Fault Node Recovery Algorithm for a Wireless Sensor Network

Hong-Chi Shih, Student Member, IEEE, Jiun-Huei Ho, Bin-Yih Liao, Member, IEEE, and Jeng-Shyang Pan, Senior Member, IEEE

Abstract—This paper proposes a fault node recovery algorithm to enhance the lifetime of a wireless sensor network when some of the sensor nodes shut down. The algorithm is based on the grade diffusion algorithm combined with the genetic algorithm. The algorithm can result in fewer replacements of sensor nodes and more reused routing paths. In our simulation, the proposed algorithm increases the number of active nodes up to 8.7 times, reduces the rate of data loss by approximately 98.8%, and reduces the rate of energy consumption by approximately 31.1%.

Index Terms—Genetic algorithm, grade diffusion (GD) algorithm, gradient diffusion algorithm, wireless sensor networks (WSN).

I. INTRODUCTION

RECENT advances in micro processing, wireless and battery technology, and smart sensors have enhanced data processing [3], [11], [13], wireless communication, and detection capability. 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 center [2], [5], [8]. Therefore, to increase the sensor area and the transmission area [1], [12], the wireless sensor network usually contains many sensor nodes. Generally, each sensor node has a low level of battery power that cannot be replenished. When the energy of a sensor node is exhausted, wireless sensor network leaks will appear, and the failed nodes will not relay data to the other nodes during transmission processing. Thus, the other sensor nodes will be burdened with increased transmission processing.

This paper proposes a fault node recovery (FNR) algorithm 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 paths. Thus, the algorithm not only enhances the WSN lifetime but also reduces the cost of replacing the sensor nodes.

II. RELATED WORK

The traditional approaches to sensor network routing include the directed diffusion (DD) [9] algorithm and the grade diffusion (GD) [13] algorithm. 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.

A. Directed Diffusion Algorithm

A series of routing algorithms [10], [14] for wireless sensor networks have been proposed in recent years. C. Intanagonwiwat et al. presented the Directed Diffusion (DD) algorithm [9] in 2003. The goal of the DD algorithm is to reduce the data relay transmission counts for power management. The DD algorithm is a query-driven transmission protocol. The collected data is transmitted only if it matches the query from the sink node. In the DD algorithm, the sink node provides the queries in the form of attribute-value pairs to the other sensor nodes by broadcasting the query packets to the whole network. Subsequently, the sensor nodes send the data back to the sink node only when it fits the queries.

B. Grade Diffusion Algorithm

H. C. Shih et al. presented the Grade Diffusion (GD) algorithm [7] in 2012 to improve the ladder diffusion algorithm using ant colony optimization (LD-ACO) for wireless sensor networks [6]. The GD algorithm not only creates the routing for each sensor node but also identifies a set of neighbor nodes to reduce the transmission loading. Each sensor node can select a sensor node from the set of neighbor 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.
The WSN may fail due to a variety of causes, including the following: the routing path might experience a break; the WSN sensing area might experience a leak; the batteries of some sensor nodes might be depleted, requiring more relay nodes; or the nodes wear out after the WSN has been in use a long period of time. In Fig. 2, the situation in which the outside nodes transfer event data to the sink node via the inside nodes (the sensor nodes near the sink node) in a WSN illustrate the accommodation measures for non-working nodes. The inside nodes thus have the largest data transmission loading, consuming energy at a faster rate. If all the inside nodes deplete their energy or otherwise cease to function, the event data can no longer be sent to the sink node, and the WSN will no longer function.

III. FAULT NODE RECOVERY ALGORITHM

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. 3. The FNR algorithm creates the grade value, routing table, neighbor nodes, and payload value for each sensor node using the grade diffusion algorithm. In the FNR algorithm, the number of nonfunctioning sensor nodes is calculated during the wireless sensor network operation, and the parameter $B_{th}$ is calculated according to (1).
In Fig. 3, the FNR algorithm creates the grade value, routing table, a set of neighbor nodes, and payload value for each sensor node, using the grade diffusion algorithm. The sensor nodes transfer the event data to the sink node according to the GD algorithm when events appear. Then, $B_{th}$ is calculated according to (1) in the FNR algorithm. If $B_{th}$ is larger than zero, the algorithm will be invoked and replace nonfunctioning sensor nodes by functional nodes selected by the genetic algorithm. Then the wireless sensor network can continue to work as long as the operators are willing to replace sensors.

$$B_{th} = \max \{ \text{Grade} \} \sum_{i=1}^{N_{i}} T_i$$

$$T_i = \begin{cases} 1, & \frac{N_{i}^{\text{new}}}{N_{i}^{\text{original}} < \beta} \\ 0, & \text{otherwise} \end{cases}$$

In (1), Grade is the sensor node’s grade value. The variable $N_{i}^{\text{original}}$ is the number of sensor nodes with the grade value $i$. The variable $N_{i}^{\text{now}}$ is the number of sensor nodes still functioning at the current time with grade value $i$. The parameter $\beta$ is set by the user and must have a value between 0 and 1. If the number of sensor nodes that function for each grade is less than $\beta$, $T_i$ will become 1, and $B_{th}$ will be larger than zero. Then, the algorithm will calculate the sensor nodes to replace using the genetic algorithm.

The parameters are encoded in binary string and serve as the chromosomes for the GA. The elements (or bits), i.e., the genes, in the binary strings are adjusted to minimize or maximize the fitness value. The fitness function generates its fitness value, which is composed of multiple variables to be optimized by the GA. At each iteration of the GA, a predetermined number of individuals will produce fitness values associated with the chromosomes.

There are 5 steps in the genetic algorithm: Initialization, Evaluation, Selection, Crossover, and Mutation. Descriptions of the steps follow.

**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 nonfunctioning. The elements in the genes are either 0 or 1. A 1 means the node should be replaced, and a 0 means that the node will not be replaced.

Fig. 4 represents a chromosome. The chromosome length is 10 and the gene is 0 or 1, chosen randomly in the initialization step. In this case, there are 10 sensor nodes not functioning, and their node numbers are 9, 7, 10, 81, 23, 57, 34, 46, 66, and 70.

**B. Evaluation**

In general, the fitness value is calculated according to a fitness function, and the parameters of the fitness function are 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 nonfunctioning sensor nodes are replaced is calculated, and the fitness function is shown as (2)

$$f_n = \max \{ \text{Grade} \} \sum_{i=1}^{N_{i}} \frac{P_i \times TP^{-1}}{N_{i}^{\text{original}} \times TN^{-1}} \times i^{-1}.$$ 

In (2):

- $N_{i}$ = 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 $i$.
- $TN$ = total number of sensor nodes in the original WSN.
- $TP$ = total number of routing paths in the original WSN.

In (2), 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. The process is shown in Fig. 5.

**D. Crossover**

The crossover step is used in the genetic algorithm to change the individual chromosome. In this algorithm, we use the
In the simulation, the proposed algorithm increases the number of active nodes up to 8.7 times. The number of active nodes is enhanced 3.16 times on average after replacing an average of 32 sensor nodes for each calculation. The algorithm reduces the rate of data loss by approximately 98.8% and reduces the rate of energy consumption by approximately 31.1%. Therefore, the FNR algorithm not only replaces sensor nodes, but also reduces the replacement cost and reuses the most routing paths to increase the WSN lifetime.

REFERENCES


