BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **16**, No 2
Sofia • 2016 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2016-0019

A Fault-Tolerant Routing Algorithm for Wireless Sensor Networks Based on the Structured Directional de Bruijn Graph

Chuiwei Lu[1], *Defa Hu*[2]

[1]*Computer Institute, Hubei Polytechnic University, Huangshi 435003, Hubei, China*
[2]*School of Information, Hunan University of Commerce, Changsha 410205, Hunan, China*
Emails: Lcwzm@126.com hdf666@163.com

Abstract: *Wireless Sensor Network* (*WSNs*) *nodes with low energy, run out of energy easily and stop working, which results then in routing failures and communication blocking. The paper puts forward a FTRSDDB algorithm based on the structured directional de Bruijn graph to enhance the performance of fault-tolerant routing for WSNs. The algorithm randomly deploys some super nodes with abundant energy and powerful performance in WSNs. These nodes are responsible for the collection of topology information from the WSNs to build redundant routing table, and provide data forwarding and routing update service for popular nodes. The FTRSDDB algorithm optimizes network topology structure using de Bruijn graph, and can quickly find neighbor nodes failure and invalid routing path, and then calculate new routing information with low cost, which greatly improves the performance of fault-tolerant routing of WSNs. Experiments show that the FTRSDDB algorithm takes on better performance compared with other fault-tolerant routing algorithms, even that exist malicious nodes attack in the WSNs.*

Keywords: *Wireless sensor networks, directional de Bruijn graph, fault-tolerant routing.*

1. Introduction

In the past couple of years, Wireless Sensor Networks (WSNs) have been widely used in civil and military fields, such as environmental monitoring, object tracking, forest fire prevention, battlefield targets monitoring and so on. WSNs are mainly made up of miniature sensors of small size and limited resources. The sensors generally rely on battery power, but need to have the battery changed or charging often, which is not feasible. There is also the risk that the sensor nodes are captured by enemy. Once some sensor nodes deplete their energy or are captured by enemy, many local routing paths will be broken. The broken and inaccurate routing will cause the data to be sent to the error destination or will be unable to send out radically, which seriously deteriorates the performance of the data transmission of the WSNs. Because the failures of the sensor nodes are random and uncontrollable, the routing fault occurs frequently in WSNs. To guarantee the rapid and accurate data transmission in WSNs, an appropriate fault-tolerant routing algorithm should be designed to overcome the problem. In fact, the research of the fault-tolerant routing in WSNs has been a hot topic as of late.

There are four main types of fault-tolerant routing algorithms [1-5] for WSNs: the planar fault-tolerant routing algorithm, the data for Centre fault-tolerant routing algorithm, the multilevel fault-tolerant routing algorithm and the multipath fault-tolerant routing algorithm.

Flooding algorithm and Gossiping algorithm are typical representatives of planar structure routing algorithm [6]. The basic idea of Flooding algorithm is to send a copy of the data node to all neighbors. The algorithm is simple, as long as there is a path to the destination, data can be sent, and therefore has a high fault tolerance. But when the network is larger, there will be a broadcast storm. Gossiping algorithm is an improved version based on Flooding algorithm, it uses random transmission mode, it no longer broadcasts data to all neighboring nodes, but transfers data by picking a node randomly in the neighboring node. The method avoids the broadcast storm problem effectively, but brings the problem of high delay and slower speed. Directed Diffusion routing algorithm (abbreviated as DD algorithm) [7] and GRAB algorithm [8] is a typical representative of data-centric fault-tolerant routing algorithm, it uses gradient to represent data forwarding direction the node. But DD algorithm always looks for the path with least delay to transmit data, which will cause these nodes in the path premature death due to energy depletion. The DD algorithm also uses the flooding data strategy, therefore isn’t suitable for large networks. The GRAB algorithm is an improved algorithm of DD algorithm, which uses a directed broadcast strategy to reduce the broadcast storm. The hierarchical fault-tolerant routing algorithm separates all the nodes into two categories based on nodes performance. Two types of nodes are responsible for different functions. The typical representatives of such algorithms include LEACH [9], TEEN [10] and PEGAGIS [11]. LEACH algorithm uses the method of multi-path vote to select the cluster head, so the network is divided into several clusters, it balances the node energy consumption and improves the scalability and robustness of the network, and is very suitable for large scale networks. But LEACH algorithm only fits for homogeneous network, and doesn’t reflect the difference of nodes. TEEN algorithm is an improved algorithm of LEACH algorithm, it uses a hard threshold and soft threshold to adjust data transmission dynamically, and it not only can significantly reduce the number of times of data transmissions, but also can balance energy consumption and rise routing success rate. PEGASIS structure is a chain structure, and the nodes in chain only communicate with its neighbor nodes. Each node in the chain is responsible for data collection, data transfer and data integration. This chain structure reduces the communication distance among nodes; it is more energy-saving and simple, but only fit for small-scale network. Typical examples of Multi-path fault-tolerant routing algorithms are SMR [12] and EAMR

[13]. SMR algorithm builds many transmission paths from the source node to the destination node in network, and obtains ideal fault-tolerance routing through redundant paths, but SMR algorithm use the flooding way to transmit data, which will consume a lot of resources. EAMR algorithm has been improved on SMR algorithm; it divides the data into a finite number of copies on multiple routing paths, which can be helpful in avoiding the flooding problem.

Existing fault-tolerant algorithms have many defects, such as low efficiency, high energy consumption, not suitable for heterogeneous WSNs and so on. Aiming at those defects, we design an improving fault-tolerant routing algorithm based on the improved directional de Bruijn graph. The algorithm is named FTRSDDB, and it uses the directional de Bruijn graph theory to build network topology and redundancy routing table which can effectively guide the super nodes provide data forwarding and routing update service for the popular nodes with low performance. In our algorithm, the super node can quickly find neighbor nodes failure and invalid routing path, and calculate new routing information with low cost, which greatly raises the capability of withstanding fault routing for nodes. Comparing with the congener algorithms, such as Gossiping, DD, and LEACH, the FTRSDDB algorithm demonstrates better fault-tolerant routing performance.

2. Structured directional de Bruijn graph

Many complex network systems can be represented via a graph from the point of view of topology. Using the graph theory, the scholar can carry out better research and improve the network performance using quantitative analysis. The concept of graph theory was always used to evaluate network communication character, especially the static characteristics of the network. Graph is an integrated set made up of several sides and points. Its common expression form is $G(P, E)$, and the *P* denotes the points set which indicates the WSNs nodes. The *E* represents an integer; the side is the connection link between the nodes, and indicates the communication connection between nodes in WSNs. If node *a* and node *b* are the neighbors which have connection in graph *G,* we can use the side: $e\langle a, b\rangle$ to represent connects of the two nodes. There are two types of sides, one is a unilateral side, the other is a bidirectional side. If the side has a start point and an end point, it is a directional side, and the graph is a directional graph.

The number of the sides of a node is named its rank or dimension. The routing path in the graph is a node series, where the distance between node *a* and *b* is the shortest routing path between the two nodes. It can be represented by Dist(*a*, *b*), if Dist(*a*, *b*)=∞， it means *a* can’t reach *b*, and *b* can’t reach *a*. The diameter of the graph is the maximum distance between two nodes, named Max[Dist(a, b)]. The sides set with start side and end side overlap is called a ring or a loop. The perimeter of the graph is its largest length of the loop. In the directed graph, the amount of directed sides reached a certain node is named indegree of the node, and the amount of the directed sides started from a certain node is named outdegree of the node.

The directional graph theory has been widely applied in distributed network system. The de Bruijn graph is a special case of Euler graph. The common expression form of the de Bruijn graph is $G(m,\ n)$; m ($m \geq 2$) represents a kind of mathematics radix, which means the node parameters can be indicated by m number system; n ($n \geq 1$) means the dimension of graph, and means node identification can be expressed as n number of bits. That is any node can be expressed as $x_0 x_l \ldots x_{n-1}$ ($0 \leq x_k < m$, $0 \leq k \leq n - 1$). In the graph, there are m^{n+1} sides, and there are m sides start from $x_0 x_l \ldots x_{n-1}$. The end points of the sides are $x_1 x_2 \ldots x_n$, and there are m sides end with $x_0 x_l \ldots x_{n-1}$, and their start points are $x_n x_0 x_1 \ldots x_{n-2}$.

A standard topology structure of the directional de Bruijn graph $G(2,\ 3)$ is shown as Fig. 1.

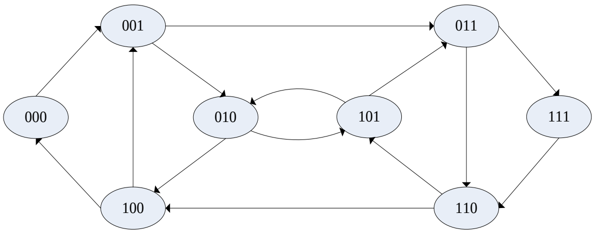

Fig. 1. Topological structure of standard directional de Bruijn graph

The directional de Bruijn graph has excellent characters in mathematical logic:

- Reach any node via less than n hops.
- The number of sides is less than that of nodes, so the total routing message will decrease, which is benefited to raise routing efficiency.
- The proportion of the routing table volume and network diameter is easy to be optimized in the de Bruijn graph.

However, there are a few disadvantages to using the standard directed de Bruijn graph to design WSNs topology structure. An entire directed de Bruijn graph $G(m,\ n)$ contains m^n nodes. But it does not consider the performance difference and geographical distribution of the nodes.

According to the advantages and disadvantages of the standard de Bruijn graph, we propose the concept of structured directional de Bruijn graph which partly improves the de Bruijn graph theory. According to the different performance of the nodes, we separated the nodes into two types: Super node and popular node. The super node will burden much more responsibility on routing forwarding and data transmission, and also responsible for managing the neighbor popular nodes. The popular nodes only provide the basic data acquisition and communication functions, moreover, we separate the WSNs into some subnets based on the geographical distribution and performance of nodes. A subnet is equivalent to a sub-graph of de Bruijn graph. To manage topological structure much more efficiently, we put super nodes and popular nodes into two relative sub-graphs, and formed a kind of two-layer structure de Bruijn graph. For implementing rapid routing between two sub-graphs, we design a main/auxiliary routing mechanism.

For directed de Bruijn graph $G(m, n)$, we sign the first j, $1 \le j \le n$, bits as main routing marking $x_0x_1... x_j$, and the rest sign as auxiliary routing marking $x_{j+1}x_{j+2}...x_n$.

According our improving idea to the standard de Bruijn graph, it has been transformed into a new structure form shown as follow.

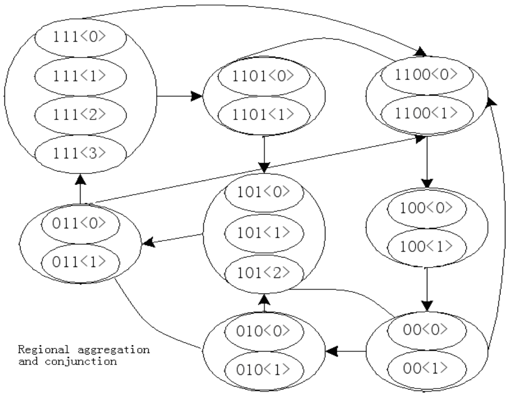

Fig. 2. Topology of directional de Bruijn graph

As is shown in Fig. 2, the improved de Bruijn graph takes on structured, hierarchical, inhomogeneous characteristics, and has been divided into a number of zones with two-layer structures. Those zones vary in size, and have a unique prefix that delegate its organ flag. A number of nodes aggregate a zone based on the principle of short-distance in physical topology. The improving de Bruijn graph well match the practical situations of WSNs, and become the main basis of our optimal strategy in fault-tolerant WSNs routing.

3. Optimal strategy of fault-tolerant routing based on de Bruijn graph

We divide the WSNs into two types of subnets (sub-graphs). A subnet is made up with high performance nodes; the other is made up with low performance nodes. One high performance node manage a subnet, and response for the routing message or data forwarding, so as to raise the capability of fault-tolerant routing. The subnet is strictly structured according to directional de Bruijn graph. The character of topology compactness of directional de Bruijn graph is benefited to maintain the communication relation between two subnets and find the disable nodes quickly, which can highly raise the working efficiency of WSNs.

3.1. Concept and definition

Most networks can be abstracted a directional de Bruijn graph expressed as

$$G = (P, E),$$

P is vertex set, and E is side set, the directional side from node P_i to P_j is expressed as $\vec{P}_{ij}$, and the distance between the two nodes is the shortest path length. For easier analysis and simulation to the algorithm, we propose three definitions.

Definition 1. $G=(P,E)=S(2,j)\times L(2,j)$.

Definition 2. $P=\langle A,\ B\rangle$ denotes a vertex in the directional de Bruijn graph,

$$A=\{a_n a_{n-1}\dots\ a_1\},\ B=\{\ b_m b_{m-1}\dots b_1\},$$

$$\{\ a_n a_{n-1}\dots\ a_1\}\in S\ ,\{\ b_m b_{m-1}\dots b_1\}\in L\times\{a_n a_{n-1}\dots\ a_1\}\ ,$$

and when $\{a_n a_{n-1}\dots\ a_1\}=\{a_n{}'a_{n-1}{}'\dots\ a_1{}'\}$, $\langle\{\ b_m b_{m-1}\dots b_1\},\ \{\ b_m{}'b_{m-1}{}'\dots b_1{}'\}\rangle$ will become one side of sub-graph $L(2,j)$; when $\{b_m b_{m-1}\dots b_1\}=\{\ b_m{}'b_{m-1}{}'\dots b_1{}'\}$, $\langle\{a_n a_{n-1}\dots\ a_1\},\ \{a_n{}'a_{n-1}{}'\dots\ a_1{}'\}\rangle$ will become one side of sub-graph $S(2,j)$.

Definition 3. If the node $u=\{a_n a_{n-1}\dots\ a_1,\ b_m b_{m-1}\dots\ b_1\}\in S$, it will search the neighbor nodes according to the frontal *n* bits of $\{a_n a_{n-1}\dots\ a_1\}$, and find the sub-graph $S(2,j)$ by the graph message provided by the neighbor nodes. If $u=S(2,i)\times\{a_n a_{n-1}\dots a_1,\ b_m b_{m-1}\dots b_1\}\in L$, the node *u* will find the sub-graph $L(2,j)$ according to the frontal *m* bits of $\{\ b_m b_{m-1}\dots b_1\}$. Completing the above searching work, the node *u* will construct its routing table based on the graph message provided by its neighbor nodes.

3.2. Classification of WSNS nodes

The status of the popular node and super node is classified based on some parameters, such as bandwidth, link delay, calculation velocity, geographic position, time of online and so on. The status is also dynamically regulates according to the network circumstance, which means that status of super node and popular node can be exchanged under certain rules. The new joining node is always defined as the popular node, and randomly login neighbor sub-graph $L(2,\ m)$. After running a period of time, the node can be possibly upgraded as a super node base on its performance. The upgraded method refers to the performance (1). At extreme situation, if new node is the first node of the WSNs network, it will become super node naturally, and accepted the following nodes to join it. When the quantity of the joining nodes reaches a certain threshold, a sub-graph $L(2,\ j)$ will be established, but the scale of the sub-graph should be controlled. If the scale is too big, the super node is possibly in danger of overload, single point failure and so on. So it is necessary to set a scale upper limit $S_{\max}$ according to the real situation.

For an accurate method of defining the identity of the WSNs node, we propose a performance function shown as follows:

$$P=\lambda B+\mu D+\omega C+\varepsilon T\ , \tag{1}$$

where parameter *P* denotes node performance, parameter *B* denotes the bandwidth, parameter *D* denotes the network delay, parameter *C* denotes the calculation capability of the node, parameter *T* denotes the online time of the node. As all these parameters had different units of measurement, they can’t be weighted average. We should transform these parameters into one common measurement units according to the same rule. So (1) can calculate performance score of every node. Parameters

$\lambda, \omega, \varepsilon, \mu$ are constant parameters, and $\lambda+\mu+\omega+\varepsilon=1$, these constants are set according to the practical requirement. When performance score reaches certain threshold, popular node will upgrade to a super node, and vice versa, less than certain threshold, the super node will degrade to a popular node.

Each node periodically calls Equation (1) to calculate its real-time performance score by which to adjust node's identity. When sub-graph $L(2, j)$ emerges many super nodes according to (1), if the quantity of super nodes does not exceed certain threshold, these nodes will work as spare super node. When the super node is disabled for some reason, a spare super node will be selected to replace it.

3.3. Function of the two types of WSNs nodes

The popular nodes are the majority members in the WSNs network, and also the main storage region of the network resource, most routing information and data transmission is processing among the popular nodes.

A super node and many popular nodes with a few spare super nodes compose a relatively independent group: sub-graph $L(2, j)$. The super node is the manager of the sub-graph, and also the member of the $S(2, i)$. It has following function.

- Accept the joining application of the new node, and control the scale of the sub-graph $L(2, j)$
- Maintain the nodes identity information, routing information, resource information of the sub-graph $L(2, j)$, and backup those information to the spare super nodes in regular time.
- Update the super nodes information in the $S(2, i)$, master the latest and full-scale information of routing and resource distribution.
- Transfer the latest routing information to the popular nodes in the sub-graph $L(2, j)$, and assist data searching and data transmission in the $L(2, j)$.
- When super node burdens too heavy loads, it is possible to overload, sub-graph $L(2, j)$ schedule spare super node of $L(2, j)$ to share part of burden.

To divide flat structured WSNs network into two-layer architecture proves useful for improving the performance of fault-tolerance routing and load balance of WSNs. In this paper, we introduce the fault tolerance on the basis of the structure partition mentioned above. The super node masters full-scale routing information and resource distribution information. The popular nodes own lower performance, and its resource is also fewer. So our algorithm makes the super node server for the popular nodes. It is a good way to schedule super node's surplus capability to help low performance nodes. The way balances the performance different among the all nodes, and raise the integrate performance of the WSNs.

When a number of super nodes withdraw or failure in the same time will bring about devastating results to WSNs, so we propose the concept of spare super node to solve this problem: spare super node are consisting of the nodes whose performance has reached the standard super node. The current super node needs to periodically back up information on these spare super nodes. Once the current super node withdraws or fails, the WSNs will quickly elect a spare super node to replace it. Because these spare super nodes have already owned the whole backup information of sub graph $L(2, j)$, the replacement process only need few cost, which is benefit to maintain the performance of the routing fault-tolerance.

4. Structural design for WSNs based on directional de Bruijn graph

To improve the routing fault-tolerance performance of WSNs, the most effective method is to divide WSNs into two-layer structure according to structured directional de Bruijn graph theory. One layer was made up of high performance nodes; the other layer is made up of low performance nodes, and separated into a number of sub-graphs. All the low performance nodes are uniformly allotted to these sub-graphs. A high performance node manages a sub-graph, and provides the routing message or the data forwarding service to the sub graph. These methods are benefited to raise the capability of routing fault-tolerance of WSNs.

4.1. Abstraction of WSNs structure

According to Definition 1, WSNs is abstracted to a structural directional de Bruijin graph: $G = (P, E)$. Based on the graph G, we continue to divide directional graph G into two sub-graphs which use binary tag to flag every node. $L(2, j)$ is a sub-graph made up with popular nodes, $S(2, i)$ is a sub-graph made up with super nodes(with high performance). The two sub-graphs construct a close-relative topological structure, so as to use super nodes to raise the performance of fault-tolerant routing. The number of popular nodes managed by a super node should be limited. The FTRSDDB algorithm limits $1 \leq j–i \leq 8$, which means that a super node can service no more than 2^8=256 popular nodes.

4.2. Node classification and function definition

Using two-layer network architecture, the super nodes plays a vital role. But the risk is also increasing significantly for super node, such as overloaded, fixed-point attack and single point failure. It is necessary to design special measures to resolve it. The popular node belongs to a sub-graph $L(2, j)$, but it still belongs to the WSNs, so every popular node is permitted to communication with the nodes of other sub-graph. When popular nodes can’t search data or encounter routing error, they can ask super node for helping to provide route information or data searching. So the burden of the super node is lighten a lot, and the scale upper limit S_{max} is raised. In fact, the good performance of fault-tolerance routing of FTRSDDB algorithm is based mainly on the above mentioned ideal. According to our algorithm, the WSNs topology is formatted as Fig. 3.

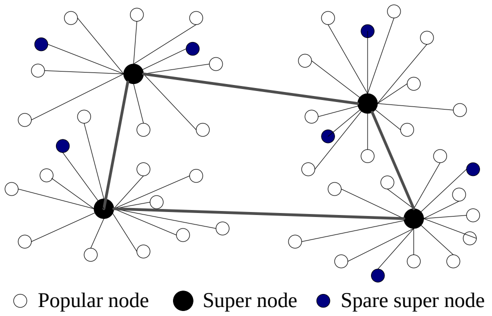

Fig. 3. Two-layer WSNs topology

5. Implementation of the FTRSDDB algorithm

The FTRSDDB algorithm has been designed into three sub-algorithms: the algorithm of node joining and sub-graph building, the algorithm of analysis and forwarding of routing information, the algorithm of node state discrimination and routing rebuilding. Three sub-algorithms complement each other to achieve our FTRSDDB algorithm.

5.1. Algorithm of node joining and sub-graph building

Definition 4. The newly joining node is P_0, the super node is P_s; P_{near} is the neighbor nodes of P_0, P_{0IP} is the IP address of P_0, P_{0port} is the port number of P_0, and the *m* is self-definite integer. The algorithm contents are illustrated as follows.

Step 1. The node P_0 calls hash algorithm to generate a globally unique identifier $P_{0id.}$

Step 2. Identify sub-graph position of P_0 and search the node that matches front *m*-bit binary string of P_{0id} as a neighbor node, i.e., P_{near}.

Step 3. P_0 gets the information of the adjacent super node P_s from P_{near}.

Step 4. P_0 sends a joining request MSG(P_{0id}, P_{0IP}, P_{0port}, New_Join) to P_s. The parameter New_Join is on behalf of the joining message of a new node.

Step 5. If the scale of the current sub-graph is below the threshold S_{max}, the super node P_s accepts the joining request of P_0, and return the message MSG(Popular, Position, Nb$_{u,v}$(P_s)) to the P_0, which indicates the registration was successful. Otherwise, the super node refuses the joining request, and returns the message MSG(Full, Failure) which denotes refusing request.

Step 6. If P_0 receives the refusing message, the P_0 jumps back to Step 1 and restart. If P_0 receive the accepting message, the P_0 initializes its routing table based on the parameters including in the joining request message, and try to send the message MSG(P_{0id}, P_{0IP}, Joined) to the node in the set of Nb$_{u,v}$(P_s).

Step 7. Any active node in sub-graph $L\ (2,\ j)$, such as P_c, when it receives the message from P_0, needs to send a response message MSG(P_{cid}, P_{cIP}, P_{cPort}, NB$_{u,v}$(P_c)) to $P_{0.}$ There is $(S_x \le u \le L_y,\ S_n \le v \le L_m)$. After P_0 has removed the duplicated items and done clustering for $\mathrm{Nb}_{u,v}(P_s)$, it updates the neighbor nodes table, and continue to communicate with the nodes in the table.

Step 8. When P_0 has exchanged the routing and topology information with the entire active nodes in the sub-graph $L\ (2, j)$, its initialization work is over. The future work is to periodically communicate with these nodes to collect the latest network topology changes by which to do routing adjustments for P_0.

5.2. Algorithm of analysis and forwarding of routing information

That routing messages are quickly and accurately transferred in the WSNS is an important evaluation basis of fault-tolerant routing performance. The FTRSDDB algorithm can make full use of the advantages of two-layer topology architecture to achieve fast and accurate routing information forwarding. The algorithm of specific content is shown as follows.

Step 1. When the node Pa = $\langle \mathrm{Pa}_x, \mathrm{Pa}_y \rangle$ sends a message to the destination node Pc = $\langle \mathrm{Pc}_x, \mathrm{Pc}_y \rangle$, it may be required a series of relaying nodes to forward the message. The node Pa generates a relaying nodes set (Transmit_Set), and uses the following steps to select next hop node to expand the set.

Step 2. If $\mathrm{Pa}_x = \mathrm{Pc}_x$ and $\mathrm{Pa}_y \neq \mathrm{Pc}_y$, selects the next hop in the sub-graph $L(2, j)|\ \mathrm{Pa}_x$. The selection method is to match the front *m*-bit hash value of the node identifier. That is to select the nodes which at least match *m*-bit with P_a identifier as candidate next hop. If the candidate node is a spare super node, selects it as the next hop in prior. The *m* is a self-defined integer.

Step 3. If $\mathrm{Pa}_x \neq \mathrm{Pc}_x$ and $\mathrm{Pa}_y = \mathrm{Pc}_y$, selects the next hop in the sub-graph $L(2, j)|\ \mathrm{Pa}_y$, matching approach is identical with Step 2, but the matching goal is P_c.

Step 4. If $\mathrm{Pa}_x \neq \mathrm{Pc}_x$ and $\mathrm{Pa}_y = \mathrm{Pc}_y$, select the super node in the sub-graph $L(2, i)|$ Pa as the next hop.

Step 5. The nodes included in Transmit_Set are suitable as a next hop, the priority to select the next hop in Steps 2, 3, 4 is followed by lower. Pa selects high priority node as next hop and forward information. If the node becomes unavailable for some reason, select the following node as the next hop, and so on.

Step 6. If all the nodes in Transmit_Set are unavailable, to reduce the choice threshold of the next hop. According to the approach illustrated in Step 2, the node which is less one bit matching with the Pa, such as the Pc, is joined into Transmit_Set. That is MatchBits(Pa, Pc) = $m - 1$. Then the algorithm jumps to Step 5 to continue the implementation. We will set the maximum routing hop *n*. If the hops overtake *n*, the routing is failed, and the node Pa will selects the next node in the Transmit_Set and jump to execute Step 5.

5.3. Algorithm of node state discrimination and routing rebuilding

When a sensor node exhausts its battery energy or is captured by enemy, it will inevitably affect the routing accuracy of the WSNs. Before the node reports a failure, it should inform all the relevant nodes to remove the associated routing information as soon as possible. In addition, if a node cannot contact with its relevant nodes in a certain period of time, it should consider the nodes failed. The impact of node capture is worse than node failure due to battery depletion. The captured node possibly reveals secret and diffuses false information of routing or data, which will bring about seriously destruction to data transmission. To simplify the algorithm and save resources, in the algorithm, we do the identical treatment to failure node and captured node. The algorithm is described below.

Step 1. Before the popular node Pa withdraws the WSNs owing to battery exhaustion, it must send a exit message MSG(Pa_{ID}, EXIT) to the corresponding super node in the sub-graph $S(2, i)$. The super node recycles the ID that assigned to the Pa, and notifies the active nodes in the sub-graph $L(2, j)$ to update their routing table.

Step 2. The nodes in the sub-graph $L(2, j)$ uses periodic messages MSG(PING) to detect the survival status with each other, once finds that a popular node fails to contact, the event is immediately reported to the corresponding super node. The super node recycles the ID that assigned to the node, and notifies the active nodes in the sub-graph $L(2, j)$ to update their routing table.

Step 3. When the super node withdraws the WSNs for some reasons, it must send the exit message MSG(Ps_{ID}, EXIT) to its neighbor super nodes in the sub-graph $S(2, i)$. The neighbor nodes immediately remove the information of the super node from their routing table and update the routing information, and then send MSG(Ps_{ID}, Super, EXIT) to all the nodes in $L(2, j)$ sub-graph. In the meantime, a spare super node will become the new super node through competition. The new super node will take over the management power of the quondam sub-graph $L(2, j)$. If the new super node cannot be produced in a certain time, the popular nodes in the original sub-graph will dismiss and try to join other neighbor sub-graph $L(2, j)$.

Step 4. If the node in the sub-graph $S(2, i)$ and $L(2, j)$ can't get in touch with its super node within the certain time, it will consider that the current super node is failed, and then jump to Step 3 to search suitable super node.

6. Simulation experiments analysis

Using network simulation software, we have done a performance comparison among FTRSDDB algorithm and other classical fault-tolerant routing algorithms, such as Gossiping, DD and LEACH. Performance parameters of comparison include transmission delay and transmission success rate of data packet under a variety of working conditions.

We use Opnet modeler 10.0 to do simulation experiments. The experiment circumstance is Windows XP, Intel i5 CPU with 3.2 GHz and 4G memory. The total quantities of WSNs nodes are 10^5, in which there are 200 super nodes. All nodes uniformly distribute in a planar region within 500 × 500 m, and the sink node is in the center of region. The popular nodes generate one data packet (contain timestamp information) average per second, and transmit them to sink node. The sink node calculates the quantities of received data packets, transmission delay and transmission success rates and etc. The below are the experiment results.

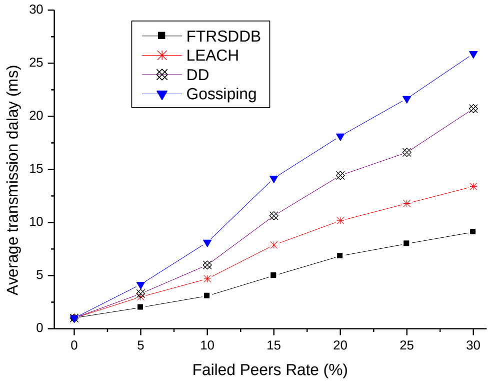

Fig. 4. Transmission delay (node death rate increase from 0 to 30%)

From Fig. 4, we can see that the data transmission delay of all algorithms are increasing quickly with part of the node gradually die due to battery exhaustion. The transmission delay of Gossiping and DD algorithm are increasing more fast, which is related to their data flooding mode. The transmission delay of LEACH algorithm is a little low, which is benefited to its hierarchy network structure, but its delay performance is still inferior to our FTRSDDB algorithm.

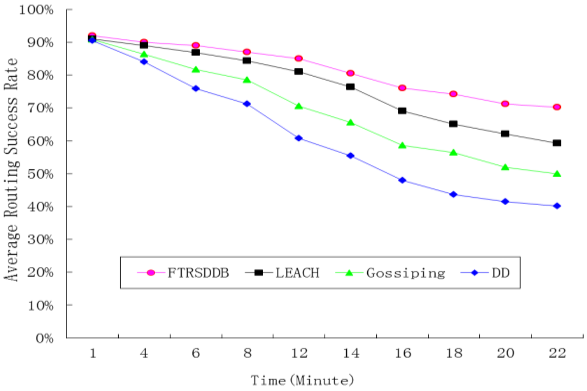

Fig. 5. Routing success rate (node death rate increase from 0 to 30%)

From Fig. 5, we can see that routing success rate of FTRSDDB algorithm is better than the three algorithms, and its fault-tolerant routing performance is also best among the four algorithms. Network topology constructed by the directional de Bruijn Graph can quickly and accurately perceives the status changes of wireless sensor nodes, which is conducive to quickly update routing table of nodes. So the FTRSDDB algorithm can respond quickly to the abnormal nodes and adjust the routing path for normal nodes in time. These excellent characters are the main factors that contribute to FTRSDDB algorithm obtain the best fault-tolerant routing performance among the above algorithms.

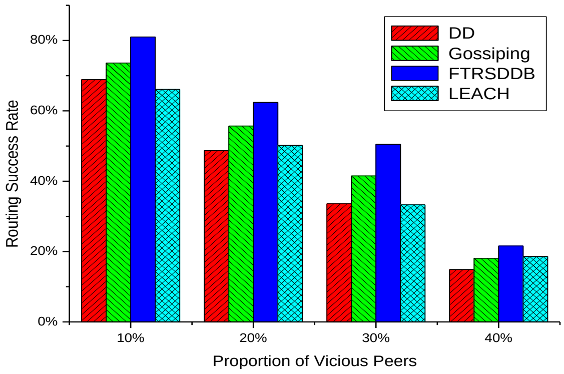

Fig. 6. Routing success rate (malicious node proportion is respectively equal to 10, 20, 30, 40%)

The malicious nodes in the WSNs can diffuse error information to destroy the network routing, which is an important challenge to all fault-tolerant routing algorithms. FTRSDDB algorithm can analyze the error information and exclude most of them to improve performance of fault-tolerant routing. It is the unique features that other algorithms don't own. As can be seen from Fig. 6, along with the number of malicious nodes increasing, the routing success rate of above algorithms are all rapidly declining, which indicates that the number of malicious nodes have significant impact on routing performance. Routing success rate of the FTRSDDB algorithm is much higher than other three fault-tolerant routing algorithms. It indicates that the algorithm with hierarchical structure is more easily obtain better performance in fault tolerance routing.

7. Conclusion

This paper has researched the routing failure problem that exists in the WSNs, and proposed an optimized fault-tolerant Routing algorithm named FTRSDDB for WSNs. The algorithm describes the WSNs topology based on the structured directional de Bruijn Graph, which optimizes the network structure and improves the speed and accuracy of analysis to routing failure problems. The FTRSDDB also utilizes super nodes to balance the energy consumption and data transmission for WSNs nodes. Even some nodes fail due to battery depletion; the FTRSDDB can still get better performance in fault-tolerant routing. The simulation experiments have demonstrated that even under the condition of network attacks launched by some malicious nodes, comparing with other common fault-tolerant routing algorithms, the FTRSDDB algorithm still owns better performance score.

***Acknowledgements*:** The work is supported by the Key Project of Hubei Provincial Department of Education (No D20144403), the Outstanding Youth Science and Technology Innovation Team Project of Hubei Polytechnic University (13xtz10), Innovative Talent Project of Hubei Polytechnic University (13xjz06C), the National Natural Science Foundation of China (Grant No 61202464)

References

1. Peng, Y. H., Q. Y. Song, Y. Yu, F. Wang. Fault-Tolerant Routing Mechanism Based on Network Coding in Wireless Mesh Networks. –Journal of Network and Computer Applications, Vol. **37**, 2014, No 1, pp. 259-272.
2. D' Innocenzo, A., M. D. Di Benedetto, E. Serra. Fault Tolerant Control of Multi-Hop Control Networks. – IEEE Transactions on Automatic Control, Vol. **58**, 2013, No 6, pp. 1377-1389.
3. Li, H. B., Q. Y. Xiong, W. R. Shi. Mechanism of Immune System Based Multipath Fault Tolerant Routing Algorithm for Wireless Sensor Networks. – International Journal of Distributed Sensor Networks, Vol. **11**, 2013, No 5, pp. 77-85.
4. Nabizadeh, H., M. Abbaspour. IFRP: An Intrusion/Fault Tolerant Routing Protocol for Increasing Resiliency and Reliability in Wireless Sensor Networks. – International Journal of Ad Hoc and Ubiquitous Computing, Vol. **14**, 2013, No 1, pp. 52-69.
5. Li, H. B., P. Gao, Q. Y. Xiong. A Vascular-Network-Based Nonuniform Hierarchical Fault Tolerant Routing Algorithm for Wireless Sensor Networks. – International Journal of Distributed Sensor Networks, Vol. **10**, 2012, No 6, pp. 64-73.
6. Hedemiemi, S., A. Liestman A Survey of Gossiping and Broadcasting in Communication Networks. – Networks, Vol. **18**, 1988, No 4, pp. 319-349.
7. Intanagonwiwat, C., R. Govindan, D. Estrin. Directed Diffusion. A Scalable and Robust Communication Paradigm for Sensor Networks. – In: Proc. of MobiCom, June 2000, pp. 56-57.
8. Ye, F., S. Lu, L. Zhang. GRAdient Broadcast: A Robust. – Large Sensor Network, Vol. **10**, 2001, No 7, pp. 78-82.
9. Heinzelman, W. R., A. Chandrakasan, H. Balakrishman. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. – In: Proc. of 33rd Annual Conference System Sciences on Hawaii, October 2000, pp. 66-75.
10. Manjeshwar, A., D. P. Agrawal. APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks. – In: Proc. of 15th Parallel and Distributed Processing Symposium, March 2001, pp. 2009-2015.
11. Lindsey, S., C. Raghavendra. PEGASIS: Power-Efficient Gathering in Sensor Information System. – In: Proc. of IEEE Aerospace Conference, July 2002, pp. 1125-1130.
12. Lee, S. J., M. Gerla. Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks. – In: Proc. of IEEE ICC, February 2001, pp. 49-58.
13. Shah, R. C., J. M. Babaey. Energy Aware Routing for Low Energy Ad Hoc Sensor Networks. – In: Proc. of Wireless Communications and Networking, November 2002, pp. 350-355.