A Self-adaptive Fault-Tolerant Mechanism in Wireless Sensor Networks

Wei Xiao[1,2*], Ming Xu[1], Yingwen Chen[1]

[1]School of Computer, National University of Defense Technology, Chang sha, China
[2]Hunan Normal University, Chang sha, China
{xiaowei}@hunnu.edu.cn

Abstract. This work was motivated by the idea of getting admirable fault-tolerant and efficient performance in application when smart sensors are engaged in 'in-network' computing in wireless sensor networks. In such applications, it is crucial for the sensors to form an optimal network topology and tune to transmission attempt rates in a way that optimize network throughput. However, we cannot ignore the influence from the fault node occurring in the network when the optimal network topology is constructed. In this paper, we proposed a self-adaptive way which identifies the node's confidence rate in accordance with its fault possibility, which in turn figures out the weighted volume between any two adjacent nodes. Moreover, we discussed and tried to solve the problems on FTMAWSS (fault-tolerant maximum average-weighted spanning subgraph) in weighted connected graph initiating from clustering WSNs. Simulation results confirmed the validity of the proposed algorithm with a high degree of accuracy and demonstrated that our proposed way could be scaled to large networks.

Keywords: Fault-tolerant, self-adaptive, sensor networks.

1 Introduction

The potential range of WSNs applications cover a large number of domains from physiological, habitat and environmental monitoring, condition-based maintenance, smart spaces, military, precision agriculture, transportation and inventory tracking [1,2]. Because sensing devices now combine the functions including sensing, computing, and wireless communication, so this smart sensor may have only modest computing power but the ability to communicate will allow sensors to organize themselves into a network and collaborate to execute tasks more complex than just sensing and forwarding the information. This networks environment is defined as 'in-network' computing.

As we known, a global computation proceeds in step comprising of certain local computations at each sensor. Thus, the faster the sensors complete their share of local computations the higher will the global computation rate be. The more frequently

This research was supported by the Hunan Province Natural Science Foundation of China under the grant NO.06JJ50107.

exchange of results among neighboring sensors can occur, the more rapidly the progress of the computation may run. Therefore, when the computing rate of nodes is fixed, the packet transport rate is critical capability for the preferable ability of holistic networks computing in sensor networks [6]. And also in [6], it gives a way to optimize communication throughput and form an optimal network subgraph, which express best certain performance (computing rate or delay) besides the additional complexity. However, the decreasing of practical value in given subgraph for the influence of fault node cannot be considered.

Nodes in WSNs are prone to failure due to energy depletion, hardware failure, communication link errors, malicious attack, and so on. So it is essential to provide fault tolerant techniques for distributed wireless sensor networks. Moreover, there are some desired ways to work out faults in the WSNs. One important approach for dealing with the inherent lack of structure in sensor networks has been dominating set based clustering [3,4]. In this paper, we consider only 'soft fault' which may occur in a system on either a permanent or a transient basis. The fault nodes could be detected with special algorithm using transmitted values at sink or aggregated node. Then we could assign nodes with confident rate (CR) respectively and modify transport attempt probability (TAP) at nodes dynamically. So the self-adaptive of sensor networks can be used to construct an optimal network subgraph with fault tolerant in this paper in order to form an effective, practical subgraph topology.

The remainder of the paper is organized as follows. Section 2 provides briefly some relation work, and then gives our difference work. Section 3 the two main aspects in my work are proposed respectively and the algorithms are given. Section 4 presents the simulation experiments and some analysis. Finally section 5 concludes this study and future directions.

2. Related Works

On the one hand, in recent work, many self-adaptive algorithms and protocols have been developed in ad hoc and sensor networks. Its various aspects, including topology discovery and control, scheduling, localization and energy efficiency, have been addressed in the papers. For example, topology formation and scheduling for communication networks have been discussed in [5].The optimal self-organization topology building , include centralized as well as distributed algorithms, is give in ad hoc wireless sensor networks in [6]. [7] Presents a distributed evolutionary algorithm for reorganizing network communication and a message efficient clustering algorithm for sensor networks. Moreover, various self-adaptive procedures are described in [8].On the other hand; in-network aggregation approaches, such as gossip and hash-based algorithms are used to minimize the energy spent in the aggregation operation. Hybrid algorithms have been proposed that methodically try to optimize the use of efficient tree aggregation in [9].Dasgupta [10] presents a tree-based aggregation algorithm. Heidemann et al. [11] proposes a directed diffusion mechanism in which a user's queries are forwarded to an application aware sensor node based on a least-cost algorithm.

Our work differs from the previous work in the following aspects. Firstly, different from the normal aggregation approaches in fusion nodes, the infections of fault values from sensor to the final products will be considered in our works. We can alleviate fault sensor effect to final answer in fusion nodes furthest, by means of reducing confidence rate of fault nodes step by step. Secondly, the better networks throughput topology in networks will be studied all together. We achieve a FTMAWSS (fault-tolerant maximum average-weighted spanning subgraph) in the connected graph, which is modified from MAWSS (maximum average-weighted spanning subgraph) in [6].

3. Self-adaptive Fault-tolerant and optimal throughput topology

We assume that a large number of static sensor nodes are placed in a region. These sensors are not only engaged in collecting-and-delivering task but simple integrating data job. Nevertheless, the uppermost values aggregation is completed in fusion nodes, which have strong computing and transmitting power. Fig.1. shows a traffic model for a sensor node in our computing networks. A simple integrate algorithm running on each sensor uses local values and data packet from the other neighbor sensors. And the data packet to be sent to the neighbors is queued up in a packet queue. The nodes within the transmission range of a sensor are designated as its neighbors.

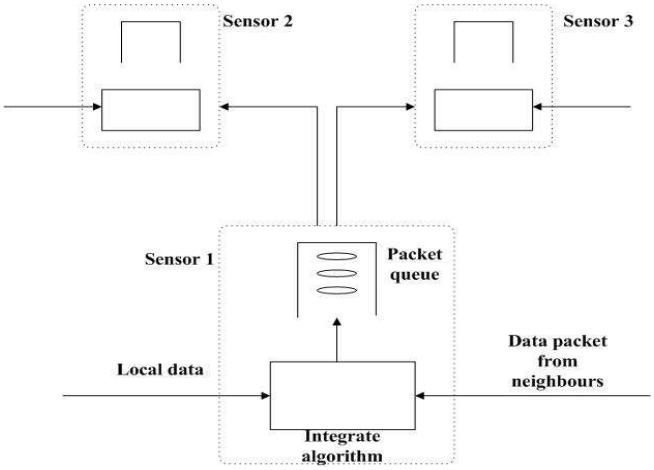

Fig.1. a traffic model for sensors carrying out values and integrate computation

3.1 preliminaries

Let $G(V, E)$ denote a connected weighted graph with vertex set V ($|V| = N$), edge set E and weight function $\psi : E \to R^+$.The weight value of an edge $(i, j) \in E$ is denoted by $\psi(i, j)$. Edge weight values generally include link length, skip numbers, and delay times .etc. In our work, the edge weight will be given in following section. We assume all sensors transmit using a fixed transmit power. Without loss of generality, in this paper we assume G is clustering with a fixed sink node and every sensor nodes in G can sense the same event happen in the deploying field. The fusion node has enough computing power and transport ability to achieve data aggregating and information diffusion. A sensor cannot transmit and receive simultaneously.
We assume that time is slotted and channel access is random. In each slot, sensor i decides to transmit packet with probability α_i and decides to receive with probability $(1-\alpha_i)$; α_i is called the ***transport attempt probability (TAP)*** of sensor i, which can be defined as data packet transport throughput and modified based on environment or node state change. When a sensor' TAP is higher, the more data packets are conveyed to other nodes. The more frequently exchanges of results among neighboring sensors can occur, the more rapidly will the overall computation converge. The more rapid the progress of the computation, the more fast the variations of a spatio-temporal process that can be tracked. Thus, we could search a fine way to construct the optimal communication throughput subgraph.

On the other hand, when the sink or aggregated node encounters fault data from fault sensor nodes, detecting it and throwing its impact on the aggregative values in fusion nodes is best choice. So we assign ***confidence rate (CR)*** to each sensor nodes based on deviation between their values with mean values, decrease fault effect on fusion datum, i.e., sensor i will be assigned c_i as confidence rate. Moreover, we give a relational function between CR and TAP, and use it to tie with these two main parameters in my task.

3.2 modified self-adaptive fault-tolerant method

In this section, we assume that the sensor graph is multipath routing and can be used with DOI (Duplicate and Order Insensitive) data structure for duplicate elimination [12]. After the value from sensor sensing event and the packet from neighbor are sent to the sink or aggregated node, the detection of data failure and the CR assignment are processed quickly at the sink. In this paper, fusion sensor includes the sink or aggregated node.

Firstly we initialize set $c_i = \frac{1}{N}$ and then a weighted averaging process is done at the fusion sensor. The average value is given using formula (1).

$$avg = \frac{\sum_{i=1}^{N} c_i}{N} \quad \textbf{(1)}$$

Where c_i is confidence rate of sensor i which is assigned by fusion sensor, d_i is value from sensor i sense in circumstance. N is the number of nodes in G.

Afterward the maximum distance of the value to the average is computed. Then, this value is used to compute dissimilarity of each node value from the average, see (2). Formula (3) typically used to find distance of two objects [13].

$$m = \max(\{|avg - d_i|, \forall i \in V\}) \quad \textbf{(2)}$$

$$D(d_i, avg) = \cos([1 - \frac{|avg - d_i|}{m}] \times \frac{\pi}{2}) \quad \textbf{(3)}$$

The current confidence rate value to each node has an inverse relation to the dissimilarity of node to the average. In other words, if dissimilarity is large, confidence is low and if it is low, the confidence is large. We use (4) and (5) to get current confidence to each node.

$$c_{i-cur} = 1 - D(d_i, avg) \quad \textbf{(4)}$$

$$c_i = \gamma \times c_{i-old} + (1-\gamma) \times c_{i-cur} \quad \gamma \in [0,1) \quad \textbf{(5)}$$

Where c_i is final confidence rate at this round of decision making process is, γ is an empirical parameter. Through this process, nodes that are faulty get less and less weighted so their values get less effective. Fusion sensor assigns confidences rate to detectors; after each round of decision making process, fusion sensor reevaluates these confidences. The confidence rate to sensor will be increased if it has large correlation with final answer in this round and vice versa. As a result of this adaptation process, large weights are assigned to sensors which their history of decisions shows more correlation to the decisions of fusion sensor. Besides above action, the other effect related with transport attempt probability is given in following section.[13]

3.3 The relational function between transport attempt probability and confidence rate

The relationship between transport attempt probability α_i and confidence rate c_i will be considered in this paper, it is key factor in our proposed tasks. The objective we can achieve higher throughput and best fault-tolerant capacity will be come true, if

we find the novel relationship between α_i and c_i. The following formula (6) is given as our selective function.

$$\alpha_i = \begin{cases} \varepsilon_{upper}, & if \quad c_i \geq \varepsilon_{upper}; \\ c_i^2, & if \quad \varepsilon_{lower} \leq c_i < \varepsilon_{upper}; \\ 0, & otherwise. \end{cases} \tag{6}$$

Where ε_{upper} and ε_{lower} are defined as the highest confidence rate and the lowest confidence rate respectively.

In order to calculate transport attempt probability (TAP), bounds on c_i need to be established. We consider that the TAP α_i will increase exponentially with the increase of the CR c_i among bounds. When c_i is under ε_{lower} that mean sensor i is fault with high probability and lost his basic transporting and computing ability with higher probability, so we can assign its TAP is zero. However, when c_i in sensor i exceed ε_{upper} too much, α_i would be assigned much higher fixed value. Nevertheless, the receive rate $(1-\alpha_i)$ is special lower when sensor i is assigned a very higher value, so the small quantity of data packets from neighbors are received. Therefore, we assign a fixed value ε_{upper} as TAP when c_i exceed ε_{upper}. These changes can be seen in fig .2. The best bounds on α_i created from c_i will be investigated carefully in order to gain efficient throughput topology in given sensor networks.

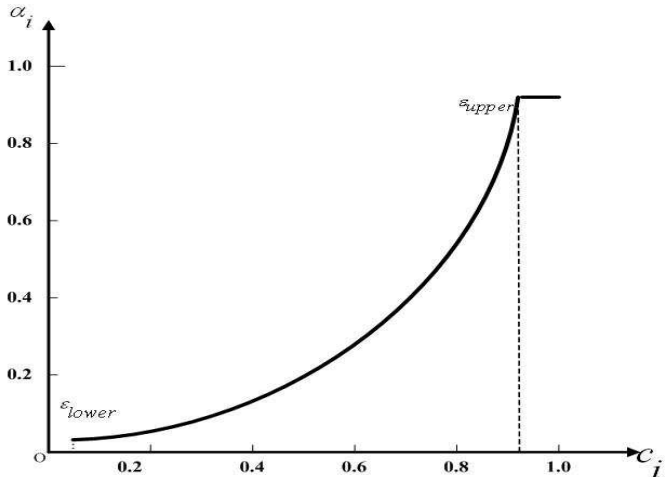

Fig.2. function about confidence rate and transport attempt probability

3.4 The algorithm on constructing the fault-tolerant optimal throughput subgraph

In my applications, the sensors will monitor the environment and collect datum autonomously. Then the sensed values are delivered to sink or fusion sensors. Our work focuses on this paradigm. We consider a random access communication model.

In $G(V,E)$, let $N_i(G)$ is the set of neighbors of i and $n_i(G)$ is the number of neighbors of i in topology G. Let for every $(i,j)\in E$, p_{ij}, formula (7) give, denote the probability of successful transmission from sensor i to j and also can be defined as weight value in edge (i,j). $G(V,E)$ With p_{ij} for every edge (i,j) also can be defined as weighted connected graph.

$$p_{ij}=\begin{cases}0, & if\ \alpha_i=0 \quad or \quad \alpha_j=0;\\ \alpha_i(1-\alpha_j)P((\frac{d_{ij}}{d_0})^{-\eta}\geq\beta), & otherwise.\end{cases} \qquad (7)$$

Where d_{ij} is the distance between i and j, and d_0 is the near-field crossover distance. The last term in (7) is $P((\frac{d_{ij}}{d_0})^{-\eta}\geq\beta)$ where $(\frac{d_{ij}}{d_0})^{-\eta}$ denotes the path loss with exponent η. β is a given threshold.

Let $G'=(V',E')$ denote a subgraph of a given connected graph G. $N_i(G')$ is the set of neighbors of i in topology G'. $n_i(G')$ is the number of neighbors of i in topology G'.

We can give formula (8) for each i. $M_i(G')$ may be defined as the average throughput of sensor i:

$$M_i(G')=\frac{1}{n_i(G')}\sum_{j\in N_i(G')}p_{ij} \qquad (8)$$

$$\xi(G')=\sum_{i\in V'}M_i(G') \qquad (9)$$

Define a function ξ on G' as (9), which give the total throughput in G' and also we can define $\hat{G}=\arg\max_{G'\in Gsub}\xi(G')$ where G_{sub} the set of all connected spanning subgraphs of G. Moreover, G_{sub} is nonempty since $G\in G_{sub}$. Otherwise, we recall that TAP α_i is tuned by CR c_i which is adjusted self-adaptive fault-tolerant method. So we call $\hat{G}$ the fault-tolerant maximum average-weighted spanning subgraph (FTMAWSS). Because maximum average-weighted spanning subgraph

(MAWSS) for directed and undirected graphs is NP-complete [6], FTMAWSS is NP-complete too.

In the following, we will give centralized FTMAWSS algorithm. The basic idea of constructing the following subgraph comes from [6]. Firstly, we let $\max e(i)$ denotes the heaviest outgoing edge for node i, and $W \max e(i)$ denotes its weight. $E \max e(G) = \{\max e(i) | i \in G\}$ is the set of maximum weight outgoing edges of all the nodes in G. It is clear that FTMAWSS contains $E \max e(G)$. Hence if $(V, E \max e(G))$ is strongly connected, we finished. Otherwise, we consider the $\hat{\psi}(i, j) = W \max e(i) - \psi(i, j)$, which help us to construct minimum weight out-branching rooted at each i. The minimum weight branch pick out edges with small $\hat{\psi}(i, j)$ which are the edges with large $\psi(i, j)$. The resulting graph is taken as an approximation to the FTMAWSS. Algorithm is following:

Algorithm 1 Algorithm for finding an approximation $\hat{G}$ to the FTMAWSS of the graph G

Update CR

1. get measures from sensors in graph G, compute average and confidence rate for every sensor nodes using formula (1)-(5).
2. send to all sensors with their updated CR(confidence rate) separately.
3. all sensors determine their TAP(transport attempt probability) separately in accordance with formula (6)

Create FTMAWSS

4. If $(V, E \max e(G))$ is strongly connected then $\hat{G} = (V, E \max e(G))$
5. Else
6. for all $(i, j) \in E$, $\hat{\psi}(i, j) = W \max e(i) - \psi(i, j)$ and set $G_{\min} = (V, E, \hat{\psi})$
7. For all $i \in V$, find the minimum weight out-branching of $G_{\min}$ rooted at i, be defined as E^i_{out}.
8. $\hat{G} = (V, \bigcup_{i \in V} E^i_{out})$

We can use this algorithm to create fault-tolerant subgraph from a connected or strongly connected graph. Fig.3. shows topology structure till 300 slots with 100 random sensors deployed in specify area with no fault sensors. Fig.4. shows FTMAWSS constructed from Fig.3 with some fault sensors. In Fig.4, the red dots are the fault nodes, which turn to isolated nodes in FTMAWSS.

Fig.3. Topology discovered till 300 slots by 100 sensors with no fault node.

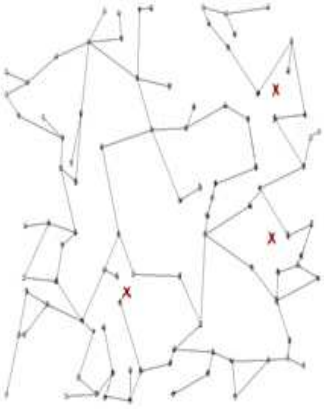

Fig.4. shows FTMAWSS constructed from Fig.3 with three fault sensors.

4. Simulation Results

In this section we provide our simulation results to evaluate its performance using the NS-2 simulator [15]. In this set of simulations, nodes are randomly distributed in a 1000m × 1000m region. We have chose scenarios which consist of several sensors field of different population sizes ranging from 100 to 500 sensor nodes. In the simulation, the propagation model is the two ray ground model, the MAC protocol is IEEE 802.11, and the routing protocol is AODV. The broadcast data packets were sent using CSMA/CA. The sink node will be located on the top of the network. The nodes beside the sink would all be sources in experiment field. Table 1 lists the simulation parameters we have used in our experiments.

Table 1 Simulation parameters

Parameters	value
Simulation time (s)	500
Number of node	100-500
Source data rate (evenMsgs/s)	10
Number of source nodes	100-500
Radio range (m)	20
Transmit energy (mW)	14.88
Receive energy (mW)	10.25
Dissipation in idle (mW)	8.13
Dissipation in sleep (Mw)	0.012

We assume slot synchronization among sensors. We use synopsis diffusion approach and DOI (Duplicate and Order Insensitive) data structure for duplicate elimination [14] to guarantee multi-path routing scheme in networks initially. In each experiment we picked the average of ten runs of our algorithms as the output results. We used ε_{upper} =0.85 and ε_{lower} =0.08 in our experiments and the following metrics to evaluate our mechanism:

- Average delay: average latency from the moment a packet is transmitted to the moment it is received at the sink.
- Average packet loss ratio: number of total packets is lost at nodes to the number of total packets sent at nodes ratio.
- Average effective packet delivery ratio: number of correct packets is received at sink to the number of total packets is received at sink ratio.

Note that, we will also use term FTMAWSS/MAWSS to denote an algorithm for adopting a FTMAWSS/MAWSS topology in the following experiment.

4.1. the effectiveness of FTMAWSS

Firstly, we consider distinction of networks ability in FTMAWSS comparing with result in MAWSS [6]. Node failures are simulated by injecting especial values to a fixed fraction of the nodes simulated. These nodes were randomly chosen from the sensor field and inject fault values at a random time during the simulation.

Average delay is particularly important in WSNs applications, which demand a fast and dependable response. Generally, when network size is increased, the delay gets higher due to the greater number of hops a packet travel from source to sink. In MAWSS, packets will backdate then cause higher delay when meeting drawback node, because MAWSS don't consider isolating fault node in his construction. As shown in the graph of Fig. 5, for a fixed network size, say 400 nodes, the delay of MAWSS with 20% node failures is 90% higher than FTMAWSS with 20% node failures. We also see that the increasing proportion of delay time in FTMAWSS is lower when the number of failed node is increased.

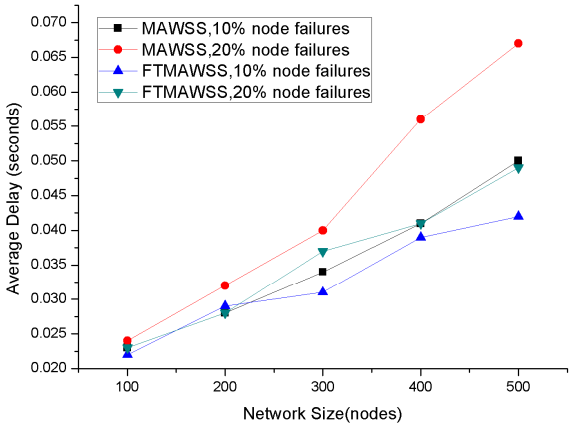

Fig.5. average delay with node failures

Sensor network reliability can be measured by its average effective packet delivery ratio, which reflects the correct rate of packets transmissions to the sink. Fig. 6 shows that FTMAWSS is able to maintain a reasonable packet delivery rate even at a high percentage of node failures. This reason is that the failure nodes are detected and isolated by the algorithm which constructed FTMAWSS. However, the effective packet delivery ratio of MAWSS will get lower when network size is increased. Especially as network size is 500 nodes, the effective packet delivery ratio is no more than 0.82. Packet loss ratio is another parameter for sensor networks to ensure reliability and creditability. With the same reason, as shown in the graph of Fig. 7, the packet loss ratio of MAWSS is higher than that of FTMAWSS at every network size.

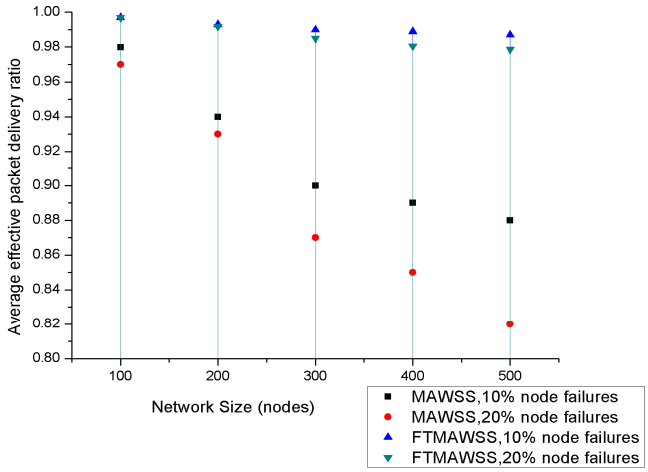

Fig.6. average effective packet delivery ratio with node failures

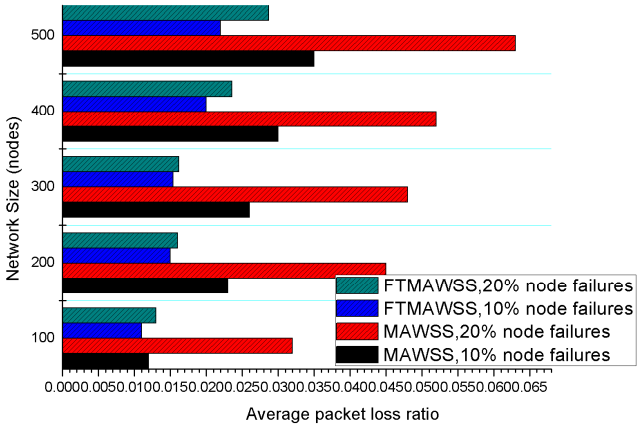

Fig.7. average packet loss ratio with node failures

4.2 the effectiveness of different topologies

In our last set of experiments we add a new topology subgraph building algorithm, which include simply process: in connected or strong connected graph, every node choose two edges connected to its neighbors randomly, till the subgraph is connected or strongly connected. If node has only one neighbor then choose this edge. We call this constructed subgraph topology is RANDOM, certainly which don't be considered with fault tolerant.

Afterward, we fixed the network size, say 500 nodes, and change the fault node percent. Fig.8 shows that FTMAWSS and MAWSS can gain steady average delay till the node failure percent is bigger than 10%. On the other hand, as the node failure is increased, the average delay of RANDOM is raising very quickly. It is clear that the subgraph randomly created can't achieve the whole well transmission capability; despite might choose some edges are the best weighted ones. However, it is surprised for us that the effective packet delivery ratio of RANDOM is higher than that of MAWSS when the failure node percent is bigger than 12%. Fig.9 shows this thing and express the other instance is natural. We think this abnormality might be explained that the probability of packet transmitting through fault nodes is decreasing in RANDOM at high density networks.

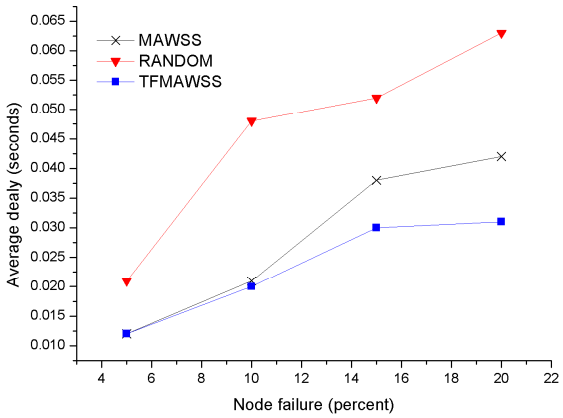

Fig.8. average delay with various topologies

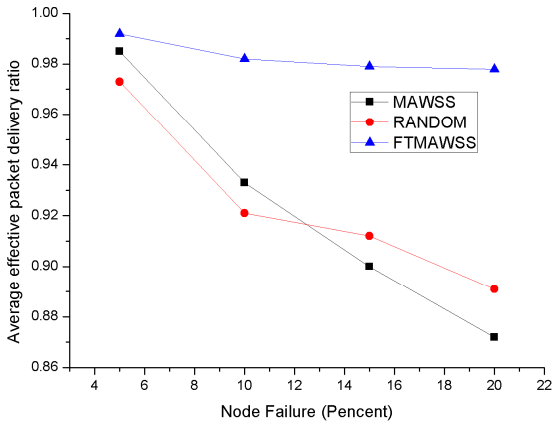

Fig.9. average effective packet delivery ratio with various topologies

5. Conclusion

In this paper, we propose a self-adaptive fault-tolerant mechanism in the sensor networks. In “in-network” computing, the sensor nodes are busy with sensing, computing, and transporting combined packets. We use confidence rate, the sink giving, to ascertain and adjust the node transport attempt rate in order to achieve holistic best network throughput. Simulation experiments confirm the validity of our ways and ensure our algorithm is reliability and scalability.

References

[1] J. Kahn, R. Katz, K. Pister.: Next century challenges: mobile networking for smart dust, In: pro.of ACM/IEEE Intl. conf. on Mobile computing and Networking (Mobicom 99), August 17-19, 1999, Seattle, WA, pp. 271--278.(1999)

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci.: A survey on sensor networks. In: IEEE Communications Magazine, August, pp. 102--114.(2002)

[3] D. J. Baker and A. Ephremides.: The Architectural Organization of a Mobile Radio Networks via a Distributed Algorithms. In: IEEE Transactions on Communications, COM29 (11), pp. 1694—1701. (1981)

[4] F. Kuhn, T. Moscibroda, and R. Wattenhofer.: Initializing Newly Deployed Ad Hoc and Sensor Networks. In: Proceedings of the 10th Int. Conf. on Mobile Computing and Networking (MOBICOM), (2004)

[5] R. Krishnan and D. Starobinski.: Message-efficient self-organization of wireless sensor networks. In: proc. IEEE WCNC2003, pp. 1603--1608.(2003)

[6] A. Karnik, A. Kumar.: Distributed optimal self-organization in ad hoc wireless sensor networks, In: IEEE/ACM transactions on networking, vol. 15, No.5, 2007, pp. 1035--1045.(2007)

[7] M. Post, A. Kershenbaum, and P. Sarachik.: A distributed evolutionary algorithm for reorganizing network communications. In: proc. IEEE MILCOM. (1985)

[8] L. Clare, G. Pottie, and J. Agre.: Self-organizing distributed sensor networks. SPIE-The Int. Sco. for Optical Eng. pp. 229--237, 1999.(1999)

[9] Laukik Chitnis, Alin Dobra, Sanjay Ranka.: Aggregation methods for large-scale sensor networks. In: ACM Transactions on Sensor Networks ,2008,4 (2),pp.1--36.(2008)

[10] K. Kalpakis K. Dasgupta and P. Namjoshi.: An efficient clustering-based heuristic for data gathering and aggregation in sensor networks. In: IEEE. WCNC., 2003, pp.1948--1985.(2003)

[11] J. Heidemann.: Building efficient wireless sensor networks with low-level naming. In: 18th.sym. syst. Princ., 2001, pp. 146--159.(2001)

[12] S. Nath, Phillip B. Gibbons, Zachary R. Anederson.: Synopsis diffusion for robust aggregation in sensor networks. In: 2nd international conference on embedded networked sensor systems, 2004, pp. 250-262.(2004)

[13] A. Abbasi, E. Ghadimi, A. Khonsari.: A distributed clustering algorithm for fault-tolerant event region detection in wireless sensor networks. In: Lecture notes in computer science. springer, berlin. 2007, pp 493-502. (2007)

[14] Considine, J., LI., F., Kollios, G., Byers, J.: Approximate aggregation techniques for sensor databases. In: 20th international conference on data engineering, 2004,pp. 449--460.(2004)

[15] S. McCanne and S. Floyd. ns Network Simulator, http://www.isi.edu/nsnam/ns/.