

### A. FORMAL DESCRIPTION OF THE PROBLEM IS AS FOLLOWS

Given,

- A set of sensor nodes S in a region
- A uniform communication radius d
- The problem is to place a set of relay nodes R, such that
    - The whole network G is connected
    - G is 2-connected.

The objective of the problem is to minimize |R| where |R| denotes the number of relay nodes in R.The authors proposed a (6 C ")-approximation solution for the case 1 of the minimum relay node placement problem (MRP-1 for short), and then proposed a (24C")-approximation solution for case 2 (MRP-2 for short), where " is an arbitrary positive number and running time is polynomial when " is fixed. The solutions were further extended to the scenario where communication radii of sensor nodes and relay nodes are different. The basic idea of the solutions is to partition the problem into two phases. The 1st phase is to place some relay nodes to cover all sensor nodes [3-5]. The second phase is to add more relay nodes to make the whole network connected=2-connected.The solution is based on two fundamental works. The first is the covering with disks problem. Given a set of points in the plane, the problem is to identify the mini- mum set of disks with prescribed radius to cover all the points. In a polynomial time approximation scheme (PTAS) for this problem was proposed. That is, for any given error, the ratio of the solution found by the scheme to the optimal solution is not larger than. The running time is polynomial when € is fixed. The scheme was called min-disk-cover scheme. The other fundamental work is the Steiner tree problem with minimum number of Steiner points. Given a set of terminals in the Euclidean plane, the problem is to find a Steiner tree such that each edge in the tree has length at most d and the number of Steiner points is minimized. Du et al. proposed a 2.5-approximation algorithm for the STP-MSP. The algorithm was called STP-MSP algorithm. Note that sensor nodes do not participate in data forwarding. STP-MSP algorithm cannot be directly applied to the problem. Based on earlier foundational works, the (6 + €) approximation algorithm for MRP-1 is as follows.

### B. ALGORITHM FOR THE SOLUTION

***Step 1: To search the relay nodes R***

- Input: S, a set of sensor nodes with locations. €, any given error that is larger than 0. d , the communication radius of sensor nodes and relay nodes.
- Output: G, a connected network including sensor nodes and relay nodes.
- Use the min-disk-cover scheme to place a set of relay nodes $R_1$, such that for8s 2 S; 9r 2 $R_1$, and recovers.
- Use $R_1$ as an input of the STP-MSP algorithm to place additional relay nodes $R_2$, such that G is connected.
- Output G and the position of each relay node.

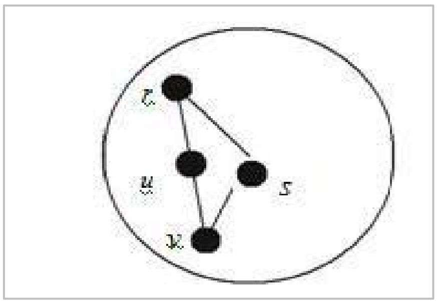

*Figure 39-1 Communication circle of sensor*

***Step 2:***

- Input: S , a set of sensor nodes with locations.€ any given error that is larger than 0. d , the communication radius of sensor nodes and relay nodes.
- Output: G, a 2-connected network including sensor nodes and relay nodes.
- Run algorithm 1 to get a set of relay nodes R, such that S C R is connected.
- Add three backup nodes in the communication circle of each r 2 R. The set of all backup nodes in this step is denoted by $R^0$
- Output G and positions of relay nodes in R C $R^0$

***Step 3:***

- Input: R _ r > 0, "> 0, and set of sensor nodes X D f$x_1$; : : :; $x_n$ g.
- Output: Set of relay nodes Y D f$y_1$; $y_l$ g.
- Apply 5-approximation algorithm in [23] to place set of relay nodes
- Z D f$z_1$ ; : : :; $z_k$ g, such that the resulting network is connected.
- Duplicate each of the relay nodes in Z to obtain Y

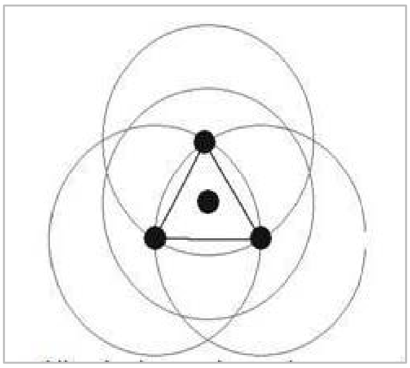

*Figure 39-2 Adding backup nodes to the communication circle of r*

## ANOTHER APPROACH (TOPOLOGY CONTROL)

Although node placement provides a method to achieving fault tolerance in a WSN, the property of fault tolerance may be not valid due to movements and energy depletion of nodes. Therefore, topology control is required to construct and maintain the property of fault tolerance in WSNs [9]. A faulttolerant topology control protocol was proposed it first constructs a Connected Dominating Set (CDS) as a backbone of the network. For each node in the CDS, it adds necessary neighbours of the node to the backbone, such that it meets the required vertex connectivity degree. The power on/off model is adopted to turn on the nodes in the backbone to meet connectivity requirement, and other unnecessary nodes are off. Period rotation is used to keep the fairness among nodes.

There are several selection metrics. One is powerbase selection: the node in CDS selects nodes with more power one by one till the resulting graph is local k vertex connected [10]. Another metric is connection degree. The nodes with higher connection degree are first selected. It is because that the nodes with higher connection degree are supposed to have shorter delay. Simulation results show the improvement of network lifetime with a desired vertex connectivity degree.

The problem is to adjust each sensor's transmission range, such that there exist k-vertex disjoint communication paths from each sensor to the set of super nodes. The objective is to minimize the total power consumed by sensors. Three solutions were proposed. The first k-approximation algorithm consists of two steps. In the first step, a given graph is reduced to a direct graph where super nodes are merged as a root. In the second step, existing optimal solution for the Min-Weight k-Out Connectivity problem is adopted to compute the minimal transmission range of each sensor. The two steps are briefly introduced one by one.

The given graph is denoted by G (v, E, c), where V is the set of nodes, E is the set of edges, and c is the set of weight of the edge (indicating the power consumed in the edge). The reduced graph is constructed as follows. All super nodes in V are merged into one node called the *root*. Edges between sensors remain the same, and an edge between a sensor and a super node is replaced with an edge between the sensor and the root. The weight of the edge remains the same. It should be pointed out that if a sensor is connected to more than one super node, only the edge to the closest super node is kept. After that, every undirected edge between two sensors is replaced with two directed arcs that point to each of them. An undirected edge between a sensor and the root is replaced with one directed arc from the sensor to the root. The process of the step is illustrated in Figure below. The algorithm in the second step is based on the reduced graph from the first step. It applies existing optimal solution for the Min-Weight k-Out Connectivity problem in the reduced graph. The final transmission range of each node is the transmission range used to meet the longest edge in final result. Detail of the algorithm is as follows.

***Algorithm***

- Construct the reduced graph of G.
- Reverse the direction of each arc in the reduced graph and keep the weight of the arc the same.
- Apply the optimal solution for the Min-Weight k -Out Connectivity problem.

- Reverse back the direction of each arc. **for** each sensor **d**
- Adjust transmission range to meet the longest arc in the graph.
- **end for.**

### A. PICTORIAL CONCEPT

Fault-tolerant algorithms for collaborative target detection in sensor networks in which sensor nodes can either fail due to harsh environmental conditions or maliciously. Both algorithms are based on sensor nodes sharing information to reach consensus.

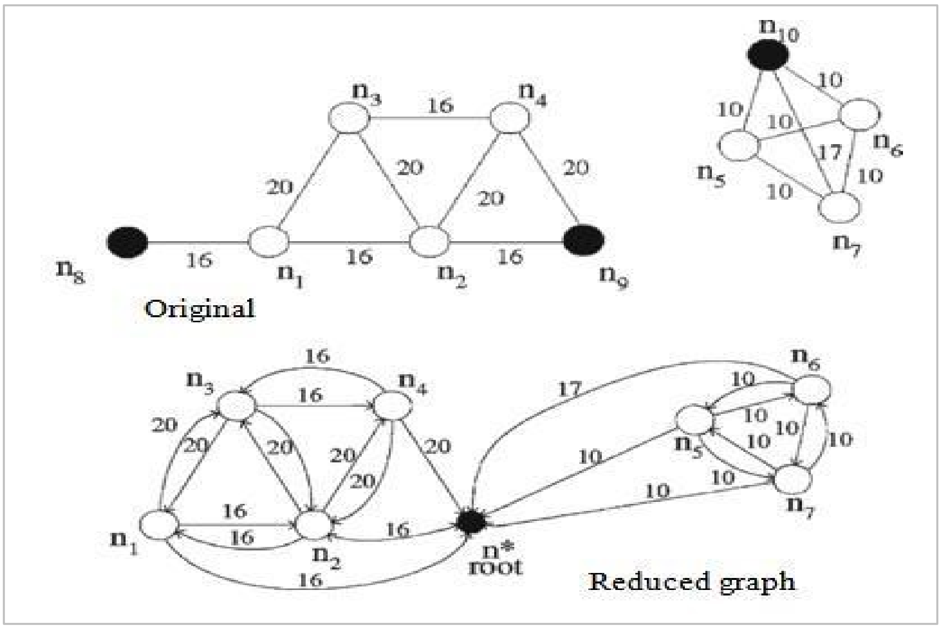

*Figure 39-3 Original graph and reduced graph*

Fault-tolerant algorithms for collaborative target detection in sensor networks in which sensor nodes can either fail due to harsh environmental conditions or maliciously. Both algorithms are based on sensor nodes sharing information to reach consensus.

The first algorithm, called *value fusion*, works as follows. Each node obtains raw energy measurements from every node, computes an average by removing the largest n and smallest n values, and compares this average to a threshold for final decision for a given n. The second algorithm, called *decision fusion*, does not work on raw measurement but rather on local decision of each sensor node. It works in the same way as the value fusion algorithm. The authors mention that there is no need for dropping the data when all nodes are known to be fault-free.

***Algorithm for target detection***

- Each sensor in a given neighbourhood obtains its signal measurements.
- Each sensor computes its median.
- If the median exceeds a threshold the sensor becomes an event sensor.

***Algorithm for target localization***

- Obtain the estimated signal strength from all event sensors in a given neighbourhood.
- Compute the local event sensor that has the maximum signal strength in a given neighbourhood and label them root sensors.
- For each root sensor compute the location of a target based on the geometric centre of a subset of event sensors.

***Algorithm for target identification***

- For each epoch, apply above Target Detection and Target Location algorithm.
- After collecting raw data for T epochs, the base station applies a clustering algorithm to group the estimates into a final target position computation. Each group is one target.
- If the size of a group is less than half the number of epochs (i.e., T =2), then with high probability this group is a false alarm; otherwise, report a target and obtain the estimate of the position of the target using the geometric centre of all raw data within the group.

### B. BY INCREASING CAPACITY OF A CELLULAR NETWORK

As the demand for wireless service increases, the number of channels assigned to a cell ultimately becomes inadequate to support the essential number of users. At this point, cellular design techniques are needed to provide more channels per unit coverage area. Techniques such as cell splitting, sectoring, and coverage zone approaches are used to expand the capacity of cellular systems and increases the fault tolerance ability. There are several methods for enhancing capacity of a Cellular network:

### C. CELL SPLITTING

It is is the process of sub dividing a congested cell into smaller cells, each one having its own base station and analogous reduction in antenna height and transmitted power ultimately increases capacity of a cellular system. By keeping D/R ratio constant entire system is rescaled.

### D. CELL SECTORING

In Wireless network, a single unidirectional antenna at base station can be replaced by multiple directional antennas covering a particular sector for signal radiation and power is transmitted in single desired direction by decreasing number of interfering co-channel cells and co-channel interference, ultimately increases S/I ratio which turns to enhance the system performance. On the amount of sectoring used, the cochannel interference is reduced.

### E. MICROCELL ZONE CONCEPT

On increased number of hand off, load on the switching and control link increases because of sectoring. By microcell zone concept, this problem can be resolve.

### F. IMPROVISED TECHNIQUE USING KRUSHKAL’S ALGORITHM

FGSS: Fault-tolerant Global Spanning Sub graph

It present a centralized greedy algorithm, FGSSk that builds kconnected spanning sub graphs. Kristal’s algorithm is wellknown algorithm to construct the minimum spanning tree (1connected spanning sub graph) of a given graph. FGSS is a generalized version of Kruskal's algorithm for k2The algorithm is given in Algorithm 1.

FGSS$_k$

**INPUT:** $G$ ($V; E$), a$k$-connected simple graph;

**OUTPUT:** $G_k(V_k; E_k)$, a$k$-connected spanning sub-graph of $G$; $V_k$:= $V$ , $E_k$:= ;;

sort all edges in $E$ in an ascending order of weight (as defined

in *Definition 2* ); **for** each edge($u_0; v_0$)in the order **do** **if** $u_0$is not$k$-connected to$v_0$in$G_k$ **then** $E_k$:= $E_k$ [ $f(u_0; v_0)g$;

**else if** all nodes are in the same$k$-connectedcomponent **then** exit;

**end if ;**

**end for ;**

By using network flow techniques, a query on whether two vertices are $k$-connected can be answered in $O(n+ m)$ time for any fixed $k$, where $n$ is the number of vertices and $m$ is the number of edges in the graph. For $k$ 3, there also exists $O$ (1) time algorithms. Therefore, the time complexity of FGSS$_k$ is $O(m(n + m))$, and can be improved to $O(m)$for $k$ 3.

## CONCLUSIONS

The goal of the paper is to investigate current research work on fault tolerance in WSNs. it studied how fault-tolerant techniques were addressed in node placement, topology control, target and event detection, data gathering and aggregation, and sensor surveillance. The paper focused on the application layer and introduced representative works in each application. Actually, there are other applications where fault tolerance attracts attention, such as clustering, time synchronization, gateway assignment, etc.Although extensive works have been done on fault tolerance in each layer of the WSN system, cross-layer solutions are expected in future. Use of the resource could be more efficient if resource can be properly integrated and scheduled in different layers. Therefore, cross-layer solutions are expected to have better performance than current solutions.A new trend of WSNs is to cooperate or integrate with other wireless de-vice/systems, such as actuator networks and RFID system. For example, there are an increasing number of applications that require the network system to interact with the physical system or environment via actuators. That is, it requires the use of sensor networks along with actuators to build wireless sensor and actuator net-works (WSANs). Although fault tolerance techniques for WSNs could be reused in WSANs, there are new challenges that require new solutions. The sensors Network, report their data to the actuator may either switch to another actuator or directly pass the data to the sink.

## REFERENCES

[1] V. Dhiman and T.P. Sharma," Optimal Node Deployment for Fault Tolerant Wireless Sensor Networks: A Survey", International Journal of Advanced Research in Computer Engineering & Technology.,Vol. 4 Issue 5, May 2015.

[2] F. Hussen, K. Elleithy and A. Razaque.," Implementation of Fault Tolerance Algorithm to Restore Affected Nodes in Scheduling Clusters", International Journal of Computer Networks & Communications., Vol.4, No.1, January 2012.

[3] D. K. Baruah and L.P. Saikia.," A Review on Fault Tolerance Techniques and Algorithms in Cloud Computing Environment", International Journal of Advanced Research in Computer Science and Software Engineering. Vol. 5, Issue 5, May 2015.

[4] P.D. Kale and R.M. Tugnayat.," A Survey of Fault Detection and Management Techniques in Wireless Sensor Networks", International Journal of Engineering and Technical Research, Vol.2, Issue-6, June 2014

[5] Bhuiyan, M.Z.A.; Wang, G.; Cao, J.; Wu, J. Deploying wireless sensor networks with fault-tolerance for structural health monitoring. IEEE Trans. Comput. 2015, 64, 382–395.

[6] Zhu, Y.H.; Qiu, S.; Chi, K.; Fang, Y. Latency aware IPv6 packet delivery scheme over IEEE 802.15.4 based battery-free wireless sensor networks. IEEE Trans. Mob. Comput. 2016.

[7] A.K. Marandi and D.A. Khan.,"Statistical analysis of defect removal effectiveness to improve the software quality and reducing the estimated cost", IEEE Xplore. 04 May 2015.

[8] F. Araujo,´ and L. Rodrigues. "On the Monitoring Period for Fault-Tolerant Sensor Networks," LADC 2005, LNCS 3747, Sao˜ Salvador da Bahia, Brazil, October 2005.

[9] J.L. Brediny, E.D. Demainez, M.T. Hajiaghayiz, and D. Rus, "Deploying Sensor Networks with Guaranteed Capacity and Fault Tolerance," MobiHoc 2005, urbana-champaign, IL, 2005.

[10] M. Cardei, S.Yang, and J.Wu, "Algorithms for Fault-Tolerant Topology in Heterogeneous Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 19, no. 4, pp. 545–558, 2008.

[11] Samira Choukhi, et. al.; "A Survey on Fault Tolerance in Small and Large Scale Wireless Sensor Networks" , Computer Communication 2015.