A trust evaluation method for improving nodes utilization for wireless sensor networks

Shen Haibo*, Zhuang Kechen and Zhang Hong

School of Computer Science and Engineering,

Nanjing University of Science & Technology, Nanjing, China

[Email: hbshen29@163.com, xjzkcgf@126.com, zhhong@mail.njust.edu.cn]

*Corresponding author: Shen Haibo

Received February 14, 2017; revised September 19, 2017; accepted October 21, 2017; published March 31, 2018

Abstract

Existing trust evaluation models for wireless sensor networks can accurately and objectively evaluate trust value of nodes, but the nodes' energy saving problem was ignored. Especially when there are a few malicious nodes in a network, the overall trust value calculation for all nodes would waste lots of energy. Beside that, the network failure caused by nodes death was also not considered. In this paper, we proposed a method for avoiding energy hole which applied trust evaluation models and a trust evaluation method based on information entropy, so as to achieve the purpose of improving nodes utilization. Simulation results show that the proposed method can effectively improve nodes utilization, and it has reasonable detection rate and lower false alert rate compared to other classical methods.

Keywords: Wireless sensor networks; nodes utilization; energy hole; information entropy; trust evaluation

http://doi.org/10.3837/tiis.2018.03.008 **ISSN : 1976-7277**

1. Introduction

Wireless sensor networks (WSNs) are usually randomly deployed in harsh environments for monitoring events such as temperature changes, fire, etc. The energy of sensor node is limited, and because of the harsh and unattended environment, the energy of sensor node can not be timely supplied. How to keep WSNs running safely under the premise of spending less energy has become the research focus. Traditional cryptography network security methods are highly secure, but they also cost numerous energy. Currently, the common method is evaluating the trust value of sensor nodes, which can ensure the security of WSNs.

Trust evaluation models can accurately detect malicious nodes, however there is another problem we should solve, which is how to improve the utilization of sensor nodes. The main task of WSNs is to transmit events in the monitoring area to Sink node. How to transmit more events before nodes run out of energy is one of main indicators to assess nodes utilization. In this paper, the indicators to assess nodes utilization include the nodes lifetime, the success rate of event transmission and the number of event transmission packet.

1. The nodes lifetime

Nodes lifetime refers to the average lifetime of all sensor nodes in the monitoring area. If the nodes lifetime is long, it indicates that nodes' energy consumption is balanced.

2. The success rate of event transmission

Assume that in a certain period the monitoring area generated *m* events in total, and Sink node received *n* events. Then the success rate of event transmission in this period is *n*/*m*. It's wasted effort for transmitting events if events sensed by nodes can't be successfully transmitted to Sink node. This requires WSNs to be secure and fully connected.

3. The number of event transmission packets

In trust evaluation models, forwarding trust value packets to Sink node will consume part of the energy. Besides, trust value calculation will also consume energy. Therefore, the number of event transmission packets will be reduced. When designing trust evaluation model for WSNs, it needs to consider not only security and accuracy but also energy saving. So that nodes can transmit more events information packets.

In this paper, we proposed a method for avoiding energy hole which applied to trust evaluation models and a trust evaluation method based on information entropy, so as to achieve the purpose of improving nodes utilization. Simulation results show that the proposed method can effectively improve nodes utilization; it has reasonable detection rate when the percentage of malicious nodes is below 20%; and it has lower false alert rate compared to other classical methods.

2. Related Work

2.1 Trust Evaluation Models

The general steps of WSNs trust evaluation models are as follows: all nodes in each period will calculate the trust value of neighbors according to the interactions with neighbors, and then send these trust value to Sink node, finally Sink node aggregates these trust value to calculate the final trust of all nodes. Over time, the trust value of legitimate nodes will become higher and higher while the trust value of malicious nodes will become lower and lower. S. Ganeriwal et al. [1] proposed a reputation-based framework for high integrity sensor networks (RFSN), which used a distributed trust management. Each node builds a trust table by watchdog mechanism, and the trust includes direct trust and indirect trust. The interactions between nodes are recorded to revise current trust, so the trust value of nodes is objective. This is why RFSN is used so widely. R. A. Shaikh et al. [2] proposed a group-based trust management scheme for clustered wireless sensor networks (GMTS). It uses distributed trust management within the cluster and centralized trust management between clusters. GMTS does not focus on calculating trust of a single node, so it requires less memory to store the trust records. In addition, GTMS uses integer from 0 to 100 to represent trust value, which can reduce the cost for storing trust value. A. Boukerche et al. [3] proposed an agent-based trust and reputation management scheme for wireless sensor networks (ATRM). The model is based on hierarchical structure and mobile agents, and the agent of every node has a trust management mechanism. Mobile agents manage trust to achieve the minimum load of trust information transmission. Jiang Jinfang et al. [4] proposed an efficient distributed trust model for wireless sensor networks (EDTM). The trust calculation in EDTM includes calculation of direct trust, recommendation trust, and indirect trust. Direct trust should consider communication trust, energy trust and data trust; trust reliability and familiarity are defined to improve the accuracy of recommendation trust; indirect trust is calculated by Trust Chain which is established by recommenders. Besides, some researchers proposed trust evaluation methods for WSNs based on cluster [5-8], in which the cluster members layer and the cluster heads layer use different methods to calculate trust values; some researchers proposed trust evaluation methods for WSNs through aggregating direct and indirect trust [9-10].

These WSNs trust evaluation models resolved WSNs security issues, but they did not effectively utilize all sensor nodes. On one hand, they can’t ensure the uniform distribution and coverage density of nodes, so there would be information redundancy exist. On the other hand, they spend too much energy to transmit packets which contain trust value.

2.1 Improving nodes utilization

As mentioned in section 1, when designing trust evaluation methods for WSNs, we should consider how to improve the nodes lifetime, the success rate of event transmission and the number of event transmission packet.

Scholars have proposed many routing protocols for WSNs to improve the nodes lifetime, such as LEACH [11], TEEN [12] and APTEEN [13]. Usman M J et al. [14] modified LEACH, and increased nodes lifetime better than LEACH.Author [15] proposed a coverage balancing based trust evaluation method (CBTE) for wireless sensor networks, which can effectively improve nodes lifetime. So in this paper, we will mainly focus on how to improve the other two indicators based on CBTE.

The premise of improving the success rate of event transmission is to ensure that the WSNs is fully connected. Data flow in WSNs is a multi-to-one pattern, this means that the nodes closer to Sink node need to take on more load [16]. Therefore, these nodes deplete their energy too early, and result in energy hole problem around Sink node. In CBTE, the sensor nodes are initially deployed uniformly, the simulation results also show that the size of working nodes is stable and the coverage of monitoring area is also balanced. So when using sensor nodes deployment method in CBTE, there would appear energy hole with the running of the network. Then WSNs can’t be full connectivity, and it will reduce the success rate of event transmission.

In [16] a good method of analyzing energy hole in WSNs is proposed, which divided the monitoring area into some concentric rings with Sink node in the center. It avoided energy hole by balancing energy consumption of each ring. Based on this, methods to solve energy hole problem in WSNs are mainly divided into three aspects, such as: Method in [17], the width of each ring is transmission radius of nodes, but the number of nodes between rings is not uniform. The ring closer to Sink node has higher nodes distribution density, so that it can balance energy consumption of each ring; Method in [18], the distribution of all nodes is uniform, but the width of each ring is not uniform. The ring closer to Sink node has larger width, so that it also can balance energy consumption of each ring; Method in [19], there are two kinds of nodes, working nodes and sleeping nodes. The distribution of working nodes is uniform, and the width of each ring is transmission radius of nodes. It uses sleep mechanism to balance energy consumption of each ring.

These methods focus on the energy of receiving data and sending data when analyzing the energy consumption of each ring. The main purpose of this paper is to optimize the trust evaluation method of WSNs. When evaluating trust value of one node, it needs to aggregate the direct trust values at neighbors, which are calculated by the intercepting nodes. So in this paper when analyzing how to avoid energy hole, we must consider the energy consumed in intercepting data. This will be discussed in section 3.

In addition, we proposed a trust evaluation method based on information entropy to improve the number of event transmission packet. This will be discussed in section 4.

3. Deployment of Sensor Nodes

3.1 Network Model and Assumptions

We use the same network model with [16], all nodes are deployed in a circle monitoring area with the radius R, and the only Sink node is deployed in the center, as shown in **Fig. 1**. The transmission radius of all nodes is r, and the network is divided into $M=R/r$ rings with the width r. From the inside out, C_i denotes the i-th ring, and Q_i denotes the density of nodes (including working nodes and sleeping nodes) in i-th ring. As can be seen from **Fig. 1**, neighbors of nodes in C_i only appear in C_{i+1}, C_i and C_{i-1}. The main purpose of this section is to find out the relationship of nodes number between two neighboring rings, and deploy sensor nodes according to this to avoid energy hole.

To assist in the analysis, several assumptions are presented in this section as follows:

- There is no other communication in addition to the data sent to Sink node one-way from all nodes;
- Working nodes in C_i (i>1) can send data to C_{i-1} by one hop, particularly, working nodes in C_1 can send data directly to Sink node;
- All data do not need to be aggregated in the whole process of sending to Sink node;
- There are no network congestion and network latency, and also no packet loss and packet resending.
- According to simulation results of CBTE, the density of working nodes in whole monitoring area is stable. So this section assumes that the density of working nodes at any time is same, denoted as q.

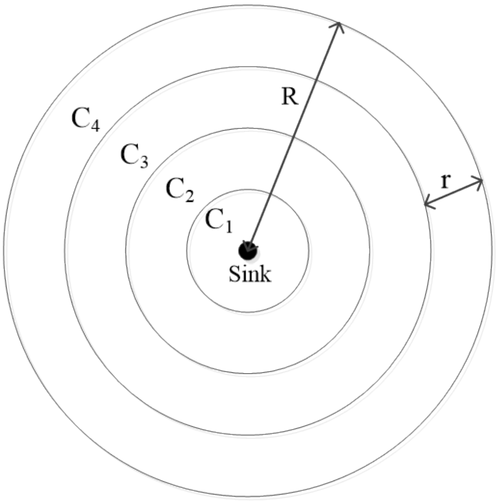


Fig. 1. Network model

3.2 Analysis of Energy Consumption in Each Ring

Sensor nodes have four states: sending state, receiving state, intercepting state and sleeping state. Nodes in sending state consume most energy, and nodes in receiving state consume less energy than nodes in sending state. Nodes in intercepting state consume almost the same energy with nodes in receiving state. Nodes in sleeping state consume extremely few energy, which is negligible compared to the previous three states. Energy consumption model in this paper is: Sink node has enough energy, and the initial energy of all nodes are same, denoted as ε; Node consumes energy e_1 by sending 1 bit data, consumes energy e_2 by receiving 1 bit data, and consumes energy e_3 by intercepting 1 bit data.

3.2.1 Energy Analysis of Sending and Receiving State

According to the assumptions in section 3.1, working nodes in C_M only need to send data generated in C_M to $C_{M\text{-}1}$, however working nodes in C_i ($i<M$) need both to transmit data sent from C_{i+1} to $C_{i\text{-}1}$ and send data generated in C_i to $C_{i\text{-}1}$.

Working nodes update every ΔT_u time, and every period ΔT has N_u ($N_u=\Delta T/\Delta T_u$) working nodes updating. After working nodes updating every time, each working node generates constant X packets with size L bits in the later ΔT_u time. In following analysis, we denote the area of C_i with S_i and the nodes density of C_i with Q_i.

For C_M, during one period ΔT, the data amount sent to $C_{M\text{-}1}$ is

$$D_M = S_M \cdot q \cdot N_u \cdot X \cdot L \tag{1}$$

Because all nodes in C_M do not receive data, the energy consumed in sending data is

$$E_M = D_M \cdot e_1 = S_M \cdot q \cdot N_u \cdot X \cdot L \cdot e_1 \tag{2}$$

For $C_{M\text{-}1}$, during one period ΔT, the data amount sent to $C_{M\text{-}2}$ is

$$D_{M-1} = D_M + S_{M-1} \cdot q \cdot N_u \cdot X \cdot L = \left(S_M + S_{M-1}\right) \cdot q \cdot N_u \cdot X \cdot L \tag{3}$$

All nodes in $C_{M\text{-}1}$ consumed energy in sending and receiving data is

$$E_{M-1} = D_M \cdot e_2 + D_{M-1} \cdot e_1 = S_M \cdot q \cdot N_u \cdot X \cdot L \cdot e_2 + \left(S_M + S_{M-1}\right) \cdot q \cdot N_u \cdot X \cdot L \cdot e_1 \tag{4}$$

And the like, for any C_i ($i<M$), during one period ΔT, the data amount sent to $C_{i\text{-}1}$ (particularly, C_1 send to Sink node directly) is

$$D_i = q \cdot N_u \cdot X \cdot L \cdot \sum_{k=i}^{M} S_k \tag{5}$$

All nodes in C$_i$ consumed energy in sending and receiving data is

$$E_i = q \cdot N_u \cdot X \cdot L \cdot e_2 \cdot \sum_{k=i+1}^{M} S_k + q \cdot N_u \cdot X \cdot L \cdot e_1 \cdot \sum_{k=i}^{M} S_k \tag{6}$$

So for each C$_i$, during one period ΔT, it consumed energy in sending and receiving data is as formula (7):

$$E_i = \begin{cases} q \cdot N_u \cdot X \cdot L \cdot e_2 \cdot \sum\limits_{k=i+1}^{M} S_k + q \cdot N_u \cdot X \cdot L \cdot e_1 \cdot \sum\limits_{k=i}^{M} S_k & i < M \\ S_M \cdot q \cdot N_u \cdot X \cdot L \cdot e_1 & i = M \end{cases} \tag{7}$$

3.2.2 Energy Analysis of intercepting State

When working nodes send or receive data, they also need to intercept the data sent by their neighbors. When neighbor sends a packet, node firstly intercepts the packet header. If the next hop is itself, the node receives the entire packet. Otherwise, the packet will be ignored by the node. The length of each packet header is constant, *l* bits. This part of the data will be intercepted by neighbors. When working nodes in C$_i$ ($1<i<M$) send data to Sink node, working nodes in C$_{i+1}$, C$_i$ and C$_{i-1}$ will intercept the data. Particularly, the data sent from C$_1$ will only be intercepted by C$_1$ and C$_2$, and the data sent from C$_M$ will only be intercepted by C$_M$ and C$_{M-1}$.

Because the average number of neighbors of each working node is

$$N_{neib} = \lfloor q \cdot \pi r^2 - 1 \rfloor \tag{8}$$

where, $\lfloor * \rfloor$ denotes round down. The data sent by each working node will be intercepted by N_m (N_m=N_{neib}) neighbors on average.

For C$_M$, during one period ΔT, the data amount sent to C$_{M-1}$ is as formula (1), and the data amount of packet header in it is $D_M^H = S_M \cdot q \cdot N_u \cdot X \cdot l$. This part of data will be intercepted by N_m neighbors on average. So the data amount intercepted by C$_M$ and C$_{M-1}$ is

$$I_{M,M-1} = N_m \cdot D_M^H = N_m \cdot S_M \cdot q \cdot N_u \cdot X \cdot l \tag{9}$$

Correspondingly, the energy consumptions of intercepting this part of data for C$_M$ and C$_{M-1}$ are respectively as follow:

$$F_{M,2} = \frac{S_M}{S_{M-1} + S_M} \cdot e_3 \cdot I_{M-1,M} = \frac{2M-1}{4M-4} e_3 N_m q N_u X l S_M \tag{10}$$

$$F_{M-1,1} = \frac{S_{M-1}}{S_{M-1} + S_M} \cdot e_3 \cdot I_{M-1,M} = \frac{2M-3}{4M-4} e_3 N_m q N_u X l S_M \tag{11}$$

For C$_{M-1}$, during one period ΔT, the data amount sent to C$_{M-2}$ is as formula (3), and the data amount of packet header in it is $D_{M-1}^H = (S_M + S_{M-1}) \cdot q \cdot N_u \cdot X \cdot l$. So the data amount intercepted by C$_M$, C$_{M-1}$ and C$_{M-2}$ is

$$I_{M,M-1,M-2} = N_m \cdot D_{M-1}^H = N_m \cdot (S_M + S_{M-1}) \cdot q \cdot N_u \cdot X \cdot l \tag{12}$$

Correspondingly, the energy consumptions of intercepting this part of data for C$_M$, C$_{M-1}$ and C$_{M-2}$ are respectively as follow:

$$F_{M,3} = \frac{S_M}{S_{M-2} + S_{M-1} + S_M} \cdot e_3 \cdot I_{M,M-1,M-2} = \frac{2M-1}{6M-9} e_3 N_m q N_u X l (S_M + S_{M-1}) \tag{13}$$

$$F_{M-1,2} = \frac{S_{M-1}}{S_{M-2}+S_{M-1}+S_M}\cdot e_3 \cdot I_{M,M-1,M-2} = \frac{1}{3} e_3 N_m q N_u Xl\left(S_M + S_{M-1}\right) \tag{14}$$

$$F_{M-2,1} = \frac{S_{M-2}}{S_{M-2}+S_{M-1}+S_M}\cdot e_3 \cdot I_{M,M-1,M-2} = \frac{2M-5}{6M-9} e_3 N_m q N_u Xl\left(S_M + S_{M-1}\right) \tag{15}$$

For C$_{M-2}$, during one period ΔT, the data amount sent to C$_{M-3}$ is D_{M-2}, and the data amount of packet header in it is $D_{M-2}^{H} = \left(S_M + S_{M-1} + S_{M-2}\right)\cdot q \cdot N_u \cdot X \cdot l$. So the data amount intercepted by C$_{M-1}$, C$_{M-2}$ and C$_{M-3}$ is

$$I_{M-1,M-2,M-3} = N_m \cdot D_{M-2}^{H} = N_m \cdot \left(S_M + S_{M-1} + S_{M-2}\right)\cdot q \cdot N_u \cdot X \cdot l \tag{16}$$

Correspondingly, the energy consumptions of intercepting this part of data for C$_{M-1}$, C$_{M-2}$ and C$_{M-3}$ are respectively as follow:

$$F_{M-1,3} = \frac{S_{M-1}}{S_{M-3}+S_{M-2}+S_{M-1}}\cdot e_3 \cdot I_{M-1,M-2,M-3} = \frac{2M-3}{6M-15} e_3 N_m q N_u Xl \sum_{k=M-2}^{M} S_k \tag{17}$$

$$F_{M-2,2} = \frac{S_{M-2}}{S_{M-2}+S_{M-1}+S_M}\cdot e_3 \cdot I_{M-1,M-2,M-3} = \frac{1}{3} e_3 N_m q N_u Xl \sum_{k=M-2}^{M} S_k \tag{18}$$

$$F_{M-3,1} = \frac{S_{M-3}}{S_{M-3}+S_{M-2}+S_{M-1}}\cdot e_3 \cdot I_{M-1,M-2,M-3} = \frac{2M-7}{6M-15} e_3 N_m q N_u Xl \sum_{k=M-2}^{M} S_k \tag{19}$$

So, for C$_M$, during one period ΔT, the energy consumptions of intercepting data is

$$F_M = F_{M,2} + F_{M,3} = \frac{2M-1}{4M-4} e_3 N_m q N_u XlS_M + \frac{2M-1}{6M-9} e_3 N_m q N_u Xl\left(S_M + S_{M-1}\right) \tag{20}$$

And for C$_{M-1}$, during one period ΔT, the energy consumptions of intercepting data is

$$\begin{aligned} F_{M-1} &= F_{M-1,1} + F_{M-1,2} + F_{M-1,3} = \frac{2M-3}{4M-4} e_3 N_m q N_u XlS_M \\ &+ \frac{1}{3} e_3 N_m q N_u Xl\left(S_M + S_{M-1}\right) + \frac{2M-3}{6M-15} e_3 N_m q N_u Xl \sum_{k=M-2}^{M} S_k \end{aligned} \tag{21}$$

And the like, for C$_i$ (2<i<M-1), during one period ΔT, the energy consumptions of intercepting data is

$$F_i = \frac{2i-1}{6i-9} e_3 N_m q N_u Xl \sum_{k=i-1}^{M} S_k + \frac{1}{3} e_3 N_m q N_u Xl \sum_{k=i}^{M} S_k + \frac{2i-1}{6i+3} e_3 N_m q N_u Xl \sum_{k=i+1}^{M} S_k \tag{22}$$

For C$_2$, during one period ΔT, the energy consumptions of intercepting data is

$$F_2 = \frac{3}{4} e_3 N_m q N_u Xl \sum_{k=1}^{M} S_k + \frac{1}{3} e_3 N_m q N_u Xl \sum_{k=2}^{M} S_k + \frac{1}{5} e_3 N_m q N_u Xl \sum_{k=3}^{M} S_k \tag{23}$$

For C$_1$, during one period ΔT, the energy consumptions of intercepting data is

$$F_1 = \frac{1}{4} e_3 N_m q N_u Xl \sum_{k=1}^{M} S_k + \frac{1}{9} e_3 N_m q N_u Xl \sum_{k=2}^{M} S_k \tag{24}$$

3.3 Nodes Deployment

According to the analysis in section 3.2, for each C$_i$, during one period ΔT, its total energy consumption is

$$E_i^{total} = E_i + F_i \tag{25}$$

So the lifetime of C$_i$ is $\frac{S_{\mathrm{i}} \cdot Q_i \cdot \varepsilon}{E_i^{total}} \cdot \Delta T$. Ideally, if all rings exhaust their own energy simultaneously, the utilization of nodes energy achieves the best. That is to say the lifetimes of all rings are equivalent. So we get formula (26):

$$\frac{S_1 \cdot Q_1 \cdot \varepsilon}{E_1^{total}} \cdot \Delta T = \frac{S_2 \cdot Q_2 \cdot \varepsilon}{E_2^{total}} \cdot \Delta T = \cdots = \frac{S_{M-1} \cdot Q_{M-1} \cdot \varepsilon}{E_{M-1}^{total}} \cdot \Delta T = \frac{S_M \cdot Q_M \cdot \varepsilon}{E_M^{total}} \cdot \Delta T \tag{26}$$

So, $S_i Q_i \cdot E_{i-1}^{total} = S_{i-1} Q_{i-1} \cdot E_i^{total}$. Combined with formulas in section 3.2, we get the relationship of S$_{i\text{-}1}$Q$_{i\text{-}1}$ and S$_i$Q$_i$ as follow:

$$\frac{S_{i-1}Q_{i-1}}{S_i Q_i} = \begin{cases} \dfrac{e_2 L \sum_{k=2}^{M} S_k + e_1 L \sum_{k=1}^{M} S_k + e_3 N_m l \left[\frac{1}{4} \sum_{k=1}^{M} S_k + \frac{1}{9} \sum_{k=2}^{M} S_k \right]}{e_2 L \sum_{k=3}^{M} S_k + e_1 L \sum_{k=2}^{M} S_k + e_3 N_m l \left[\frac{3}{4} \sum_{k=1}^{M} S_k + \frac{1}{3} \sum_{k=2}^{M} S_k + \frac{1}{5} \sum_{k=3}^{M} S_k \right]} & i = 2 \\[2ex] \dfrac{e_2 L \sum_{k=3}^{M} S_k + e_1 L \sum_{k=2}^{M} S_k + e_3 N_m l \left[\frac{3}{4} \sum_{k=1}^{M} S_k + \frac{1}{3} \sum_{k=2}^{M} S_k + \frac{1}{5} \sum_{k=3}^{M} S_k \right]}{e_2 L \sum_{k=4}^{M} S_k + e_1 L \sum_{k=3}^{M} S_k + e_3 N_m l \left[\frac{5}{9} \sum_{k=2}^{M} S_k + \frac{1}{3} \sum_{k=3}^{M} S_k + \frac{5}{21} \sum_{k=4}^{M} S_k \right]} & i = 3 \\[2ex] \dfrac{e_2 L \sum_{k=i}^{M} S_k + e_1 L \sum_{k=i-1}^{M} S_k + e_3 N_m l \left[\frac{2i-3}{6i-15} \sum_{k=i-2}^{M} S_k + \frac{1}{3} \sum_{k=i-1}^{M} S_k + \frac{2i-3}{6i-3} \sum_{k=i}^{M} S_k \right]}{e_2 L \sum_{k=i+1}^{M} S_k + e_1 L \sum_{k=i}^{M} S_k + e_3 N_m l \left[\frac{2i-1}{6i-9} \sum_{k=i-1}^{M} S_k + \frac{1}{3} \sum_{k=i}^{M} S_k + \frac{2i-1}{6i+3} \sum_{k=i+1}^{M} S_k \right]} & 3 < i < M-1 \\[2ex] \dfrac{e_2 L \sum_{k=M-1}^{M} S_k + e_1 L \sum_{k=M-2}^{M} S_k + e_3 N_m l \left[\frac{2M-5}{6M-21} \sum_{k=M-3}^{M} S_k + \frac{1}{3} \sum_{k=M-2}^{M} S_k + \frac{2M-5}{6M-9} \sum_{k=M-1}^{M} S_k \right]}{e_2 L S_M + e_1 L \sum_{k=M-1}^{M} S_k + e_3 N_m l \left[\frac{2M-3}{6M-15} \sum_{k=M-2}^{M} S_k + \frac{1}{3} \sum_{k=M-1}^{M} S_k + \frac{2M-3}{4M-4} S_M \right]} & i = M-1 \\[2ex] \dfrac{e_2 L S_M + e_1 L \sum_{k=M-1}^{M} S_k + e_3 N_m l \left[\frac{2M-3}{6M-15} \sum_{k=M-2}^{M} S_k + \frac{1}{3} \sum_{k=M-1}^{M} S_k + \frac{2M-3}{4M-4} S_M \right]}{e_1 L S_M + e_3 N_m l \left[\frac{2M-1}{6M-9} \sum_{k=M-1}^{M} S_k + \frac{2M-1}{4M-4} S_M \right]} & i = M \end{cases} \tag{27}$$

In this paper, nodes in C$_M$ are firstly deployed. Nodes density in CBTE is Q_M=10.87/r^2, so the number of nodes deployed in C$_M$ is

$$S_M Q_M = \left[\pi (Mr)^2 - \pi \left((M-1) r \right)^2 \right] \cdot \frac{10.87}{r^2} = 10.87\pi (2M-1) \tag{28}$$

Then, nodes are deployed from the outer ring to the inner ring one by one according to formula (27).

4. Trust Evaluation Method Based on Information Entropy

Information entropy [20] is proposed by C. E. Shannon, which references the concept of thermodynamic. Entropy in thermodynamics is a physical quantity which describes the confusion of molecule, and the information entropy describes the uncertainty of information sources. The larger the information entropy is, the more uncertain information sources are.

The mathematical expression of information entropy is shown in formula (29):

$$H = -\sum_{i} p_i \cdot \log_2 p_i \tag{29}$$

Where p_i denotes the probability of i-th information source, and H is information entropy.

In WSNs, we evaluate the uncertainty of malicious interaction of all nodes. For any node, all of its neighbors are information sources. Because these information sources are discrete, we use discrete information entropy to calculate the information entropy of malicious interaction. The larger the information entropy of malicious interaction is, the more uncertain information sources are. That is to say the node is uncertain whether neighbors have malicious interactions. So in order to save energy, the node will not send trust value of neighbors to Sink node. On the contrary, the smaller the information entropy is, the more certain the node is that neighbors have malicious interactions. So the node will send trust value of neighbors to Sink node.

Definition 1. *n*-unit information entropy. The information entropy which has *n* non-zero information sources is called *n*-unit information entropy.

4.1 Information entropy of malicious interaction

In WSNs, there are three direct attacks when malicious nodes send packets: refuse forwarding attack, tampering attack, and flooding attack. When malicious nodes refuse forwarding packets or tamper packets, they will be discovered by their neighbors timely. If malicious nodes run out of neighbors' energy by flooding attack, their neighbors will also detect the abnormality. Because the number of packets sent actively per unit time is limited whether in real network or simulation experiments. In real network, nodes send packets to Sink node actively when they sense some event, and in simulation experiments, we set every node the number of packets sent actively per second instead of the number of sensing events per second.

When nodes detect packets has been tampered, it indicates there are malicious nodes in their neighbors, then they need to send trust value of their neighbors to Sink node. When nodes detect packets has been refused forwarding, it can't indicate there must be malicious nodes in their neighbors, because network congestion and network latency will also lead to packets loss. At this time, we need to calculate the information entropy of refusing forwarding to determine whether there are refuse forwarding attacks or not. When nodes detect their neighbors have sent packets actively, it is also necessary to calculate information entropy of sending actively to determine whether there are flooding attacks or not.

4.1.1 Information entropy of refusing forwarding

Assuming that the packet loss rate is p_l, in other words the average loss rate of each node is p_l. Comparing p_l with the packet loss rate of neighbor at one period can only detect part of malicious behaviors. If the latter is larger than the former, it can't indicate that the neighbor node is malicious node, because WSNs itself exist some packet loss caused by network congestion and network latency. On the contrary, it also can't indicate that the neighbor node is legitimate node, because malicious node may keep dropping packets with a rate below p_l, so that it can attack network continuously without being detected. Calculating the information entropy of refusing forwarding can effectively solve this problem, and malicious nodes will be promptly detected.

As mentioned in section 3, nodes evaluate the trust values of neighbors by intercepting their behavior. For node *i*, each period it counts the number of packet loss of its neighbors. Assuming that the number of packets which are sent to neighbor node *j* from node *i* is N_{ij}, and the number of packet loss by nodes *j* is L_{ij}. L_{ij} is standardized as $(L_{ij})_s = \lfloor L_{ij}/N_{ij} \rfloor \times 100$, which denotes the number of packet loss when node *i* sent 100 packets to neighbor node *j*. p_l is standardized as $(P_l)_s = \lfloor p_l \times 100 \rfloor$, which denotes the average number of packet loss when one node sent 100 packets to another node. So the contribution to information entropy of refusing forwarding for $(L_{ij})_s$ is calculated by the formula (30) and (31):

$$d_{ij} = \left| (L_{ij})_s - (P_l)_s/2 \right| \quad 0 < j < n \tag{30}$$

$$p_{ij} = \frac{d_{ij}}{\sum\limits_{1 \le k \le n} d_{ik}} \quad 0 < j < n \tag{31}$$

Where, *n* denotes the number of neighbors of node *i*.

The bigger the difference in (30) is, the bigger the contribution is. When $(L_{ij})_s$ is much deviated from $(P_l)_s$, some abnormal situation must be there which makes the entropy decrease.

So, the information entropy of refusing forwarding at the current period is calculated as follows (32):

$$\left(HR_i\right)_t = -\sum_{1 \le j \le n} p_{ij} \cdot \log_2 p_{ij} \tag{32}$$

If $\left(HR_i\right)_t \ge \left(HR_i\right)_{t-1}$, it indicates node *i* is uncertain whether there are refuse forwarding attacks in its neighbors. In order to save energy, node i does not send the trust value of its neighbors to Sink node. On the contrary, it indicates that there are very likely refuse forwarding attacks in its neighbors, so node *i* needs to send the trust value of its neighbors to Sink node.

The above method can't detect malicious nodes when malicious nodes attack WSNs as follows: Malicious node at first period drops a large number of packets, and at the after every period drops fewer packets than last period. In this situation, the information entropy at each period is larger than last period. Node is uncertain whether there are refuse forwarding attacks in its neighbors and will not send the trust value of its neighbors to Sink node. The malicious node can't be detected, but the fact is that malicious node attacks network continuously. So it is necessary to calculate the minimum information entropy of refusing forwarding under the premise of reasonable packet loss. The information entropy at each period must not only be not less than last period but also be not less than the minimum information entropy, so that the above attack can't destroy the network.

In formula (30), when $(L_{ij})_s \le (P_l)_s$, the packet loss is reasonable. Therefore, under the premise of reasonable packet loss, the maximum of d_{ij} is $(P_l)_s/2$ and the minimum of d_{ij} is 0. d_{ij} =0 has no contribution to information entropy, so we only consider the non-zero information sources, and the minimum of d_{ij} is 1/2.

Assuming that node *i* has *N* non-zero information sources at one period. The algorithm of calculating the minimum information entropy is as **Algorithm 1**. As we can see this is a recursive process, and it will stop when *N* decrease to 2. Every round, if *N* is an odd, *H* can be reduced to (*N*-1)-unit information entropy by Corollary 3 in Appendix as step 3 and 4. If *N* is an even, *H* can be expressed as the sum of one (*N*/2)-unit information entropy and *N*/2 2-unit information entropy by Corollary 4 in Appendix as step 9.

Algorithm 1. algorithm of calculating minimum information entropy

Input: $H = H(p_{i1}, p_{i2}, \cdots, p_{i,N})$; $S = \{p_{i1}, p_{i2}, \cdots p_{i,N}\}$.

Output: *H*.

```
1   While (N > 2){
2       If (N == odd){
3           Bubble_Sort(S);          //Set S is bubble sorted one round and
                                     //the minimum is in the final of set S
4           $H = H(p_{i1}+\frac{p_{i,N}}{N-1}, p_{i2}+\frac{p_{i,N}}{N-1}, \cdots, p_{i,N-1}+\frac{p_{i,N}}{N-1})$;
5           $S = \left\{p_{i1}+\frac{p_{i,N}}{N-1}, p_{i2}+\frac{p_{i,N}}{N-1}, \cdots, p_{i,N-1}+\frac{p_{i,N}}{N-1}\right\}$;
6           $N = N-1$;
7       }
8       Else{
9           $H = H(p_{i1}+p_{i2}, p_{i3}+p_{i4}, \cdots, p_{i,N-1}+p_{i,N})$
                $+\sum_{n=0}^{N/2-1}\left(p_{i,2n+1}, p_{i,2n+2}\right)H(\frac{p_{i,2n+1}}{p_{i,2n+1}, p_{i,2n+2}}, \frac{p_{i,2n+2}}{p_{i,2n+1}, p_{i,2n+2}})$;
```

```
            $S = \{p_{i1} + p_{i2}, p_{i3} + p_{i4}, \cdots, p_{i,N-1} + p_{i,N}\}$;
            $N = N/2$;
        }
    }
```

Finally, the reasonable minimum of information entropy of refusing forwarding which has N non-zero information sources is $(HR_i)_{\min} = \lfloor \log_2 N \rfloor H(\frac{1}{1+(P_l)_s}, \frac{(P_l)_s}{1+(P_l)_s})$.

So if $(HR_i)_t \geq (HR_i)_{\min}$ and $(HR_i)_t \geq (HR_i)_{t-1}$, it indicates that node i is uncertain whether there are refusing forwarding attacks in its neighbors and does not send trust value of neighbors to Sink node. Otherwise node i needs to send trust value of neighbors to Sink node.

4.1.2 Information entropy of sending actively

Nodes in real network will send packets to Sink node actively when they sense events. However in simulation experiments we set every node the number of packets sent actively per second instead of the number of sensing events per second. Assuming that the number of packets sent actively per second is Ns. After each period, comparing the number of packets sent actively by neighbors per second with Ns is pointless. Because of network congestion, if a packet hasn't been forwarded by the next hop within a certain time, the packet will be resent actively. It causes an increase in the number of packets sends actively. On the other hand, if there is serious network congestion in local area, the routing protocol will stop nodes in this area sending packets temporarily to alleviate network congestion. It causes a decrease in the number of packets sends actively. So comparing the number of packets sent actively by neighbors per second with N_s can't detect malicious behavior. Calculating the information entropy of sending actively can effectively solve this problem, and malicious nodes will be promptly detected.

For node i, each period it counts the number of packets sent actively per second by its neighbors. When node i receives a packet sent by its neighbor node j, it judges whether the packet is sent by node j actively according to ID_s in packet. Assuming that the number of packets sent actively by node j at current period is N_j, namely the number of packets sent actively by node j per second in this period is $(N_j)_s = \lfloor N_j / \Delta\text{T} \rfloor$, where ΔT denotes the duration of a period. So the contribution to information entropy of sending actively for $(N_j)_s$ is calculated by the formula (33) and (34):

$$D_{ij} = \left|(N_j)_s - N_s/2\right| \quad 0 < j < n \tag{33}$$

$$P_{ij} = \frac{D_{ij}}{\sum\limits_{1 \le k \le n} D_{ik}} \quad 0 < j < n \tag{34}$$

Where, *n* denotes the number of neighbors of node *i*.

So, the information entropy of sending actively at the current period is calculated as follows (35):

$$\left(HS_i\right)_t = -\sum_{1 \le j \le n} P_{ij} \cdot \log_2 P_{ij} \tag{35}$$

By the same token, the reasonable minimum of information entropy of sending actively which has *N* non-zero information sources is $\left(HS_i\right)_{\min} = \lfloor \log_2 N \rfloor H(\frac{1}{1+N_s}, \frac{N_s}{1+N_s})$.

So if $\left(HS_i\right)_t \ge \left(HS_i\right)_{\min}$ and $\left(HS_i\right)_t \ge \left(HS_i\right)_{t-1}$, it indicates that node *i* is uncertain whether there are flooding attacks in its neighbors and does not send trust value of neighbors to Sink node. Otherwise node *i* needs to send trust value of neighbors to Sink node.

4.2 Trust evaluation and aggregation

At the end of every period, if node *i* is certain there are no malicious nodes in its neighbors, it does not send trust value of neighbors to Sink node. Certainly, node *i* does not need to calculate the trust value of neighbors at this time. Otherwise, node *i* needs to calculate the trust value of neighbors and send them to Sink node. So when there are no malicious nodes or fewer malicious nodes in WSNs, only some of the nodes need to send trust value of neighbors to Sink node. This saves energy of nodes.

When node *i* needs to send trust value of neighbors to Sink node, it first calculates the trust value of its neighbors as formula (36) according to the attacks of malicious nodes:

$$T_{ij} = \frac{S_{ij} - D_{ij} - L_{ij} - \theta F_{ij}}{S_{ij}} \tag{36}$$

Where, S_{ij} denotes the total number of packets nodes *i* interacted with node *j* at current period; D_{ij} denotes the number of packets tampered by node *j* at current period; L_{ij} denotes the number of packets dropped by node *j* at current period; F_{ij} denotes the number of flooding attack packets sent by node *j* at current period. θ is the sign of flooding attack, if there is no flooding attack, $\theta = 0$. Otherwise, $\theta = 1$, and at this time F_{ij} is calculated as formula (37):

$$F_{ij} = \begin{cases} N_{ij} - N_s \cdot \Delta \mathrm{T} & N_{ij} - N_s \cdot \Delta \mathrm{T} > 0 \\ 0 & else \end{cases} \tag{37}$$

Where, N_{ij} denotes the number of packets that node *j* sent to node *i* actively at current period. N_s is same with N_s in formula (33).

Sink node aggregates the trust value of node *j* at current period as formula (38):

$$(T_j)_t = (1-\eta)(T_j)_{t-1} + \eta \frac{\sum_{i=1}^{K} T_{ij}}{K} \quad (38)$$

Where, $(T_j)_{t-1}$ denotes the trust value of node *j* at Sink node at last period; T_{ij} denotes the trust value of node *j* at node *i*. If Sink node has not received T_{ij}, then T_{ij} =0. *K* is the number of non-zero T_{ij}. η is the weight which is calculated as formula (39):

$$\eta = K/B_j \quad (39)$$

Where, B_j denotes the number of neighbor nodes of node *j*. Obviously, if all T_{ij} are 0, then $\eta = 0$, otherwise, $0 < \eta \le 1$. Especially, when all T_{ij} are 0, the trust value of node *j* at Sink node is same with the trust value at last period according to formula (38); and when all T_{ij} are non-zero, $\eta = 1$ according to formula (39), then the trust value of node *j* at Sink node is simplified as $(T_j)_t = \frac{\sum_{i=1}^{B_j} T_{ij}}{B_j}$.

5. Simulation Environment

In this paper, we set the simulation environment as follows: the monitoring area is a circular area with radius *R*; the transmission radius of sensor nodes is *r*; all nodes have same initial energy *ε*; Sink node is located in the center of the monitoring area. According to the requirement of the proposed method, we set that nodes send N_s packets actively per second instead of the number of events they sensed. When the energy of one node is lower than 20% of initial energy, the node stops sending actively. However, if there are less than 10% nodes who send packets actively, part of nodes whose energy are lower than 20% of initial energy will continue to send packets actively. Other related experiments parameters are set as shown in **Table 1**.

Table 1. Experiments parameters

parameter	value	parameter	value
R	70m	e_1	$161 \times 10^{-9} J/bit$
r	10m	e_2	$135 \times 10^{-9} J/bit$
ε	1J	e_3	$133 \times 10^{-9} J/bit$
ΔT	5s	p_l	6%
L	800bit	N_s	3
l	30bit		

Obviously, monitoring area is divided into seven rings with width 10m (Particularly, C_1 is a circular with radius 10m). According to the parameters above, the nodes deployment in each ring can be calculated as formula (28) and (27), as shown in **Table 2**.

Table 2. The nodes deployment in each ring

ring	nodes number	ring	nodes number
C_7	444	C_3	2390
C_6	1090	C_2	2648
C_5	1616	C_1	2454
C_4	2051		

We compared the proposed method with RFSN，EDTM and GTMS in following aspects:

- The node lifetime: For RFSN，EDTM and GTMS, node lifetime refers to the time from node adding into the network to node death. However, for the proposed method, node lifetime is only the accumulation of time when node is working.
- The success rate of event transmission: It is mentioned in section 2. Every 100 seconds we will calculate the success rate of event transmission.
- The number of packets: It mainly includes two types of packets, one is the number of event transmission packets, and the other is the number of trust transmission packets.
- The malicious nodes detection rate: The ratio of malicious nodes detected to all malicious nodes.
- The legitimate nodes false alert rate: The ratio of legitimate nodes mistaken for malicious nodes to all legitimate nodes.

At the beginning of the simulation, we selected 178 nodes randomly from all nodes as initial nodes according to CBTE. When the network reaches coverage balance in proposed method, the size of working nodes remains at 300 or so. So the nodes size of other three methods is set to be 300. Obviously, the nodes size of proposed method is 42 times the nodes size of other three methods. When we set malicious nodes artificially, the number of malicious nodes of proposed method is 42 times the number of other three methods, too.

6. Results and Discussion

To compare nodes lifetime in one figure, in every method we all sample 100 nodes randomly. **Fig. 2** compares nodes lifetime of proposed method and other three methods under the premise of no malicious nodes. As we can see, the average nodes lifetime of proposed method is about 1550 seconds, and RFSN，EDTM and GTMS are about 700，770 and 860 seconds respectively. It indicates that the proposed method can effectively increase nodes lifetime. The main reason is that the proposed method can keep working nodes coverage balancing on the basis of CBTE, and reduce lots of redundant information. In addition, each node in proposed method needs to calculate the information entropy of malicious interaction every period. If the information entropy is lower than previous period or lower than threshold, the node needs to send neighbors’ trust value to Sink node. For there are no malicious nodes, the information entropy of malicious interaction is always in reasonable range, and nodes do not need to send neighbors’ trust value to Sink node.

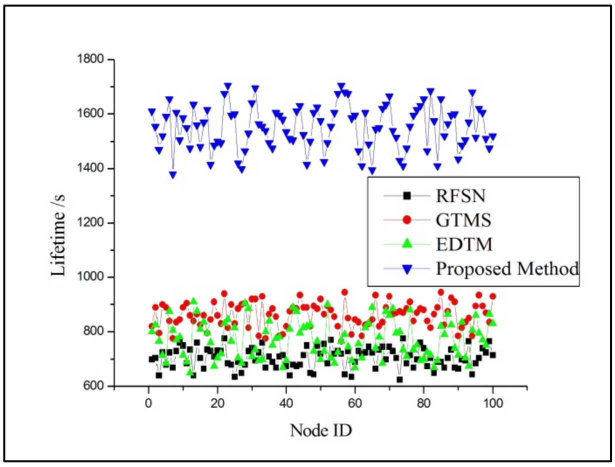


Fig. 2. Comparison of nodes lifetime

Fig. 3 shows the success rate of event transmission of proposed method and other three methods over time in the absence of malicious nodes, where each point represents the success rate of event transmission last 100 seconds. Because there is sleep mechanism in proposed method, we randomly select one node's whole working time as the time axis. As we can see, the success rate of event transmission of other three methods decline sharply at about 600 seconds, while the proposed method declines sharply at about 1400 seconds. The main reason for success rate of event transmission decreasing is the disconnected network due to the death of nodes. So **Fig. 3** illustrates that the proposed method can keep network connected for long time. In addition, the success rate of event transmission of proposed method is 100% from about 900 seconds to 1400 seconds. Because at this period the number of events generated is less and there is no network congestion. All events can be successfully transferred to Sink node. **Fig. 2** and **Fig. 3** indicate the proposed method can improve nodes utilization.

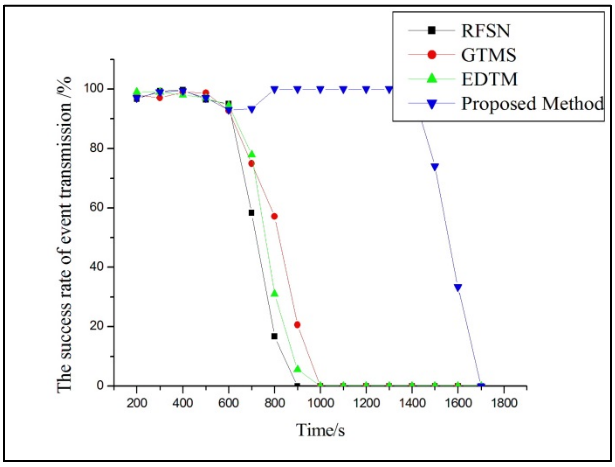
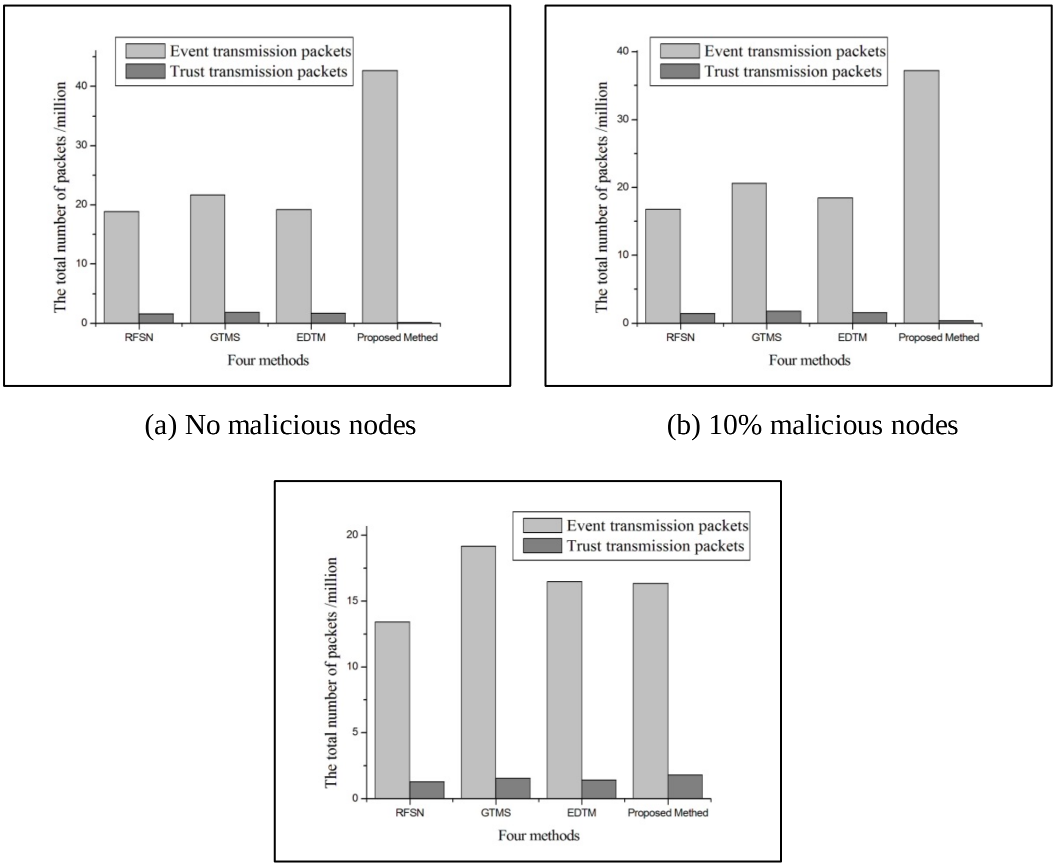


Fig. 3. Comparison of success rate of event transmission

Fig. 4 compares two types of packets of proposed method and other three methods with different malicious nodes percentage. Since the nodes size of proposed method is 42 times the nodes size of other two methods, the packets number of proposed method need to be divided by 42. As we can see, the number of trust transmission packets of proposed method is less than other three methods when there are no malicious nodes or only 10% malicious nodes. Because the proposed method calculates the information entropy of malicious interaction, only a small part of nodes need to send neighbors' trust value to Sink node. In **Fig. 4(a)** and **Fig. 4(b)** the proposed method transmits more event transmission packets than other three methods. So the proposed method improves nodes utilization when malicious nodes are few. In **Fig. 4(c)**, the numbers of trust transmission packets of four methods are almost the same. So the proposed method loses the advantage compared with other three methods.



(a) No malicious nodes

(b) 10% malicious nodes

(c) 40% malicious nodes

Fig. 4. Comparison of packets with different malicious nodes percentage

Fig. 5 shows the detection rate of proposed method and other three methods over the increment of malicious nodes. As we can see, when the percentage of malicious nodes is below about 20%, the detection rate of proposed method has no big difference from other three methods, and four methods are all higher than 80%. But with the increasing of malicious nodes, the detection rate of proposed method declines sharply. When the percentage of malicious nodes is above about 45%, the detection rate of proposed method is less than 5%, which means the proposed method almost is unable to detect any malicious node. This is because when the percentage of malicious nodes is higher, neighbors of many nodes are malicious nodes, and at this time the malicious behaviors will increase information entropy of malicious interaction. Nodes are uncertain whether there are malicious nodes in their neighbors, and do not send trust value of their neighbors to Sink node. So Sink node will not update the trust value of malicious nodes, and can’t detect malicious nodes.

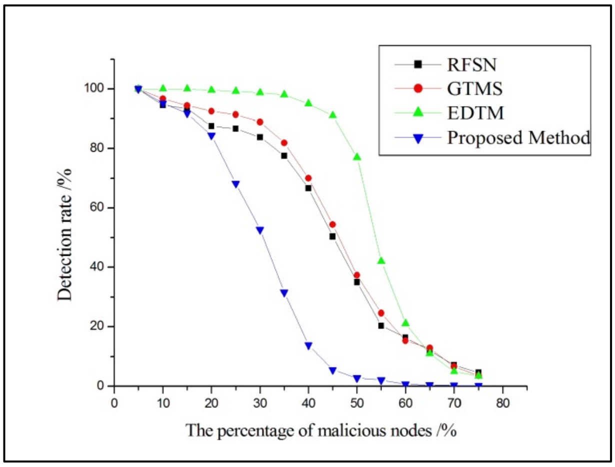


Fig. 5. Comparison of detection rate

Fig. 6 shows the false alert rate of proposed method and other three methods over the increment of malicious nodes. As we can see, the false alarm rate of proposed method is always less than other three methods, and is less than 10% when the percentage of malicious nodes is below about 30% and above about 50%. When the percentage of malicious nodes is below about 40%, the false alert rates of four methods are almost unanimous. When the percentage of malicious nodes is above about 45%, the proposed method almost can’t detect any malicious nodes according to **Fig. 5**. At this time, both legitimate nodes and malicious nodes do not send trust value of their neighbors to Sink node, so Sink node will not update the trust value of any nodes. That is to say, Sink node will not mistake legitimate nodes for malicious nodes. So the false alert rate will be very low, and decline more obviously with the increasing of malicious nodes.

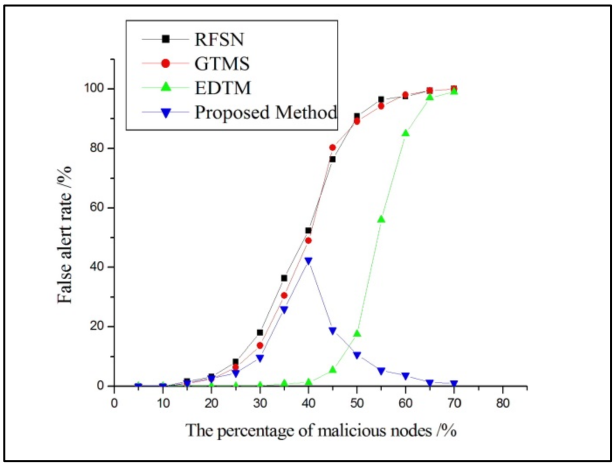


Fig. 6. Comparison of false alert rate

7. Conclusion

In this paper, we aimed to improve nodes utilization in trust evaluation models for WSNs, and proposed three indicators to assess nodes utilization: the nodes lifetime, the success rate of event transmission, and the number of event transmission packet. Existing coverage balancing based trust evaluation method can effectively improve nodes lifetime. On this basis, this paper proposed a method to avoid energy hole which applied to trust evaluation models and this method can improve the success rate of event transmission. In addition, we also proposed a trust evaluation method based on information entropy, which reduced lots of trust transmission packets and improved the number of event transmission packet. Our proposed method can effectively improve nodes utilization compared to RFSN，EDTM and GTMS. Simulation results showed that this method has reasonable detection rate when the percentage of malicious nodes is below 20% and has lower false alert rate compared with other classical methods.

The purpose of this paper is to improve node utilization, thus the detection rate of the proposed method is a bit lower than other methods. In the future, we will research more effective trust evaluation methods, which can not only improve nodes utilization, but also ensure the detection rate.

References

[1] S. Ganeriwal, L. K. Balzano and M. Srivastava, “Reputation based framework for high integrity sensor networks,” *ACM Transactions on Sensor Networks*, vol. 4, no. 3, pp. 1-37, May, 2008. Article (CrossRef Link)

[2] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee and Y. J. Song, “Group-based trust management scheme for clustered wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 11, pp. 1698-1712, November, 2009. Article (CrossRef Link)

[3] A. Boukerche and Xu Li, “An agent-based trust and reputation management scheme for wireless sensor networks,” in *Proc. of GLOBECOM - IEEE Global Telecommunications Conference*, pp. 1857-1861, November 28- December 2, 2005. Article (CrossRef Link)

[4] Jiang Jinfang, Han Guangjie, Wang Feng, Shu Lei and Guizani Mohsen, “An Efficient Distributed Trust Model for Wireless Sensor Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1228-1237, May, 2015. Article (CrossRef Link)

[5] J. Zhang, R. Shankaran, M. A. Orgun, V. Varadharajan and A. Sattar, “A trust management architecture for hierarchical wireless sensor networks,” in *Proc. of the IEEE 35th Conference on Local Computer Networks*, pp. 264-267, October 10-14, 2010. Article (CrossRef Link)

[6] J. Zhang, R. Shankaran, M. A. Orgun, V. Varadharajan, A. Sattar, “A Dynamic Trust Establishment and Management Framework for Wireless Sensor Networks,” in *Proc. of the IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing*, pp. 484-491, December 11-13, 2010. Article (CrossRef Link)

[7] V. R. S. Dhulipala, N. Karthik and R. M. Chandrasekaran, “A Novel Heuristic Approach Based Trust Worthy Architecture for Wireless Sensor Networks,” *Wireless personal communications*, vol. 70, no. 1, pp. 189-205, May, 2013.Article (CrossRef Link)

[8] Li Xiaoyong, Zhou Feng and Du Junping, “LDTS: A Lightweight and Dependable Trust System for Clustered Wireless Sensor Networks,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 924-935, June, 2013. Article (CrossRef Link)

[9] P. Trakadas, S. Maniatis, P. Karkazis, T. Zahariadis, H. C. Leligou and S. Voliotis, “A novel flexible trust management system for heterogeneous wireless sensor networks,” in *Proc. of International Symposium on Autonomous Decentralized Systems*, pp. 369-374, March 23-25, 2009. Article (CrossRef Link)

[10] T. Zahariadis, P. Trakadas, H. C. Leligou, S. Maniatis and P. Karkazis, “A novel trust-aware geographical routing scheme for wireless sensor networks,” *Wireless personal communications*, vol. 69, no. 2, pp. 805-826, March, 2013. Article (CrossRef Link)

[11] Heinzelman W R, Chandrakasan A, Balakrishnan H, “Energy-efficient communication protocol for wireless microsensor networks,” in *proc. of the 33rd Hawaii International Conference on System Sciences*, *IEEE Explore*, pp. 1-10, January 7, 2010. Article (CrossRef Link)

[12] A. Manjeshwar and D. P. Agrawal, “TEEN: a routing protocol for enhanced efficiency in wireless sensor networks,” in *proc. of the 15th International Parallel and Distributed Processing Symposium*, pp.30189a, April 23-27, 2001. Article (CrossRef Link)

[13] A. Manjeshwar and D. P. Agrawal, “APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks,” in *proc. of the International Parallel and Distributed Processing Symposium*, pp.0195b, April 15-19, 2002. Article (CrossRef Link)

[14] Usman M J, Xing Z, Chiroma H, and Herawan T, “Modified low energy adaptive clustering hierarchy protocol for efficient energy consumption in wireless sensor networks,” *International Review on Computers and Software*, vol. 9, no. 11, pp. 1904-1915, November, 2014. Article (CrossRef Link)

[15] Shen Haibo, Dong Shengjie, Jian Xu, and Zhang Hong, “Coverage Balancing Based Trust Evaluation Method for Wireless Sensor Networks,” *International Journal of Security and its Applications*, vol. 9, no. 10, pp. 381-394, October, 2015. Article (CrossRef Link)

[16] Li Jian and P. Mohapatra, “An Analytical Model For The Energy Hole Problem In Many-To-One Sensor Networks,” in *Proc. of IEEE Vehicular Technology Conference*, pp. 2721-2725, September 28-28, 2005. Article (CrossRef Link)

[17] Sha Chao, Chen Huan, Yao Chen, Liu Yao, Wang Ruchuan, “A Type of Energy Hole Avoiding Method Based on Synchronization of Nodes in Adjacent Annuluses for Sensor Network,” *International Journal of Distributed Sensor Networks*, vol. 12, no. 3, pp. 1-14, March, 2016. Article (CrossRef Link)

[18] Olariu Stephan and Stojmenovic´ Ivan, “Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting,” in *Proc. of INFOCOM 2006: 25th IEEE International Conference on Computer Communications*, pp. 1-12, April 23-29 2006. Article (CrossRef Link)

[19] Demertzis, Apostolos and Oikonomou, Konstantinos, "Avoiding Energy Holes in Wireless Sensor Networks with Non-Uniform Energy Distribution," in *Proc. of IISA 2014 - 5th International Conference on Information, Intelligence, Systems and Applications*, pp. 138-143, July 7-9, 2014.Article (CrossRef Link)

[20] C. E. Shannon, "A Mathematical Theory of Communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3-55, January, 2001. Article (CrossRef Link)

Shen Haibo is currently a Ph.D. candidate in Nanjing University of Science and Technology, supervised by Professor Hong Zhang. He received his B.S. degree in computer science from Nanjing University of Science and Technology in 2009. His research focus on wireless sensor networks security.

Zhuang Kechen is currently a Ph.D. candidate in Nanjing University of Science and Technology, supervised by Professor Hong Zhang. He received his B.S. degree in computer science from Nanjing University of Science and Technology in 2010. He had been a visiting student at University of Washington in 2013. His research interests include multimedia analysis, social network model, and intelligent transportation systems.

Zhang Hong is a professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology. His current research interests include information security, data mining and network fault diagnosis.

Appendix

1 Discrete information entropy

Corollary 1. For any *n*-dimensional vectors $P=(p_1,p_2,\cdots,p_n)$ and $Q=(q_1,q_2,\cdots,q_n)$, if $\sum_{i=1}^{n}p_i=1$ and $\sum_{i=1}^{n}q_i=1$, then $H(p_1,p_2,\cdots,p_n)=-\sum_{i=1}^{n}p_i\log_2 p_i\le-\sum_{i=1}^{n}p_i\log_2 q_i$.

Proof:

$$H\left(p_1,p_2,\cdots,p_n\right)+\sum_{i=1}^{n}p_i\log_2 q_i=-\sum_{i=1}^{n}p_i\log_2 p_i+\sum_{i=1}^{n}p_i\log_2 q_i$$
$$=\sum_{i=1}^{n}p_i\log_2\frac{q_i}{p_i}=\log_2 e\cdot\sum_{i=1}^{n}p_i\ln\frac{q_i}{p_i}\le\log_2 e\cdot\sum_{i=1}^{n}p_i\left(\frac{q_i}{p_i}-1\right)\quad\left(\because\ln x\le x-1\right)$$
$$=\log_2 e\cdot\left(\sum_{i=1}^{n}q_i-\sum_{i=1}^{n}p_i\right)=0$$

So $H\left(p_1,p_2,\cdots p_n\right)\le-\sum_{i=1}^{n}p_i\log_2 q_i$.

Corollary 2: If $0<y_2<x_2<x_1<y_1<1$, then $H(x_1,x_2)>H(y_1,y_2)$.

Proof: According to the definition of information entropy, $x_1+x_2=y_1+y_2=1$, if $x_1=y_1-\varepsilon$, then $x_2=y_2+\varepsilon$.

$$H(x_1,x_2)-H(y_1,y_2)=y_1\log_2 y_1+y_2\log_2 y_2-x_1\log_2 x_1-x_2\log_2 x_2$$
$$=y_1\log_2 y_1+y_2\log_2 y_2-(y_1-\varepsilon)\log_2(y_1-\varepsilon)-(y_2+\varepsilon)\log_2(y_2+\varepsilon)$$
$$=y_1\log_2 y_1+y_2\log_2 y_2-y_1\log_2(y_1-\varepsilon)-y_2\log_2(y_2+\varepsilon)+\varepsilon\log_2\frac{y_1-\varepsilon}{y_2+\varepsilon}$$

According to Corollary 1, for 2-dimensional vectors $P=(y_1,y_2)$ and $Q=(y_1-\varepsilon,y_2+\varepsilon)$, then $-y_1\log_2 y_1-y_2\log_2 y_2\le-y_1\log_2(y_1-\varepsilon)-y_2\log_2(y_2+\varepsilon)$.

Moreover $\varepsilon\log_2\frac{y_1-\varepsilon}{y_2+\varepsilon}>0$, hence $H(x_1,x_2)-H(y_1,y_2)>0$.

Corollary 2 shows that when there are only two non-zero information sources in neighbors, the bigger the difference of two information sources is, the smaller the information entropy is; on the contrary, the smaller the difference is, the bigger the information entropy is. When two information sources are same, the information entropy will be the maximum.

Corollary 3: For *(n+1)*-unit information entropy $H(x_1,x_2,\cdots,x_n,x_{n+1})$, where x_{n+1} is the minimum, then $H(x_1,x_2,\cdots,x_n,x_{n+1})>H(x_1+\frac{x_{n+1}}{n},x_2+\frac{x_{n+1}}{n},\cdots,x_n+\frac{x_{n+1}}{n})$.

Proof:

$$H(x_1,x_2,\cdots,x_n,x_{n+1})-H(x_1+\frac{x_{n+1}}{n},x_2+\frac{x_{n+1}}{n},\cdots,x_n+\frac{x_{n+1}}{n})$$
$$=(x_1+\frac{x_{n+1}}{n})\log_2(x_1+\frac{x_{n+1}}{n})+(x_2+\frac{x_{n+1}}{n})\log_2(x_2+\frac{x_{n+1}}{n})+\cdots+(x_n+\frac{x_{n+1}}{n})\log_2(x_n+\frac{x_{n+1}}{n})$$
$$-x_1\log_2 x_1-x_2\log_2 x_2-\cdots-x_n\log_2 x_n-x_{n+1}\log_2 x_{n+1}$$
$$=x_1\left[\log_2(x_1+\frac{x_{n+1}}{n})-\log_2 x_1\right]+x_2\left[\log_2(x_2+\frac{x_{n+1}}{n})-\log_2 x_2\right]+\cdots+$$
$$x_n\left[\log_2(x_n+\frac{x_{n+1}}{n})-\log_2 x_n\right]+\frac{x_{n+1}}{n}\left[\log_2(x_1+\frac{x_{n+1}}{n})-\log_2 x_{n+1}\right]+$$
$$\frac{x_{n+1}}{n}\left[\log_2(x_2+\frac{x_{n+1}}{n})-\log_2 x_{n+1}\right]+\cdots+\frac{x_{n+1}}{n}\left[\log_2(x_n+\frac{x_{n+1}}{n})-\log_2 x_{n+1}\right]$$

Every item is positive, So $H(x_1,x_2,\cdots x_n,x_{n+1})-H(x_1+\frac{x_{n+1}}{n},x_2+\frac{x_{n+1}}{n},\cdots x_n+\frac{x_{n+1}}{n})>0$.

Corollary 4: For *2n*-unit information entropy $H(x_1,x_2,x_3,x_4,\cdots,x_{2n-1},x_{2n})$, it can be expressed as follows:

$$H(x_1,x_2,x_3,x_4,\cdots,x_{2n-1},x_{2n})=H(x_1+x_2,x_3+x_4,\cdots,x_{2n-1}+x_{2n})$$
$$+\sum_{k=0}^{n-1}\left(x_{2k+1}+x_{2k+2}\right)H(\frac{x_{2k+1}}{x_{2k+1}+x_{2k+2}},\frac{x_{2k+2}}{x_{2k+1}+x_{2k+2}})$$

Proof:

$$\text{Right Formula}=-(x_1+x_2)\log_2(x_1+x_2)-(x_3+x_4)\log_2(x_3+x_4)-\cdots$$
$$-(x_{2n-1}+x_{2n})\log_2(x_{2n-1}+x_{2n})-x_1\log_2(\frac{x_1}{x_1+x_2})-x_2\log_2(\frac{x_2}{x_1+x_2})-x_3\log_2(\frac{x_3}{x_3+x_4})$$
$$-x_4\log_2(\frac{x_4}{x_3+x_4})-\cdots-x_{2n-1}\log_2(\frac{x_{2n-1}}{x_{2n-1}+x_{2n}})-x_{2n}\log_2(\frac{x_{2n}}{x_{2n-1}+x_{2n}})$$
$$=-(x_1+x_2)\log_2(x_1+x_2)-(x_3+x_4)\log_2(x_3+x_4)-\cdots-(x_{2n-1}+x_{2n})\log_2(x_{2n-1}+x_{2n})-$$
$$x_1\log_2 x_1+x_1\log_2(x_1+x_2)-x_2\log_2 x_2+x_2\log_2(x_1+x_2)-x_3\log_2 x_3+x_3\log_2(x_3+x_4)-$$
$$x_4\log_2 x_4+x_4\log_2(x_3+x_4)-\cdots-x_{2n-1}\log_2 x_{2n-1}+x_{2n-1}\log_2(x_{2n-1}+x_{2n})-$$
$$x_{2n}\log_2 x_{2n}+x_{2n}\log_2(x_{2n-1}+x_{2n})$$
$$=-x_1\log_2 x_1-x_2\log_2 x_2-x_3\log_2 x_3-x_4\log_2 x_4-\cdots-x_{2n-1}\log_2 x_{2n-1}-x_{2n}\log_2 x_{2n}$$
$$=H(x_1,x_2,x_3,x_4,\cdots x_{2n-1},x_{2n})$$

Corollary 4 show that any *2n*-unit information entropy (*n*>1) can be expressed as the sum of one *n*-unit information entropy and *n* 2-unit information entropy.