ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 9, September 2015*

# Fault Node Detection and Recovery in Wireless Sensor Network (WSN)

**Pranali Shejwal[1], Asmita Mokal[2], Sonali Mujagule[3], Asha Thoke[4], Priyanka Narode[5]**

**Abstract**: This paper proposes a fault node recovery algorithm to increase the lifespan of a wireless sensor network when some of the sensor nodes are getting down. The algorithm is based on the grade diffusion algorithm combined with the genetic algorithm. The algorithm can result into detection and replacement of fewer fault node and more reused routing paths.

**Keywords**: Smart Sensor Nodes, Grade Diffusion Algorithm, Gradient Diffusion Algorithm, Fault Node Recovery Algorithm, Wireless, Sink, Sensing Node.

## 1. INTRODUCTION

In sensor networks, each sensor node has limited wireless computational power to process and transfer the live data to the base station or data collection centre. A fault node recovery (FNR) algorithm is used to enhance the lifetime of a wireless sensor network (WSN) when some of the sensor nodes shut down, either because they no longer have battery energy or they have reached their operational threshold. Using the FNR algorithm can result in fewer replacements of sensor nodes and more reused routing path.

## 2. RELATED WORK

The traditional approaches to sensor network routing include the fault node recovery algorithm [1], Directed Diffusion (DD) algorithm [5] and Grade Diffusion (GD) algorithm [7]. The algorithm proposed in this paper is based on the GD algorithm, with the goal of replacing fewer sensor nodes that are inoperative or have depleted batteries, and of reusing the maximum number of routing paths. These optimizations will ultimately enhance the WSN lifetime and reduce sensor node replacement cost.

### *GRADE DIFFUSION (GD) ALGORITHM*

The Grade Diffusion (GD) algorithm [8] was developed in 2012 to improve the ladder diffusion algorithm using ant colony optimization (LD-ACO) for wireless sensor networks. The GD algorithm not only creates the routing for each sensor node but also identifies a set of neighbour nodes to reduce the transmission loading. Each sensor node can select a sensor node from the set of neighbour nodes when its grade table lacks a node able to perform the relay. The GD algorithm can also record some information regarding the data relay. Then, a sensor node can select a node with a lighter loading or more available energy than the other nodes to perform the extra relay operation. That is, the GD algorithm updates the routing path in real time, and the event data is thus sent to the sink node quickly and correctly. Whether the DD or the GD algorithm is applied, the grade-creating packages or interested query packets must first be broadcast. Then, the sensor nodes transfer the event data to the sink node, according to the algorithm, when suitable events occur. The sensor routing paths are shown in Fig. 1

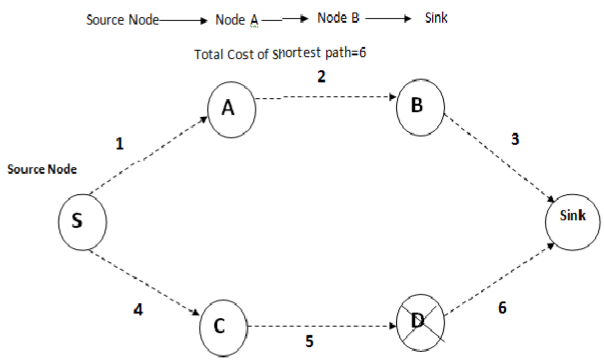


*fig 1. WSN Processing model*

## 3. RECOVERY OF DETECTED FAULT NODE USING GENETIC ALGORITHM

Genetic algorithm is suitable for parallel processing. There are operations that can be executed without data dependencies. For example, in fig.2 the evaluation process in Step B can be parallelized because each chromosome is independent of the others. If we have only one processor, the following steps cause inefficiency of memory access. The problem is that the temporal locality can only be well preserved with difficulty. In Step D and Step E, we evolve the old population to produce the new one. Next, we iterate through the population again to compute the fitness of each chromosome. If the population size is large, one chromosome is generated (in Step D and Step E) and temporarily stored in the cache, written back to the main memory due to the subsequently generated chromosomes, and then loaded into the cache again while evaluating the fitness value. Because a cache miss results in an extra data fetching procedure, the performance is degraded by the penalty cycles. Genetic algorithms are stochastic adaptive global search and optimization methods, founded on the principles of natural selection and population genetics. The population is comprised by several individuals that represent potential solutions to the problem. The basic operations first carried out by a GA are the generation and evaluation of an initial population. Subsequently, a main loop of operations that include selecting the best individuals, crossing them and applying mutation on every

 DOI 10.17148/IJARCCE.2015.49115 

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 9, September 2015*

locus (string position) is performed. The population evolves within this loop and the best string present in the final population is considered to be the solution, as convergence is reached. The main computational effort in a traditional GA resides in the evaluation of the strings quality, i.e. the estimation of the fitness function. An ideal fitness function should correlate closely with the algorithm goal, and yet should be computed quickly. The execution speed is very important, as each generation of the genetic algorithm must be iterated many times in order to produce a useable result for a non-trivial problem. Usually, the fitness function is a complex mathematical function with many parameters, while the operators like selection, mutation, crossover and replacement are of linear complexity, often working at constant rates for a given Problem.

This paper proposes a fault node recovery (FNR) algorithm for WSNs based on the grade diffusion algorithm combined with the genetic algorithm. The flow chart is shown in Fig. 2. The FNR algorithm creates the grade value, routing table, neighbour nodes, and payload value for each sensor node using the grade diffusion algorithm. In the FNR algorithm, the number of non-functioning sensor nodes is calculated during the wireless sensor network operation. , and the parameter $B_{th}$ is calculated according to equation (1).

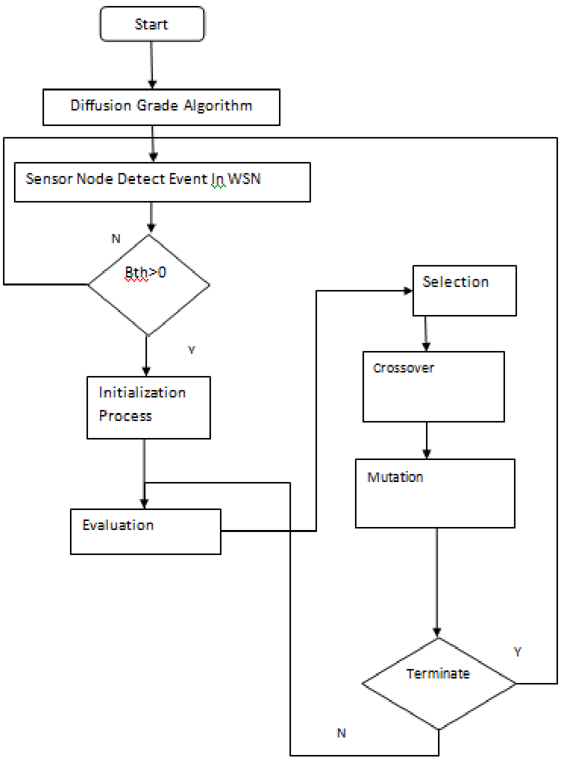


*Fig 2. Fault node recovery algorithm flow chart*

### *STEPS FOR GENETIC ALGORITHM:*

There are five steps of genetic algorithm.

#### *A. Initialization*

In the initialization step, the genetic algorithm generates chromosomes, and each chromosome is an expected solution. The number of chromosomes is determined according to the population size, which is defined by the user. Each chromosome is a combination solution, and the chromosome length is the number of sensor nodes that are depleted or non-functioning.

#### *B. Evaluation*

In general, the fitness value is calculated according to the chromosome's genes. However, we cannot put genes directly into the fitness function in the FNR algorithm, because the genes of the chromosome are simply whether the node should be replaced or not. In the FNR algorithm, the goal is also to reuse the most routing paths and to replace the fewest sensor nodes. Hence, the number of routing paths available if some non-functioning sensor nodes are replaced is calculated, and the fitness function is shown as equation (3)

$$B_{th} = \sum_{i=1}^{\max(grade)} T_i \qquad (1)$$

$$T_i = \begin{cases} 1 & ; \dfrac{N_i^{now}}{N_i^{original}} < \beta \\ 0 ; otherwise \end{cases} \qquad (2)$$

$$f_n = \sum_{i=1}^{\max(grade)} \left( \frac{P_i * TP^{-1}}{N_i * TN^{-1}} \right) * i^{-1} \qquad (3)$$

Where,

$N_i$ = the number of replaced sensor nodes and their grade value at i.

$P_i$ = the number of re-usable routing paths from sensor nodes with their grade value at i.

T N = total number of sensor nodes in the original WSN.

T P = total number of routing paths in the original WSN.

In step B, a high fitness value is sought because the WSN is looking for the most available routing paths and the least number of replaced sensor nodes.

#### *C. Selection*

The selection step will eliminate the chromosomes with the lowest fitness values and retain the rest. We use the elitism strategy and keep the half of the chromosomes with better fitness values and put them in the mating pool. The worse chromosomes will be deleted, and new chromosomes will be made to replace them after the crossover step.

#### *D. Crossover*

The crossover step is used in the genetic algorithm to change the individual chromosome. Two individual chromosomes are chosen from the mating pool to produce two new offspring. A crossover point is selected between the first and last genes of the parent individuals. Then, the

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 9, September 2015*

fraction of each individual on either side of the crossover point is exchanged and concatenated. The rate of choice is made according to roulette-wheel selection and the fitness values.

### *E. Mutation*

The mutation step can introduce traits not found in the original individuals and prevents the GA from converging too fast. In this algorithm, we simply flip a gene randomly in the chromosome. The chromosome with the best fitness value is the solution after the iteration. The FNR algorithm will replace the sensor nodes in the chromosome with genes of 1 to extend the WSN

## 4. CONCLUSION

This algorithm will improve the efficiency of network maintenance and fault repairs. If some module fails in Agent, we can still take some actions to recover this emergency. We extend the working time of the failure node as long as possible. Within this amount of time, we try to repair the fault. We can avoid the cost of updating routing tables of the entire network caused by replacing the failure node. A fault node recovery algorithm for WSN based on the grade diffusion algorithm combined with a genetic algorithm. The FNR algorithm requires replacing fewer sensor nodes and reuses the most routing paths, increasing the WSN lifetime and reducing the replacement cost.

## ACKNOWLEDGMENT

We would like to acknowledge the kind and thorough guidance provided by our guide **Dr. Anil B. Pawar** throughout this project work.

## REFERENCES

[1] Hong-Chi Shih, Jiun-Huei Ho, Bin-Yih Liao and Jeng-Shyang Pan, ”Fault Node Recovery Algorithm for a Wireless Sensor Network”, Accessed july 7 2015.

[2] M. Gen and R. Cheng, “Genetic Algorithms and Engineering Design”. New York, NY, USA: Wiley, 1997.Accessed july 9 2015.

[3] S. Corson and J. Macker, “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”. New York, NY, USA: ACM, 1999 july 14 2015.

[4] E. M. Royer and C. K. Toh, “A review of current routing protocols for ad-hoc mobile networks”, EEE Personal Commun., vol. 6, no. 2,pp. 4655, Apr. 1999. Accessed july 14 2015.

[5] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, “Directed diffusion for wireless sensor networking”, IEEE/ACM Trans. Netw., vol. 11, no. 1, pp. 216, Feb. 2003.Accessed july 14 2015.

[6] J. Pan, Y. Hou, L. Cai, Y. Shi, and X. Shen, “Topology control for wireless sensor networks”, in Proc. 9th ACM Int. Conf. Mobile Comput. Netw., 2003, pp. 286299,Accessed july 14 2015

[7] J. H. Ho, H. C. Shih, B. Y. Liao, and J. S. Pan, “Grade diffu-sion algorithm,” in *Proc. 2nd Int. Conf. Eng. Technol. Innov.*, 2012, pp. 2064–2068.

[8] J. H. Ho, H. C. Shih, B. Y. Liao, and S. C. Chu, “A ladder diffusion algorithm using ant colony optimization for wireless sensor networks,” *Inf. Sci.*, vol. 192, pp. 204–212, Jun. 2012.

[9] J. A. Carballido, I. Ponzoni, and N. B. Brignole, “CGD-GA: A graph-based genetic algorithm for sensor network design”, Inf. Sci., vol. 177, no. 22, pp. 50915102, 2007, Accessed july 14 2015.

[10] Z. He, B. S. Lee, and X. S. Wang, “Aggregation in sensor networks with a user-provided quality of service goal”, Inf. Sci., vol. 178, no. 9, pp. 21282149, 2008, Accessed july 2015.