2014 Applications and Innovations in Mobile Computing (AIMoC)

Energy Aware Fault Tolerant Framework in Wireless Sensor Network

Siba Mitra
Dept. of Computer Science & Engineering,
Birla Institute of Technology, Mesra
Kolkata Campus, India
sibamitra@bitmesra.ac.in

Ajanta De Sarkar
Dept. of Computer Science & Engineering,
Birla Institute of Technology, Mesra
Kolkata Campus, India
adsarkar@bitmesra.ac.in

***Abstract*— Wireless Sensor Network, composed of tiny sensor devices and wireless network, is mainly responsible for any kind of ambience surveillance. Due to the peripheral atmosphere in which it is deployed, tiny sensors or the network might be too much fault prone. It is beneficial if and only if sensed values are fault free and it can traverse through fault free path. Thus it is necessary to monitor the network and the sensor nodes in regular interval to generate required result for application specific decision making. Network lifetime play the crucial role in order to monitor network health. It is critical as a certain percentage of the total number of sensor nodes along with its connectivity should remain alive for smooth operation of the network. The objective of this paper is to propose an energy aware fault tolerant framework in wireless sensor network. Fault detection algorithm and maximization of network lifetime in wireless sensor network is also proposed together with the calculated energy consumption of the sensor nodes for performing various tasks, including self fault checking, in the network. Simulation result for the proposed algorithm is also presented in this paper.**

Keywords*— *Wireless sensor network, network lifetime, fault detection, energy consumption

I. INTRODUCTION

The sensor nodes in a Wireless Sensor Network (WSN) are used to monitor any ambient environment, which is unleashed and unpredictable. The sensor nodes are vulnerable to the hazardous environment, which may result to various faults like sensing unit failure, radio transceiver failure, and processor failure in them; moreover sensor node death occurs due to energy crunch consequentially. So it can be inferred that faults directly affect the network lifetime of the WSN. Lifetime of WSN means the time period till when the sensors collectively satisfies the application requirements even if some percentage of sensor nodes are already dead. With the increase in percentage of dead sensors fault prevention and avoidance strategy must be applied for reliable and better performance of WSN. A reliable sensor network contributes to dependable decision making.

Fault tolerance of a network enables it to maintain a standard quality of service (QoS). Designing an energy efficient, fault tolerant WSN is a challenging task. The main objective of this paper is to propose an energy aware fault tolerant framework in wireless sensor network. Fault detection algorithm and maximization of network lifetime in wireless sensor network is also proposed. A fault detection algorithm is presented to make the network aware about the in-network faults, both sensing and communication.

The remainder of the paper is organized as follows; in section II related research works are presented. Section III presents the design of a fault tolerant framework for WSN, and then section IV proposes the fault detection algorithm. The simulation results and discussion is presented in section V. Consequentially section VI concludes the paper.

II. RELATED WORK

Throughout the years various researchers have considered implementation of fault tolerance in WSN as a significant domain of work. Both transient and permanent faults are trivial; moreover faults can be either repairable or irreparable. So there must be some integral fault management system in the network, which can perform significant tasks to make the WSN fault tolerant.

Lee et al. in [13] proposed a distributed fault detection algorithm for homogeneous networks, which isolates faulty sensor nodes by comparing median of all neighbors' data. Both permanent and transient failures are taken care of. In their network model sensor nodes with faulty sensing unit may take part as a relay node for others.

Again Hayoung et al. have proposed a key management scheme for Medical Sensor Network in [18], which can detect faulty sensor nodes. Data security is improved as they have applied Cryptographically Generated Address in the network. Each sensor here, encrypts its address and sensed data and then exchange with neighbor, and relies on majority voting. Detection accuracy in this scheme is inversely proportional to the number of faulty sensor nodes.

In [4] Chen et al. have developed a Distributed Fault Detection (DFD) algorithm, where sensor node localization and fault detection is done. Here they have used built-in-self-test (BIST) and built-in-self-repair (BISR) approach. Here fault probability of the neighbors of likely good and likely faulty sensor node decides the actual health of a sensor node. Finally good sensor nodes are used for decision making. An improved DFD scheme with better diagnosis feature is proposed in [8]. Here faults are detected well even in less sensor node density and high node failure rate.

Zhipeng et al. in [25] proposed a centralized and passive fault detection algorithm, where primary concentration is to reduce computation and communication cost. The neighbors' data of a sensor node is considered for its trust degree calculation. Here the fault detection rate is inversely proportional to the permanent sensor node fault rate and suitable for an always on sensor node.

In the Adaptive fault tolerant event detection scheme for WSN in [24] by Yim et al. it is noticed that they have modeled

the WSN as directed weighted graph and each sensor node of it has a certain number of neighbors. The path between a pair of sensor nodes is weighted and this only decides the neighbor's confidence level from its point of view. All the neighbors with very low confidence level are isolated as faulty sensor nodes. False alarm rate (FAR) is directly proportional to fault probability.

Mahapatroa et al. in [14] have designed a distributed communication fault detection algorithm for WSN, which can handle intermittent, transient and permanent faults. They do a neighbor value analysis to check the fitness of the sensor node. Here hard sensor nodes have communication failure and soft sensor nodes operate with a changed behavior.

Another network lifetime analysis of an always-on WSN was done by Santosh et al. in [12] where they have shown that network lifetime is dependent on some of the factors like, continuous monitoring, event notification requirement, frequency of events etc. They have LPL [20] and hierarchical sensing feature in their network. They have evaluated and estimated the lifetime of an active sensor node and sleeping sensor node.

In existing approaches, mentioned in [24] and [25] communication level faults are not dealt with. High fault probability may generate more false alarms in the scheme proposed in [4]; moreover it also incurs good amount of information exchange overhead. A review of existing fault tolerant algorithm is presented in [16]. This research work proposes a fault detection algorithm, which proactively evaluates self-health and detects any fault in it. This research actually detects sensing fault and communication fault as well. This proposed algorithm and framework is energy aware as it always measures energy consumption in every task.

Next section proposes a fault tolerant framework with its major components in detail.

III. PROPOSED FAULT TOLERANT FRAMEWORK

This proposed fault tolerant framework aims to provide fault tolerant and reliable WSN. Proposed framework has three significant modules namely fault detection, diagnosis and recovery. Detection of fault means to discover that a fault has occurred, fault diagnosis is to know the fault type and the third and most significant job is to recover the loss due to occurred fault. This fault tolerant framework is shown in Figure 1. The framework is divided in three fundamental layers: *fault detection layer, fault diagnosis layer* and *fault recovery layer*. *Fault detection layer* is responsible for detecting any fault occurrence and predicting faults. Components of this layer are namely: *Sensor Monitor Listener*, *Sensor History Manager, Monitor History DB, Sensor Fault Detector* and *Sensor Fault Predictor*. *Sensor Monitor Listener* monitors health of WSN. It actually checks each sensor node (sensing device or sink) or network regularly. It can communicate to cluster head at regular interval and relay information to the *Sensor History Manager*, which runs relevant query to match the information with the existing data in the *Monitor History DB*. *Monitor History DB* is repository of historical events, which has already been detected earlier and dealt with. While matching the query *Sensor History Manager* can also send the information to *Sensor Fault Detector* and *Sensor Fault Predictor* simultaneously, so that the framework can predict the fault as well as detect the occurred one. In case, the sensed data does not match with any of the existing one then the sensor fault detector records the event in the database.

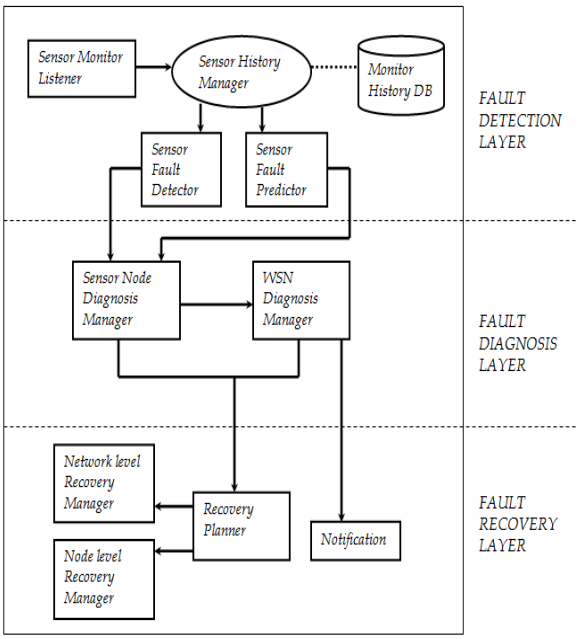

Figure 1: Proposed Fault Tolerant Framework

Sub sequentially after fault detection or prediction, detector and/or predictor actually communicate to *Sensor Node Diagnosis Manager* of *fault diagnosis layer* for diagnosis of fault. Fault can be either of sensor node level or network level. In this layer the *Sensor Node Diagnosis Manager* finds out the exact occurrence of the fault and tracks the actual origin of the fault and its attributes. It also finds out exactly which hardware in the sensor node is affected. The obtained information is further conveyed to the *WSN Diagnosis Manager*, who manages to the total diagnosis work for the whole network. *Sensor Node Diagnosis Manager* and *WSN Diagnosis Manager* both communicate to *Recovery Planner* of the next layer (*fault recovery layer*). The *fault recovery layer* is composed of four components: *Network level Recovery Manager, Node level Recovery Manager, Recovery Planner* and *Notification*. *WSN Diagnosis Manager* directly generates an alarm through *Notification* of the recovery layer. *Recovery Planner* contacts *Node Level Recovery Manager* and *Network Level Recovery Manager* to recover the fault. *Network Level Recovery Manager* does the necessary reconfiguration of the network, if required, and also removes the effects of the fault. The *Node level Recovery Manager* first does the restructuring job for the sensor node, corrects the hardware or software level errors so that the sensor node functionality is preserved and then recovery scheme is started. The *fault recovery layer* performs two important tasks , first is reconfiguration, where the system is restructured in such a format that the fault have zero effect on the correct output data, and the second is recovery, which attempts to eliminate the effects of occurred faults.

IV. DESIGN OF FAULT DETECTION ALGORITHM

With extensive use of WSN, power may get exhausted, sensing fault or transmission fault might occur or received signal strength (RSS) and link utilization (LU) may be degraded. This section mainly proposes the fault detection algorithm. Notations and symbols used in the proposed algorithm are defined in Table I. **Figure 2** represents the algorithm applicable for fault detection layer of the proposed framework.

Table I: SYMBOLS AND NOTATIONS

Symbol	**Description**
S_n	Set of n number of sensor nodes
S_i.NBR	Neighbor of i-th sensor node
num_S_i.NBR	Number of neighbors of in S_i.NBR
S_i.current_val	Value read by S_i currently
S_i.previous_val	Value read by S_i previous to the current one
S_i.diff_own	Difference of S_i.current_val and S_i.previous_val
S_i.NBR_current_val	Values read by S_i.NBR currently
S_i.mean_nbr_val	Mean of S_i.NBR_current_val
S_i.diff_nbr	Difference of S_i.current_val and S_i.mean_nbr_val
m, k, j	Finite numbers used for iteration
S_i.TEND_OWN	Tendency of S_i with respect to its own last values
S_i.TEND_NBR	Tendency of S_i as compared with its neighbors
S_i.MEAN_RSS	Mean of RSS of the neighbors of S_i
S_i.MEAN_LU	Mean of LU of S_i, during data transmission
S_i.TD	Trust Degree of each S_i
sens_θ	Threshold for sensed value
RSS_θ	Threshold for RSS
LU_θ	Threshold for LU
t_f	Time elapsed for one frame/packet transit
t_p	Data propagation time
W	No. of frames delivered from S_i to the BS in a round
L	Number of bits per frame (frame length)
TR_i	Total number of rounds S_i sends data
tot_data_{ij}	Total data transmitted from i to j in one round.
$E_{SENSING}$	Energy consumption for sensing data
$E_{RECEIVE}$	Energy consumption for receiving data
$E_{DATA\_PROCESSING}$	Energy consumption for processing received data
$E_{TRANSMIT}$	Energy consumption for data transmission
$E_{SELF\_CHECKING}$	Energy consumption for self-evaluation for faults
E_{SENS_FAULT}	Energy consumption for sensing fault evaluation
E_{COMM_FAULT}	Energy consumption for communication fault evaluation
E_{TOTAL}	Sum of all the above energy consumption
E_{INIT}	Initial energy of a sensor node

```
For each Si
    a)build (Si.NBR);
    b)check_BS();// Returns true if BS is in transmission range
// End for
//·····Building NBR and checking BS·····
For each Si
    a)Initially Si.TEND_OWN=0 and Si.TEND_NBR=0;
    b)For j=1 to k // k is a finite number
        i.  Si.current_val=Si.sense (ambient_signal);
        ii. Si.diff_own = Si.current_val~Si.previous_val;
        iii. Si.previous_val= Si.current_val;
        iv. If (Si.diff_own > sens_θ) // threshold
              {Si.TEND_OWN= Si.TEND_OWN+1;
               // otherwise Si.TEND_OWN remains unchanged.
              }
        v.  Si.mean_nbr_val=compute(mean of Si.NBR_current_val);
        vi. Si.diff_nbr = Si.current_val~ Si.mean_nbr_val;
        vii. If (Si.diff_nbr > sens_θ)
              {Si.TEND_NBR= Si.TEND_NBR+1;
             // otherwise Si.TEND_NBR remains unchanged.
              }
  //End for
    c) If (Si.TEND_OWN > k/2)
        {If (Si.TEND_OWN > Si.TEND_NBR)
              Si.TD=1; //Likely Healthy (LH)
       Else
         Si.TD=0; // Likely Faulty (LF)
        }
    d)  Else
        {If (Si.TEND_OWN >= Si.TEND_NBR)
              Si.TD=1; // Likely Healthy (LH)
        Else
              Si.TD=0; // Likely Faulty (LF)
        }
    e)  If (Si.TD=0)
          Call RecoveryPlanner ();
//End for
//·····Detection of Sensing Fault·····
For each Si
    a)Initially Si.MEAN_RSS=0 and Si.MEAN_LU=0;
    b)For m=1 to num_Si.NBR
        i.  Si.MEAN_RSS= Si.MEAN_RSS + eval_RSS (Sm);
//eval_RSS () computes the RSS of Sm, m-th neighbor of Si
        ii. Si.MEAN_LU= Si.MEAN_LU + eval_LU (Sm);
     //End for
    c)  Si.MEAN_RSS= Si.MEAN_RSS/ num_ Si.NBR;
    d)  Si.MEAN_LU= Si.MEAN_LU/ num_Si.NBR;
    e)   If (Si.MEAN_RSS < RSS_θ && Si.MEAN_LU < LU_θ)
           Call RecoveryPlanner ();
//End for
//·····Detection of Communication Fault·····
```

Figure 2: Algorithm

A. *Problem Formulation*

In this research, a WSN model is considered to have a set of similar type of sensor nodes S = {S_1, S_2, …., S_n} deployed randomly for monitoring purpose. The network can be represented as a graph G (S, E), where S is the set of sensor nodes or vertices and E is the set of links or edges, which can be represented as an ordered pair (S_i, S_j). Link exists among sensor nodes if and only if S_i and S_j remain within each others'

transmission radius. For each S_i a neighborhood table S_i.NBR contains the set of its neighbors. If base station (BS) is a member of S_i.NBR, then S_i is a susceptible cluster head (CH). So it is assumed that there are mainly two types of sensor nodes CH and leaf node. The role of CH is to acquire, aggregate and transmit data from leaf node along with its own data to the BS. The role of leaf node is to sense any ambient signal and transmit it to CH. In this model a set of cluster heads CH = {CH_1, CH_2, .., CH_p} is considered, where CH is a subset of S, the link (S_i, BS) exists and p< n/2.

For simplicity, periodic data traffic generation is considered in the network. In this model, if a sensor node has transient fault in sensing for any sensor, the proposed algorithm can detect that by redundant checking and calculating its trust degree. Permanent sensing unit failure can be treated by putting on secondary sensing unit in a multi-sensor node. Moreover RSS of receivers of each S_i and LU of S_i are decision makers for any communication fault.

The trust degree of any sensor, S_i.TD, decides whether it will transmit data or not. The value of S_i.TD can either be 0 for likely faulty (LF) node or1 for likely healthy (LH) node respectively. The evaluation is done on the basis of two comparisons. Firstly, S_i performs a comparison of S_i.diff_own with the threshold sens_θ; secondly, S_i compares Si.diff_nbr with the threshold. Both the comparisons are repeated for k times. S_i.TEND_OWN holds the total number of mismatches with its own read value and S_i.TEND_NBR holds the total number of mismatches with the neighbors' mean value. If the sensor node deviates from its own read value for more than k/2 time then S_i.TEND_OWN is compared with S_i.TEND_NBR; if latter is lesser than former, which implies reading of S_i is changing from itself, keeping a match with neighbors; so it finds it-self trustworthy hence S_i is LH. Otherwise the S_i is LF. Moreover if S_i deviates from its own reading for k/2 number of times or less, then it compares S_i.TEND_OWN and S_i.TEND_NBR; again latter value lesser than the former implies S_i has lesser number of mismatches with its neighbors than that of with it-self, hence S_i is considered to be LH. Otherwise the S_i is LF.

At a given point of time each S_i communicates with its CH and hence BS if and only if it is healthy. When a sensor node S_i performs self-checking for any communication fault, it mainly checks the RSS of its own neighbors S_i.NBR and also checks its LU for the last data transmission. S_i.mean_RSS and S_i.mean_LU holds the mean of the RSS and LU value respectively. If the obtained result does not conform by their thresholds (RSS_θ and LU_θ) then the sensor nodes may have communication fault.

B. Assumptions

Some of the design consideration to avoid fault in the proposed framework are discussed below:

- Transmission quality is proportional to RSS and Eq. (1) gives the value of standard RSS according to [23], where d is distance traversed by the signal, λ is wavelength of signal. The unit of RSS is dBm.
- Link utilization is the ratio of the time required by a sending sensor node to transmit data to the total time for corresponding data transit. Eq. (2) gives the LU according to [6], where W, t_f and t_p are mentioned in Table I.
- Energy optimization can be implemented by using duty cycling [1] in sensor nodes. The sensors go to sleep mode while not working and becomes active when required.

$$RSS = -\left(20 \log \frac{d}{\lambda}\right) + \left(20 \log (4\Pi)\right) \quad (1)$$

$$LU = \frac{W \times t_f}{t_f + (2 \times t_p)} \quad (2)$$

C. Network Lifetime Maximization Problem

A fault tolerant WSN is expected to operate and provide good quality data even in presence of faults. According to [10] for each sensor, till its sensed data reaches the BS, is referred to as one round. The lifetime of a network is directly proportional to the number of rounds of data transmission to BS. However, this network lifetime (NL) may vary with successive disconnection of leaf nodes from BS as a result of their CH death. So, maximizing NL means minimizing energy consumption and hence this is an optimization problem. If a sensor node and the corresponding network are well aware of its energy expenditure then it can act proactively against uncertain sensor node death and save energy for future use. Energy consumption for each sensor node is directly proportional to the frequency of data transit; data receive and self-fault detection activities. Hence, energy consumption is directly proportional to the time taken by S_i to complete its task and is referred to as E_{TOTAL} as represented in Eq. (3).

In this model, sensor node specification and simulation environment are represented in Table II and Table III. Total energy consumed by a sensor node includes energy consumed for all the tasks including sleep-wake up switching context. The total energy consumed for a sensor node is always less than its initial energy and presented in Eq. (4). As in Eq. (5) total data transit in the network is always greater than estimated data transit. This happens because retransmission is done when there is loss or damage of data. Eq. (6) says that total energy spend for total data transit in WSN is greater than total energy spent for data processing Eq. (7) depicts the fifth constraint stating that total energy consumption for self checking is greater than equal to individual energy consumption for sensing and communication fault detection. Eq. (8) is self explanatory.

The network lifetime (NL) maximization problem is given below:

$$\max \quad f(NL)$$

Subject to:

$$E_{TOTAL} = E_{SENSING} + E_{DATA\_PROCESSING} + E_{RECEIVE} + E_{TRANSMIT} + E_{SELF\_CHECKING} \quad (3)$$

$$E_{TOTAL} \leq E_{INIT} \quad (4)$$

$$\sum_{n=1}^{TR_i} tot\_data_{i,j} + \sum_{n=1}^{TR_i} tot\_data_{j,i} \geq \sum W \times L \times t_f \quad (5)$$

$$\sum E_{DATA\_PROCESSING} \leq E\left(\sum_{n=1}^{TR_i} tot\_data_{i,j} + \sum_{n=1}^{TR_i} tot\_data_{j,i}\right) \quad (6)$$

$$E_{SELF\_CHECKING} \geq E_{SENSE\_FAULT} + E_{COMM\_FAULT} \quad (7)$$

$$E_{SENSING} \langle E_{RECEIVE} \langle E_{TRANSMIT} \quad (8)$$

Table II: SENSOR NODE SPECIFICATIONS [22]

Data Rate	250 kbps
Frequency Range	2.4 to 2.48 GHz
Current Draw	16 A @ Receive mode 17 A @ Transmit mode 8 mA @ Active mode 8 µA @ Sleep mode (total)

Table III: SIMULATION ENVIRONMENT

Number of Sensor Nodes	30-50
Communication Range	50 meters
Area Covered	200 × 200 m²
Initial energy for a sensor node	21600 Joules
Traffic Generation	After one second
Battery Capacity	2000 mAHr
Packet Size	240 bits
Request Message Size	72 bits
Acknowledgement Size	60 bits

Next section presents the simulation result of this proposed model and algorithm.

V. RESULTS AND DISCUSSIONS

Proposed fault detection algorithm has been evaluated with the help of MATLAB version 7.11.0.584 (R2010b). This proposed algorithm is evaluated in two parts: detection of faults through simulation and checking of network lifetime through energy consumption of each sensor node. This evaluation process used the data sheets in [15] and the specifications mentioned in [22]. Thirty sensor nodes having same transmission range of 50 meters were randomly deployed in a rectangular area of 200 × 200 m^2, and corresponding BS or CH are identified and the remaining are assumed to be leaf sensor nodes. The sensor nodes were numbered 1 to 30 on a random basis.

Figure 3 shows sensor node 16 (marked square) is considered as BS and circled sensor nodes are found to communicate directly with sensor node 16 and can act as cluster heads for the remaining sensor nodes. So here the contents of CH = {3, 10, 19, 22, 27, 30}.

A. *Evaluation of Fault Detection*

Sensing faults are detected by evaluating their trust degree. Trust degree is dependent on neighbor comparison and self-comparison. The sensor node's sensed value was compared with its last reading and the mean of all neighbors' value for k (here k=10) iterations. And finally the trust degree of the sensor node, S_i.TD is checked for fault isolation. If a sensor node's status is LH for sufficient number of times then it is considered as trustworthy, but if it is LF for a good number of times then it is considered as fault prone.

Figure 4 represents faulty sensor nodes, which are marked as red triangles.

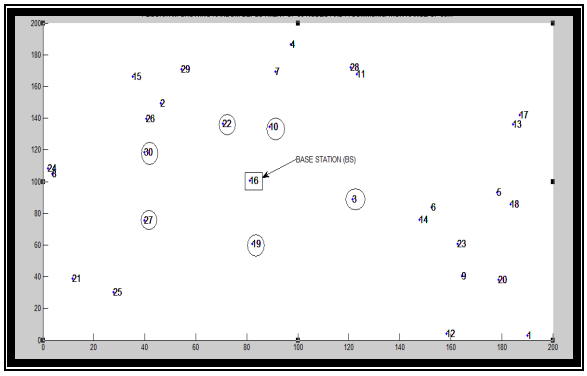

Figure 3: Positions of BS and CH

Communication faults are detected evaluating RSS (in dBm) and LU values. In order to calculate RSS and LU, transmission ranges considered were 40-100 meters and it is found that higher the communication range bigger is the loss of signal power and link utilization is lesser. Finally, any transmitting sensor nodes' RSS value and LU values were compared with threshold RSS_θ and LU_θ respectively and for any non-conformability (i.e. RSS_θ <-125 and LU_θ < 0.95) the sensor node cannot communicate properly. These faulty nodes are represented as triangle (red in color) in **Figure 5**.

B. *Computation of Energy Consumption*

Computation of energy consumption for each sensor node is carried out with basic considerations used in [15], [22]. Duration of time is calculated considering the message size and data rate as mentioned in Table II and Table III respectively. Energy required by a sensor node for each of its task is shown in Table IV.

Time related to data transmission activity includes time for clear channel availability, sending RTS bits, receiving CTS from receiver, transmit the user message and inter-converting itself from transmit mode to receive mode as and when required. Hence energy consumed by a sensor node for transmitting a data packet is approximately 0.1864152 mJ.

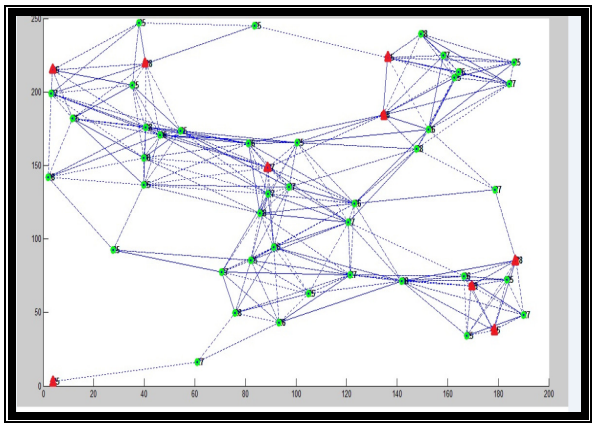

Figure 4: Sensor Nodes with Sensing Faults

Similarly, time related to receiving data also includes time for sending CTS to sender, receiving user message, sending acknowledgement and also inter-converting itself from receive mode to transmit mode as and when required. Hence energy consumed by a sensor node for receiving a data packet is approximately 0.0627456 mJ. Sensor node's data processing time depends upon the number of bits processed and logging them properly for reference. Total energy needed for self-checking of a sensor node depends upon time spent by it to detect any sensing or communication anomaly.

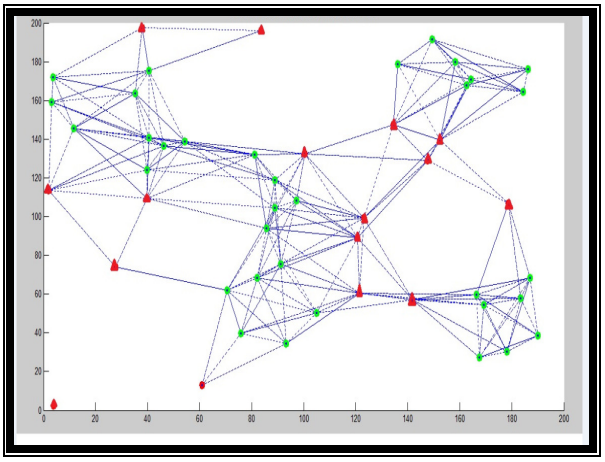

Figure 5: Nodes with Communication Fault

Figure 6 shows the number of packets transmitted by each sensor node by consuming a fixed amount of energy (assumed 5 Joules). This can give an approximate idea of total data transit in the network, which will be helpful in handling traffic burst. Number of packets transmitted by any CH is much less than a leaf node within a stipulated energy as in this case. This is because the leaf nodes are only sensing and transmitting signal to the CH, whereas CH performs the data receiving job from all its neighbors, also aggregates them meaningfully and finally transmits it to the BS.

Table IV: ENERGY CONSUMPTION FOR VARIOUS TASKS PERFORMED BY EACH SENSOR NODE

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

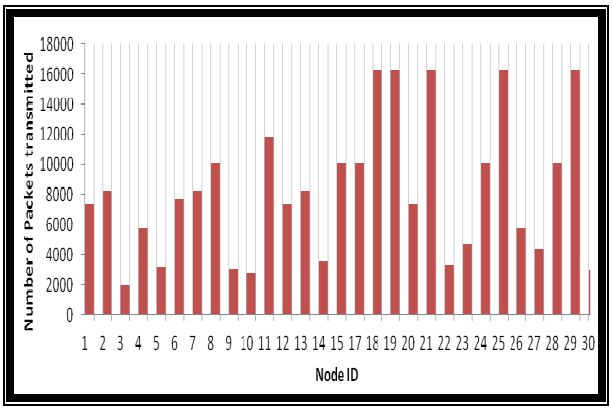

Figure 6: Packets transmission with fixed amount of energy

The plot in **Figure 7** represents the amount of energy (in Joules) consumed by each sensor node in the network. From the plot it is clear that energy consumption of CH to accomplish its total task is much more than a normal leaf node to do the same. Considering that traffic generation is taking place after every second, and other conditions remaining same it was found, that the first sensor node death occurs approximately 95-96 days after their deployment. It is obvious that when sensor nodes are going to the sleep mode it may gain some energy and may come back to active mode. During its sleep time any other CH may take the responsibility of the cluster.

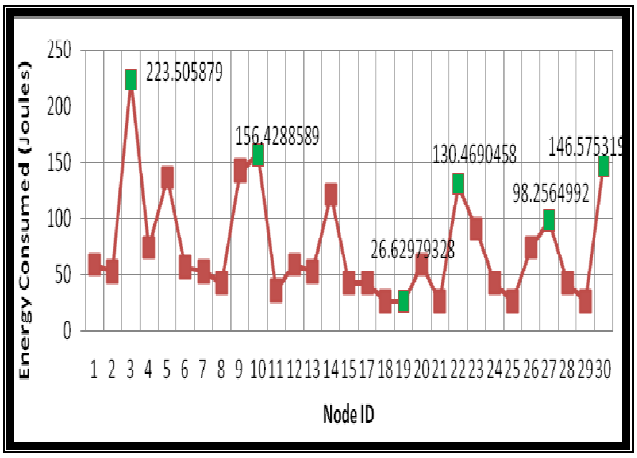

Figure 7: Energy Consumed by each Sensor Node in 24 hours

VI. CONCLUSION

This section concludes the work with future directives. After studying a few related researches in WSN, this paper proposes an energy aware fault tolerant framework. This paper also proposes a fault detection algorithm applicable in fault detection layer. Evaluation of this proposed algorithm through simulation and computation at different levels of sensor node energy consumption along with fault detection is discussed.

Moreover, it approximately predicts the first sensor node death in the network. In future, this research will rigorously work for predicting the faults to maintain the health of WSN. Even if, fault prediction is not possible always, then this model will enhance its functionality for recovering the fault as well.

REFERENCES

[1] Anastasi G., M. Conti, M. D. Francesco, A. Passarella "Energy Conservation in Wireless Sensor Networks: A Survey" in Ad Hoc Networks Journal, published by Elsevier, on May 2009, Vol. 7 Issue 3, pp. 537-568.

[2] Bari A., A. Jaekel, J. Jiang, Y. Xu "Design of fault tolerant wireless sensor networks satisfying survivability and lifetime requirements" in Computer Communications Journal, published by Elsevier, on Feb 2012, Vol. 35 Issue 3, pp. 320–333.

[3] Beder D. M., J. Ueyama, M. L. Chaim "A Generic Policy-free Framework for Fault-tolerant Systems: Experiments on WSNs" in Proc. of International Conference on Networked Embedded Systems for Enterprise Applications (NESEA), published by IEEE, on Dec 2011, pp. 1-7.

[4] Chen J., S. Kher, A. Somani "Distributed Fault Detection of Wireless Sensor Networks" in Proc. of Workshop on Dependability issues in Wireless Ad hoc Networks and Sensor Networks (DIWANS), published by ACM, on Sept 2006; pp. 65-72.

[5] Dutta R., S. Gupta, M. K. Das "Power Consumption and Maximizing Network Lifetime during Communication of Sensor Node in WSN" in Proc. of International Conference on Computer, Communication, Control and Information Technology(C3IT) published by Elsevier Procedia Technology Vol. 4 on Feb 2012 pp. 158-162

[6] Gupta P. C. "Data Communications & Computer Networks" Eastern Economy Edition, book published by Prentice Hall of India Pvt. Ltd. on 2006, pp. 254-256.

[7] Halgamuge M. N., M. Zukerman, K. Ramamohanarao, H. L. Vu "An Estimation of Sensor Energy Consumption" published in Progress In Electromagnetics Research B (PIER B) Journal, Vol. 12, on 2009, pp. 259–295

[8] Jiang P. "A New Method for Node Fault Detection in Wireless Sensor Networks" published in Sensors Journal, Vol. 9 Issue 2, on Feb 2009, pp. 1282-1294

[9] Jun Z., C. Xiangguang, L. Chuntao "The Self-diagnose Algorithm for Liquid Level Sensor on WSN Node" in Proc. Of International Conference on Intelligent System Design and Engineering Application (ISDEA), published by IEEE, on Jan 2012, pp. 1335-1338

[10] Kalpakis K., K. Dasgupta, P. Namjoshi, "Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks" in the Proc. of the IEEE International Conference on Networking (ICN'02), Atlanta, Georgia, August, 2002. pp. 685-696.

[11] Kulothungan K., J. A. A. Jothi, A. Kannan "An Adaptive Fault Tolerant Routing Protocol with Error Reporting Scheme for Wireless Sensor Networks" published in European Journal of Scientific Research, on 2011, Vol.60, No.1, pp. 19-32.

[12] Kumar S., A. Arora, T.H. Lai, "On the Lifetime Analysis of Always-on Wireless Sensor Network Applications" in Proc. of International Conference on Mobile Adhoc and Sensor Systems (MASS), published by IEEE on Nov 2005, pp. 188-190.

[13] Lee M. H., Y. H. Choi "Fault detection of wireless sensor networks" in Computer Communications Journal, published by Elsevier, on Sept 2008, Vol. 31 Issue 14 pp. 3469–3475.

[14] Mahapatroa A., P. M. Khilar "Transient Fault Tolerant Wireless Sensor Networks" in Proc. of International Conference on Computer, Communication, Control and Information Technology (C3IT) published by Elsevier Procedia Technology, on Feb 2012 pp. 97 – 101.

[15] http://www.dinesgroup.org/projects/images/pdf_files/iris_datasheet.pdf as accessed on 8th Dec 2013.

[16] Mitra S., A. D. Sarkar, S. Roy "A Review of Fault Management System in Wireless Sensor Network" in Proc. of International Information Technology Conference, CUBE published by ACM, on Sept 2012, pp 144-148.

[17] Mojoodi A. , M. Mehrani, F. Forootan, R.Farshidi "Redundancy Effect on Fault Tolerance in Wireless Sensor Networks" published in Global Journal of Computer Science & Technology, on 2011, Vol. 11 Issue 6, pp. 35-39.

[18] Oh H., I. Doh, K. Chae "A Fault Management and Moniroting Mechanism for Secure Medical Sensor Network" published in International Journal of Computer Science and Applications (IJCSA), on May 2009, Vol. 6, Issue 3, pp. 43–56.

[19] Oh S. H., C. O. Hong, Y. H. Choi "A Malicious and Malfunctioning Node Detection Scheme for Wireless Sensor Networks" in Wireless Sensor Network Journal, published by Scientific Research, on Mar 2012, Vol. 4 Issue 3, pgs 84-90.

[20] Polastre J., J. Hill, D. Culler "Versatile Low Power Media Access for Wireless Sensor Networks" in Proc. of International Conference on Embedded Networked Sensor Systems, published by ACM, on Nov 2004 pp. 95-107.

[21] Shahin F. "Zigbee Wireless Networks and Transceivers", book published by Elsevier in Imprint: Newnes, on Nov 2008 pp. 25-32.

[22] Simek M., P. Moravek, J.S. Silva "Wireless Sensor Networking in Matlab: Step-by-Step" in Proc. of ICT, on 2011, pp. 185-190.

[23] Stallings W. "Data and Computer Communications" 8th Edition, book published by Pearson Education Inc. and Dorling Kindersley Publishing Inc. on 2006, pp. 105-107, 111.

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

[25] Zhipeng G., H. Rimao, C. Yinghui, R. Lanlan "A Method for Node Fault Detection in Wireless Sensor Networks" published in Journal of China Communications, on 2011 Vol. 8 Issue 1, pp. 28-34.