Fault Recovery Algorithm for WSNs in Smart Grid

Yanfei Lin[1], Xingang Wang[2], Cheng Zhong[3], Sujie Shao[1], and Xuesong Qiu[1]
[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]State General Administration of Press and Publication, Radio and Television Broadcasting and Television Satellite Broadcast Management Center, Beijing 100045, China
[3]Power Grid of Hebei, Hebei 050022, China
Email: lyf@bupt.edu.cn; buptssj@bupt.edu.cn

Abstract —A large number of sensors are deployed in smart grid for monitoring and metering of the power equipment and other key facilities as well as users' behavior. The collected information is sent to data processing center for analyzing and processing. Because of the relatively complicated and harsh working conditions of sensors in smart grid, sensor fault occurs frequently. To ensure the normal execution of monitoring and metering, timely and accurate recovery of sensor fault is needed. Therefore, this paper proposes a kind of fault recovery algorithm for WSNs in smart grid. Firstly, sensor fault is classified according to the degree of influence on data monitoring and communication. Then different fault recovery methods are designed according to different fault types, and suitable alternative nodes are selected to restore the network function. Simulation results show that the proposed algorithm can realize sensor fault recovery using as little communication cost as possible, and extend the lifecycle of sensor network on the premise of ensuring coverage rate of target nodes.

Index Terms—Smart grid, wireless sensors, fault recovery, alternative nodes, coverage rate

I. INTRODUCTION

The power distribution equipment which is numerous and widely distributed in smart grid is the critical infrastructure for smart grid operation; users' behavior represents the operation status of smart grid. Deploying a large number of sensors to monitor and meter them, as well as transmitting and analyzing the collected data is one of the keys in smart grid operation [1], [2]. On account of the complicated and harsh working conditions of smart grid equipment, serious uncertainty exists. Perceiving data incorrectly or even permanent fault of sensors occurs easily due to vibration, noise, channel interference, fire and so on, which will affect the whole sensor network in connectivity and real-time monitoring [3]-[5]. As a result, sensor fault recovery algorithms which can efficiently repair WSNs in time and ensure the normal operation of monitoring and metering in smart grid become one of the hot research spots [6], [7].

Manuscript received May 15, 2016; revised September 20, 2016.
This work was supported by "the Fundamental Research Funds for the Central Universities".
Corresponding author email: lyf@bupt.edu.cn.
doi:10.12720/jcm.11.9.827-833

Lots of literature has studied the sensor fault recovery methods. A fault management mechanism and algorithm for WSNs is proposed in [8], it restores the network connectivity based on transpose nodes and connected set. A kind of fault recovery algorithm for WSNs based on network segmentation is proposed in [9], focusing on the communication requirements of WSNs and restoring the network communication capability based on moving neighbor nodes. But these two methods are based on moving neighbor nodes. It is not suitable for smart grid environment, in which the sensor nodes are not easily to move [10]. Literature [11] divides the monitoring region with concentric circles based on the strategy of waking up sleeping nodes. It selects regional transpose nodes to reconstruct the connectivity of network. But this method is easily to cause communication congestion, and the efficiency is low. Literature [12] uses as little communication data as possible to recover fault, but it ignores the task of monitoring target nodes.

Based on the analysis above, two questions need to be addressed for sensor fault recovery in smart grid. First, the influence on monitoring and communication ability. Second, the selection strategy of substitute nodes to restore network connectivity. Fault nodes at the edge of the network may only affect data monitoring, but that in network center also affect relay communication. Therefore, to improve the speed and efficiency of fault recovery and guarantee the data communication in most monitoring areas, we should give priority to restore network communication function. So sensor fault is classified according to the degree of influence on data monitoring and communication function. To prolong the lifecycle of WSNs in smart grid, there are a large number of sensor nodes in sleeping state on the premise of guaranteeing monitoring and communication. When choosing substitute nodes, we first determine whether the current active nodes can replace the fault nodes to complete task, if not, then we need to activate appropriate sleeping nodes as substitute to restore the network function.

To this end, this paper proposes a kind of fault recovery algorithm for WSNs in smart grid. Firstly, sensor fault is classified according to the degree of influence on data monitoring and communication function. Then different fault recovery strategy is designed according to different fault types, and suitable alternative nodes are selected to restore the network function. Simulation results show that the proposed algorithm can achieve reliable and fast sensor fault recovery using as little data traffic as possible and extend the lifecycle of sensor network.

The rest of this paper is organized as follows. Section II presents the problem model. Section III proposes the fault recovery algorithm for WSNs in smart grid. The section IV gives analysis of simulation results followed by conclusion in Section V.

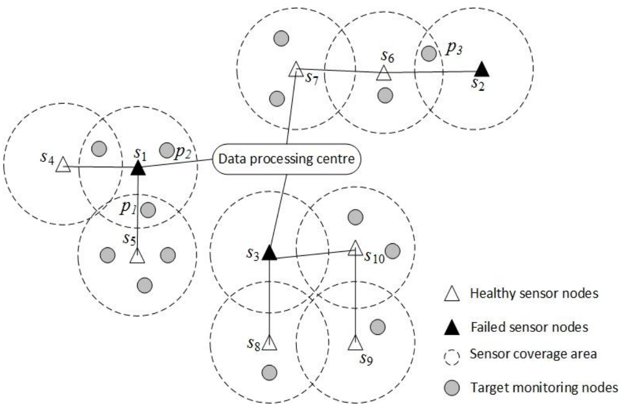

Fig. 1. WSNs in smart grid.

II. Problem Model

The WSNs in smart grid is shown in Fig. 1, which mainly compose of remote data processing center, sensor nodes as well as users and devices need to be monitored. When the sensor nodes are failed, first of all, we need to determine the fault type according to the influence on monitoring and communication. The fault types can be divided into the following three kinds:

- Influence on both monitoring and data communication: when node s_1 is failed, new nodes should be found to monitor target nodes p_1, p_2 and forward data to s_4, s_5;
- Only influence on monitoring: when node s_2 is failed, new node should be found to monitor target node p_3;
- Only influence on data communication: when node s_3 is failed, new nodes should be found to forward the data from target nodes s_8, s_{10}.

Fault recovery is carried out by selection of alternative nodes based on the result of fault classification. The alternative nodes can be chosen from active sensors or activated from sleeping sensors. According to different fault types, a single node or a node set should be selected to replace the faulty node and restore the monitoring and communication function. In addition, to balance the energy distribution of sensors and extend the lifecycle of sensor networks, whether we need to activate nodes and chose the optimal activation scheme requires further consideration.

III. Fault Recovery Algorithm for WSNs in Smart Grid

A. Fault Type Judgement

The data processing center in smart grid records the location information of all the sensors and target nodes in network. Each sensor needs to maintain three tables, a target nodes information table to record the location information of target nodes it can cover, and two neighbor nodes information table to record its neighbor nodes in active and sleeping state respectively.

The methods for acquiring information of failed sensors can be divided into two kinds according to whether they can send and receive data.

When the failed sensors can send and receive data, it send fault requests $F_{req} = (P_{cov}, S_{neiact}, S_{neislp}, TTL)$ to its neighbors, in which P_{cov} is the set of target nodes it can cover, S_{neiact} is the one and two hop active neighbor nodes set, S_{neislp} is the sleeping neighbor nodes set, TTL is the lifecycle of requests. In this paper, we set $TTL = 2$, when neighbors receive fault requests, it will decrease the TTL by 1. If the TTL value is 0, it will not forward the request to neighbors. In this way, the fault requests are only sent to one and two hop neighbor nodes. Data congestion due to flooding can be efficiently avoided.

When failed sensors lose the function of sending data, the failed sensors need to be detected by its neighbors. The active sensor nodes periodically send inquiry requests ask to its neighbors. Their neighbors received the request need to reply an ack. If a neighbor cannot be heard by ack after three times of ask requests, then the neighbor is determined to be failed. Then the fault information can only be acquired from the data processing center. Generally, a failed sensor can be detected by all of its neighbors, but all the neighbors send requests to data processing center is not necessary. It will bring a large amount of transmission cost. To this end, assuming that the fault node is i, and it has M active neighbors numbered $1, 2, \cdots, M$ in ascending order of distance. When a node finds that its neighbor is failed, it will first wait for T_{wait}^{num} time, as shown in formula (1).

$$T_{wait}^{num} = (num - 1) \times (T_{mage} + \alpha) \tag{1}$$

where $num \in (1, M)$ is No. of neighbors, T_{mage} is round-trip time of sending data to data processing center. We set α to avoid the situation that sensors cannot receive reply message in T_{mage} due to data delaying. Data processing center sends message of the fault node to all of its neighbor nodes after receiving fault requests. If a neighbor node receives the required message in waiting time, the corresponding request will not be sent. Otherwise, after waiting time arrived, request will be sent to data processing center. By setting T_{wait}^{num}, the situation that when a node is failed, all of its neighbor nodes send requests to data processing center in the same period and causing data congestion can be efficiently avoided.

After obtaining information of failed sensor, the classification of fault type can be completed according to the flow chart shown in Fig. 2. Firstly, we need to judge whether the fault has an impact on data communication according to S_{neiact}. If there is only one neighbor node in S_{neiact}, it will not affect data communication, otherwise it will. Secondly, we need to determine whether the fault has an impact on data monitoring according to whether the number of nodes in P_{cov} is 0. To prolong the lifecycle of sensor network, other sensor nodes are sleeping under the premise of finishing monitoring and data communication tasks in smart grid. So the fault which has no effect on sensor network does not exist. Sensor nodes fault can be divided into three types: impact monitoring, impact communication, impact monitoring and communication. We call them monitoring fault, communication fault and complete fault respectively.

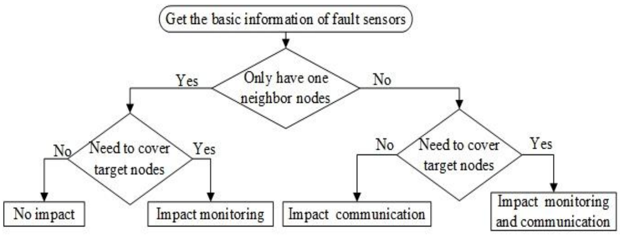

Fig. 2. Flow chart of sensor fault type judgment.

B. Alternative Nodes Selection

For the monitoring fault and complete fault, we first choose monitoring nodes to monitor target nodes of the fault nodes. Then fault recovery is completed if there is no communication fault, otherwise we will start to select communication nodes. For communication fault, we only need to select communication nodes.

1) Monitoring nodes selection

Monitoring nodes are firstly selected from the one hop and two hop active neighbor nodes of failed nodes. As data monitoring in smart grid needs higher reliability, and in order to avoid the situation that sensors which are activated for a short time break down quickly due to energy deficiency, the selection of alternative nodes should give priority to the nodes that have more remaining energy and cover less target nodes. So the selection of monitoring nodes should consider three factors: accuracy of monitoring, remaining energy and number of current coverage nodes.

The accuracy of monitoring is reflected by intensity of monitoring signal. The signal intensity decreases gradually with the increasing of distance from target nodes to sensor nodes. To express the relationship between signal intensity and distance, a monitoring signal intensity model is established in Formula (2).

$$str(s,p)=\begin{cases}1, & d(s,p)\le r\\ \exp\left(\dfrac{-\gamma_2(d(s,p)-r)^{\eta_2}}{\left(R-d(s,p)\right)^{\eta_1}+\gamma_1}\right), & r<d(s,p)\le R\\ 0, & d(s,p)>R\end{cases} \tag{2}$$

where $d(s,p)$ is the distance between sensor node s and target node p, $\gamma_1,\gamma_2,\eta_1,\eta_2$ are parameters related to physical properties of sensors, R is the monitoring radius of sensors. We set an parameters r, when $d(s,p)\le r$, the accuracy of monitoring is very high and $str(s,p)=1$. When $r<d(s,p)\le R$, the signal intensity gradually weaken with the increase of distance. When $d(s,p)=R$, $str(s,p)=\exp(-\gamma_2(R-r)^{\eta_2}/\gamma_1)$, which is the minimum signal intensity required for effectively monitoring.

Then we need to calculate the distance $d(s,p)$ between each $s\in S_{neiact}$ and $p\in P_{cov}$, when $d(s,p)\le R$, we say s can monitor p. If only one sensor can monitor p, the sensor should be selected directly. When many sensors can monitor p, we need to select the best sensor node with coverage weight w_{act} in formula (3).

$$w_{act}=\beta_1 str(s,p)+\beta_2 E_{surp}+\beta_3\frac{1}{P_{num}} \tag{3}$$

where E_{surp} represents the remaining energy, P_{num} is the number of target nodes s can cover. Three weighting parameters β_1,β_2,β_3 are determined according to the actual situation, because the energy used in data acquisition process is a little [8], the value of β_3 is small. We can sort the active neighbor sensors according to the value of w_{act}, and select a sensor with the largest value to cover p, then we can remove p from P_{cov}. Repeat the process above until P_{cov} is empty, or suitable active sensors to cover target nodes in P_{cov} no longer exist. If P_{cov} is empty, monitoring nodes selection is finished, otherwise we need to activate appropriate sleeping sensor nodes from S_{neislp}.

The selection of sleeping nodes should also consider the three factors above. The difference between the selection from sleeping nodes and active nodes is that we choose sleeping nodes that can cover more target nodes, which can decrease the number of sensors added to networks as far as possible, and prolong the lifecycle of sensor networks. Using formula (4) to calculate the coverage weight w_{slp} of each sleeping sensor can achieve combination of the three factors.

$$w_{slp}=\frac{\sum_{i=1}^{n_{cov}} str(s,p_i)}{n_{cov}}\chi_1+E_{surp}\chi_2+n_{cov}\chi_3 \tag{4}$$

The target nodes set that sleeping sensor s can cover is $(1,2\cdots n_{cov})$, n_{cov} is the number of target nodes, E_{surp} is the remaining energy of s, χ_1,χ_2,χ_3 are three weighted parameters, their values are according to the specific situation. For each $s_{slp}\in S_{neislp}$, we first calculate w_{slp}, and select the sensor with the maximum w_{slp} to activate. Then we can remove it from S_{neislp} and remove all the nodes in set $(1,2\cdots n_{cov})$ from P_{set}, repeat the process until P_{set} is empty, monitoring nodes selection is completed. Flow chart of monitoring nodes selection is shown in Fig. 3.

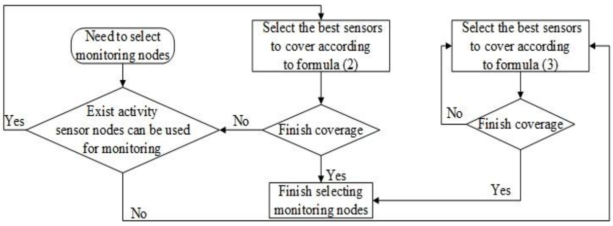

Fig. 3. Flow chart of monitoring nodes selection.

2) Communication nodes selection

Recovery of monitoring fault is finished if no sleeping sensor nodes are activated during the recovery process, otherwise fault may still exist. When newly activated sensors are unable to communicate with any of sensors in $s\in S_{neiact}$, the fault repair process in last section derives new communication fault, we call it the first kind of communication fault.

But in the recovery process of complete fault, after monitoring recovery is finished, the next step is to recover communication function. This communication fault exists not only between the sensors in $s\in S_{neiact}$, may also exists between the newly activated sensors and $s\in S_{neiact}$, we call it the second kind of communication fault.

The communication fault in the initial fault type judgment process only exists between sensors in $s\in S_{neiact}$, we call it the third kind of communication fault.

We combine the one hop neighbor nodes set $S_{neigone}$ of fault node with the newly activated neighbor nodes set $S_{neignew}$ as a new set $S_{neig}=\{S_{neigtwo},S_{neignew}\}$. For the first and second kind of communication fault, both $S_{neigone}$ and $S_{neignew}$ are not empty; for the third kind of communication fault, $S_{neigone}$ is not empty, but $S_{neignew}$ is empty. The fault recovery problem is translated into seeking relay sensor nodes to make the nodes in S_{neig} communicate normally.

We first group all sensor nodes in set S_{neig} based on hierarchical clustering algorithm. Each sensor node in the set is regarded as a group at the start. We calculate the distance between any two groups, and select two groups with the distance that is smallest and smaller than R_{tran} to merge, supposing the transmission radius of sensors is R_{tran}. When there is more than one sensor in groups, the distance between two groups is defined as the distance between the nearest sensor nodes from each group. Repeat the process above until no groups can be merged, then the grouping process is completed.

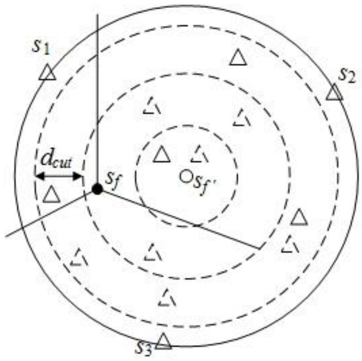

Fig. 4. Areas division of sleeping sensors.

Fig. 4 shows the areas division of sleeping sensors. The solid triangles are cluster head nodes, representing the sensors in each group having the shortest distance to fault node s_f, the dotted triangles are sleeping nodes in S_{neislp}. We can divide the plane into three equal parts with the three straight lines in Fig. 4. Then we select a head node with the farthest distance to s_f in each part, and they are respectively s_1,s_2,s_3. Although a sensor in the location of s_f can finish communication fault recovery, the difference of three distances between s_f and s_1,s_2,s_3 may be large, so does the quality of communication. So we need to choose sensors in the best location to activate. We first make a circumscribed circle of s_1,s_2,s_3, supposing its center is $s_{f'}$ and radius is $R_{f'}$. The distance between s_1,s_2,s_3 and $s_{f'}$ are $d_{1f'},d_{2f'},d_{3f'}=R_{f'}$. At this point, the position of sensor to be activated is transferred from s_f to $s_{f'}$, which can achieve a balance of the distance to s_1,s_2,s_3 and ensure higher communication quality to all the cluster head nodes. In particular, if there are only two cluster head nodes, the circumscribed circle is determined with distance between the two cluster head nodes as diameter and the midpoint of connecting line as center.

However, there may be no sleeping sensors in the location of $s_{f'}$, then we need to search for the best sleeping sensors to activate. We regard $s_{f'}$ as center and $nd_{cut}(n=1,2\cdots)$ as radius to make many concentric circle. Then all the sleeping sensor nodes in S_{neislp} are divided into different areas, including the smallest dotted circle and a plurality of concentric rings, as shown in Fig. 4. The best sleeping sensors that will be activated are selected in these areas in turn.

Because the distance $d_{if'}$ from sleeping sensors to $s_{f'}$ in the same area is close. We consider the distance factor of sleeping sensors in the same area is equal, formulating it with $\lceil d_{cut}/d_{if'} \rceil$. Combining the difference of distance between sleeping nodes and s_1, s_2, s_3, the distance factor and the remaining energy of sleeping sensors, we can calculate the communication weight w_{commi} of each sleeping sensor, as shown in formula (5).

$$w_{commi} = \delta_1 \frac{\sum_{j=1}^{3}(d_{ij}-u)^2}{3} + \delta_2 \left\lceil \frac{d_{cut}}{d_{if'}} \right\rceil + \delta_3 E_{surp} \qquad (5)$$

where three weighting parameter $\delta_1, \delta_2, \delta_3$ are determined by specific situation. d_{ij} is the distance between sleeping sensors i and $s_j \in s_1, s_2, s_3$, $u = \frac{1}{3}\sum_{j=1}^{3} d_{ij}$ is the mean square of the three distances.

We select the sensor with the largest w_{commi} to activate. Then we delete it from S_{neislp} and add it to S_{neig}. Afterwards, we need to group all the sensors in the updated set S_{neig} again and repeat the process above until all the sensors in set S_{neig} can communicate with each other. Then selection progress of communication node is finished and fault recovery is realized, as is shown in Fig. 5.

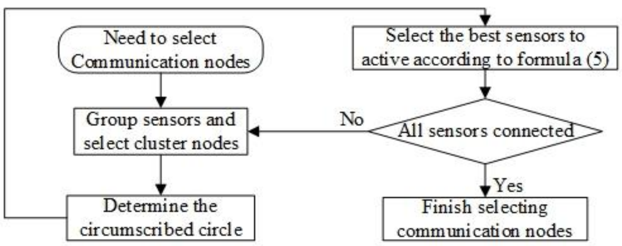

Fig. 5. Flow chart of communication nodes selection.

IV. Simulation Experients

In order to verify the performance of our algorithm, 10, 20, or 30 target nodes and 80 sensor nodes are scattered randomly in the area 100×100. The variation of the number of new activated nodes and coverage rate of target nodes is simulated in three different kinds of nodes density. In the situation of 80 sensor nodes and 30 target nodes, we compare our algorithm with [11] and [12] in two aspects: the number of message packets sent in the recovery process and the network lifecycle.

1) Number of activated nodes and coverage rate of target nodes

When the current active sensors cannot realize recovery, we need to activate sleeping sensor nodes. Fig. 6 shows the number of new active nodes changes with the failed nodes number in three kinds of node density. As can be seen in Fig. 6, the number of sensor nodes need to be activated increases slowly, as the number of fault nodes increases. Because when a sensor is failed, it determines firstly whether the current active nodes can replace the failed node. If it finds a substitute in active nodes, it does not need to activate sleeping sensors. We regard the number of target nodes that each node can cover as an important parameter during the selection process of monitoring nodes, so the demand of activating sleeping sensors can be reduced greatly.

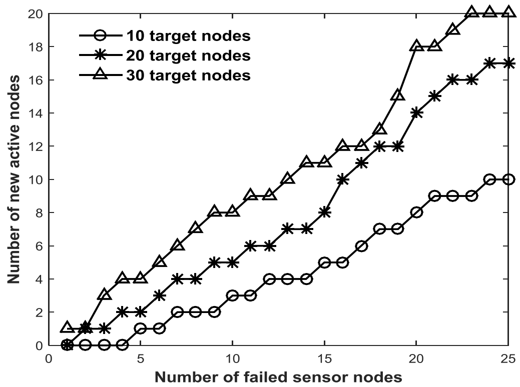

Fig. 6. The number of new activated nodes varies with the number of failed nodes.

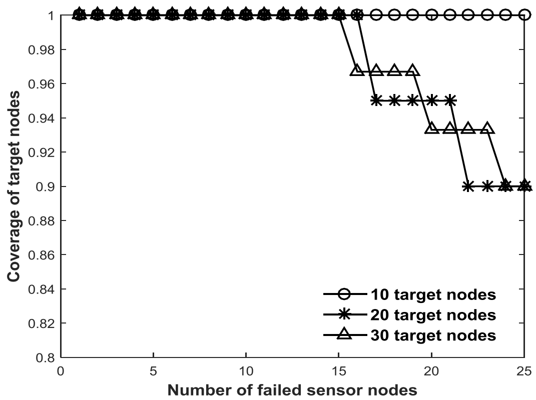

Fig. 7. Coverage rate of target nodes varies with the number of failed nodes.

In Fig. 7, when the number of failed nodes is less than 15%, coverage rate of target nodes always keeps in 100% in the three kinds of node density. When the number of failed nodes increases to 30%, the coverage rates are respectively 100%, 90% and 90%, namely, the number of uncover target nodes are 0, 2 and 3. So it can prove that our algorithm can effectively reduce the effect of sensor fault on monitoring of target nodes.

2) Number of message packets and lifecycle of sensor networks

As can be seen in Fig. 8, the number of message packets need to be sent in three algorithms increases with the increasing of number of failed sensors. However, our algorithm requires the minimum number of message packets and algorithm in [12] followed by, and algorithm in [11] requires the maximum number of message packets.

This is because our algorithm proposes a method of sending requests with delay.

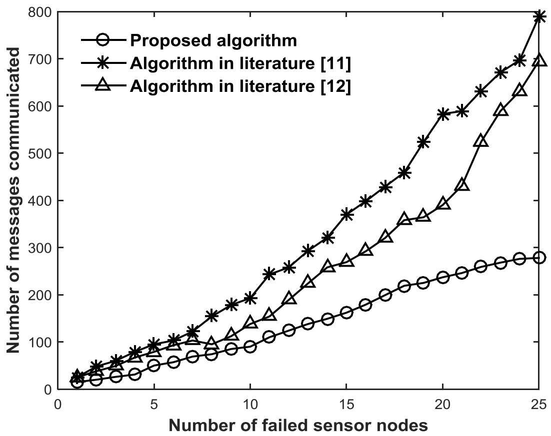

Fig. 8. The number of message packets of the three algorithms varies with the number of failed nodes.

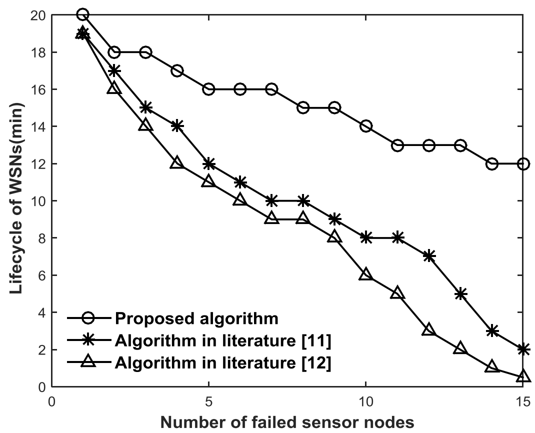

Fig. 9. The lifecycle of the three algorithms varies with the number of failed nodes.

Assume that the network lifecycle ends if it cannot monitor all the target nodes in area or cannot constitute a complete sensor networks. In Fig. 9, we can see that our algorithm has the longest lifecycle, followed by the algorithm in [11], and algorithm in [12] has the shortest lifecycle. This is because when we select alternative nodes, the remaining energy of sensors is considered.

V. Conclusions

Wireless sensors are largely used for monitoring equipment and user status in smart grid. To reduce the effect of sensor fault on monitoring and communication, a fault recovery algorithm for WSNs in smart grid is proposed. The algorithm mainly consists of two steps: fault type judgment and alternative nodes selection. We determine whether we need to select monitor nodes and communication nodes according to different fault types. The alternative nodes are firstly selected from current active nodes. Then we consider whether we need to activate sleeping nodes. Simulation experiments show that this algorithm can realize reliable and rapid sensor fault recovery, and extend the lifecycle of sensor networks with communication data as little as possible.

References

[1] R. M. Huang, X. S. Qiu, and Z. P. Gao, "A neighbor-data analysis method for fault detection in WSNss," *Journal of Beijing University of Posts and Telecommunications*, vol. 34, no. 3, pp. 31-34, June 2011.

[2] H. Redwan, G. Akele, and K. H. Kim, "Cluster-based failure detection and recovery scheme in WSNs," in *6th International Conference on Ubiquitous and Future Networks*, Shanghai, 2014, pp. 407-412.

[3] J. H. Roach and R. J. Marks, "Recovery from sensor failure in an evolving multiobjective swarm," *IEEE Transactions on Systems, man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 170-174, January 2015.

[4] T. Y. Lin, H. A. Santoso, and K. R. Wu, "Global sensor deployment and local coverage-aware recovery schemes for smart environments," *IEEE Transactions on Mobile Computing*, vol. 14, no. 7, pp. 1382-1396, July 2015.

[5] S. Keerthana and M. J. Pamila, "A survey on fault node detection and recovery mechanisms in WSNs," in *Proc. International Conference on Advanced Computing and Communication Systems*, Coimbatore, 2015, pp. 1-5.

[6] Y. F. Hu, X. M. Wu, F. Q. Wang, and H. H. Han, "A particle swarm algorithm based routing recovery method for mobile sink WSNss," in *Proc. 26th Chinese Control and Decision Conference*, Changsha, 2014, pp. 887-892.

[7] J. F. He, J. S. Sun, Y. Zhang, and Z. H. Wang, "Data recovery in WSNss with joint matrix completion and sparsity constraints," *IEEE Communications Letters*, vol. 19, no. 12, pp. 2230-2233, Dec. 2015.

[8] R. M. Huang, "The mechanisms and algorithms of fault management for WSNss," M.S. thesis, State Key Laboratory of Networking and Switching Technology, Computer Science and Technology, BUPT Univ., Beijing China, 2012.

[9] J. Gao, Z. P. Gao, R. M. Huang, and L. M. Meng, "Fault recovery method of WSNss based on density approach," *Chinese Journal of Scientific Instrument*, vol. 33, no. 2, pp. 334-341, Feb. 2012.

[10] S. Chouikhi, I. E. Korbi, Y. G. Doudane, and L. A. Saidane, "Routing-based multi-channel allocation with fault recovery for WSNss," in *Proc. IEEE International Conference on Communications*, London, 2015, pp. 6424-6430.

[11] H. S. Guo and D. Z. Wang, "Recovery method based on fragmented WSNss," *J. Chang chun Inst. Tech. (Nat. Sci. Edi.)*, vol. 12, no. 3, pp. 97-99, 2011.

[12] K. P. Sharma and T. P. Sharma, "CPFR: Coverage preserving failure recovery in WSNss," in *Proc. International Conference on Advances in Computer Engineering and Applications*, Ghaziabad, 2015, pp. 284-289.

Yanfei Lin was born in Shandong Province, China, in 1992. She received the B.S. degree from Beijing University of Posts and Telecommunications, Beijing, in 2014. She is currently pursuing the M.S. degree from the same university, both in computer science and technology. Her research interests include smart grids, network and service management.

Sujie Shao was born in Shanxi Province, China, in 1985. He received the B.S., M.S. and Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, in 2007, 2011, and 2015 respectively, both in computer science and technology. His research interests include network management and smart grid.