Received February 12, 2020, accepted February 28, 2020, date of publication March 3, 2020, date of current version March 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2978143

Trust Routing Protocol Based on Cloud-Based Fuzzy Petri Net and Trust Entropy for Mobile Ad hoc Network

XIAOLIANG WANG, PENG ZHANG, YUYUE DU, AND MEI QI

College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

Corresponding author: Peng Zhang (bigbigroc@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001100.

ABSTRACT The trust-based routing mechanisms are proposed to enhance the security of the mobile ad hoc network (MANET), which use the performance metrics of a node to evaluate the trust value of the node. However, some performance metrics are fuzzy, which are easier to be described qualitatively than to be expressed quantitatively. Therefore, the inability to quantitatively express these performance metrics leads to the inaccuracy in the calculation of the trust values of nodes. Meanwhile, some routing mechanisms add the path with the highest credibility to routing table without considering the hop counts of the route in route selection, which reduces quality of service (QoS) of the routing. Aiming at the above problems, firstly, we use cloud model to deal with the fuzziness of performance metrics. Specifically, a trust reasoning model based on cloud model and fuzzy Petri net (FPN) is presented to evaluate the credibility of nodes. Then we propose a routing algorithm based on trust entropy. Routes with the minimum trust entropy are selected to add to routing table. This routing algorithm can reflect the comprehensive effect of route hops and the trust values of nodes on routing selection, thus improving QoS in MANET. Finally, the TUE-OLSR protocol is established based on the trust entropy routing algorithm and the optimized link state routing (OLSR) protocol. What's more, the effectiveness of TUE-OLSR protocol is verified by simulation experiments, which illustrate that TUE-OLSR protocol performs better than existing trust-based OLSR protocols in terms of packet delivery ratio and average latency.

INDEX TERMS Cloud model, fuzzy Petri net, mobile ad hoc network, trust entropy, TUE-OLSR.

I. INTRODUCTION

The MANET is a self-organized network, where mobile nodes connected by wireless links and multi-hop forwarding without a fixed network infrastructure [1]. Because of its strong flexibility, MANET is mostly used in disaster relief operations, vehicular networks, military service and other fields. However, due to the distributed nature, the constantly dynamic change of network topology and the absence of an absolute control center, MANET is vulnerable to a wide variety of attacks by malicious nodes.

Malicious nodes can change the parameters of routing information and to exhaust the battery of nodes by make them traversing the wrong packet in wrong direction [2]. This type of attack prevents data traffic from being delivered to destinations. We introduce black hole attacks and grey hole attacks to illustrate this problem. In black hole attacks, a malicious node can attract all data packets by falsely claiming a shortest route to the destination and then dumps them all without forwarding them to the destination [3]. A typical variant of black hole attack is grey hole attack, where the malicious node behaves like a normal node during the route discovery process, and then silently drops some or all of the data packets sent to it for further forwarding even when no congestion occurs [4]. Selfishness is another manifestation of malicious nodes. A malicious node refuses to consume its resources such as battery, by not participating in routing operations. Therefore, the security of routing protocol is one of the key points of research.

To solve the above problems, a variety of routing protocols based on security considerations are proposed. According to the ways of preventing malicious attacks from affecting

The associate editor coordinating the review of this manuscript and approving it for publication was Shouguang Wang.

routing operations, these protocols can be classified into two types, security-based routing protocols and trust-based routing protocols. Security-based routing protocols protect routing information based on cryptographic primitives [5]–[7]. But these routing protocols have several drawbacks. Firstly, such protocols require a lot of computing resources, so they are not suitable for mobile devices with limited resources [8]. Secondly, mobile devices can be physically captured and utilized, making the authentication technology based on cryptography invalid, thus normal nodes can become malicious nodes under malicious attacks. Finally, in order to prevent malicious attacks, such protocols are vulnerable to denial of service attacks. Meanwhile, such protocols cannot prevent the bad behaviors of malicious nodes within the network. Therefore, the research on trust-based routing protocols becomes more and more important.

In order to enhance the security of MANET, it is necessary to establish a trust-based routing mechanism [9]. This kind of routing mechanism includes two aspects: trust model and trust-based routing protocol. A malicious attack has its special behavior model, which can be used to identify malicious nodes. On this basis, a trust model is proposed to collect trust factors, which can reflect the behavior and motivation of nodes. The trust model allows nodes to evaluate the credibility of other nodes in the network, so as to find out the malicious nodes which are not allowed to participate in routing operations. Traditional routing protocols select routes with the shortest-path or minimum hop counts, while trust-based routing protocols aim to establish the most trusted routes.

Trust-based routing mechanisms use the performance metrics of a node to evaluate the credibility of the node. However, some performance metrics are fuzzy and random, they are easier to be described qualitatively than to be expressed quantitatively. Specifically, for the trust model based on FPN, we need to collect the truth degree of a series of conditional propositions of the node to calculate the credibility of the node, as shown in Section IV.B. One of the conditional propositions is that the routing operations of the node is normal. We need to judge whether the routing operation of the node is normal according to the number of TC messages sent by the node, and then calculate the truth degree of this proposition. In particular, in order to calculate the truth of this proposition, we tried to set a threshold in the FPN model. We assume that if the number of normal TC messages sent by the node is higher than the threshold, then the routing operation of the node is completely normal, thus the truth degree of the proposition is set to 1. And if the number of normal TC messages sent by the node is lower than the threshold, then the routing operation of the node is completely abnormal, thus the truth degree of the proposition is set to 0. However, we found that when the number of normal TC messages sent by some nodes was lower than the threshold, the routing operation of these nodes was normal. This is because the increase of malicious nodes leads to network congestion, which leads to the loss of TC messages sent by the nodes. In this case, we cannot think that the routing operation of these nodes is completely abnormal. A better expression is to indicate the performance of routing operation of the node according to the number of TC messages sent by the node in a given period as shown in Section IV.C, that is, the performance of routing operation of the node is very poor, poor, good, very good, etc. But in order to calculate the truth degree of this proposition, we need to transform this qualitative description into a quantitative expression. The cloud model can implement the uncertain transformation between a qualitative concept and its quantitative instantiations. Thus in order to make the representation parameters more reliable, we choose the cloud model to synthetically describe the fuzziness of concepts.

Besides, some trust mechanisms in MANET add the path with the highest credibility to the routing table. Since the hop counts of route is not taken into account, the route with large hop counts is generated, which reduces the QoS of the routing.

To solve the above problems, we propose a novel trust-based routing mechanism. Firstly, a trust reasoning model based on cloud model and FPN is presented to evaluate the credibility of a mobile node. Secondly, in order to reflect the comprehensive effect of route hops and the trust values of nodes on routing selection, a routing algorithm based on trust entropy is proposed. This routing algorithm selects the route with the minimum trust entropy. Finally, we extend the OLSR by using the trust entropy routing algorithm, called TUE-OLSR. The simulation experiments have been conducted to present the effectiveness of this new protocol.

The main contributions of this paper are as follows:

(1) Some performance metrics are fuzzy and random. They are easier to be described qualitatively than to be expressed quantitatively, which leads to the inaccuracy in the calculation of the trust values of nodes. We use cloud model to deal with the fuzziness and randomness of performance metrics. What's more, cloud model can implement the uncertain transformation between linguistic concepts and quantitative values.

(2) We propose a routing algorithm based on trust entropy and the trust values of the nodes. Routes with the minimum trust entropy are selected to add to the routing table. Then we extend the OLSR by using the trust entropy routing algorithm, called TUE-OLSR. The simulation experiments have been conducted to present the effectiveness of this new protocol.

The rest of this paper is organized as follows: Section II reviews the related work. Section III describes the related concept of cloud model. Then based on cloud model and FPN, we define a novel FPN, named as Cloud-Based Fuzzy Petri Nets (CFPNs). Section IV introduces the CFPN-based trust reasoning mechanism. Section V presents the trust entropy-based routing mechanism. Simulation results are given in Section VI. Finally, Section VII gives the concluding remarks along with directions for future research.

II. RELATED WORK

A. CLOUD MODEL APPLICATIONS

As a promising tool to describe qualitative concepts, the cloud model has been paid more and more attention and applied in many fields. For example, Gao *et al.* [10] developed a comprehensive assessment method of concrete damage after disastrous fire based on cloud model and game theory. The cloud model was used to calculate the certainty degree of the grading assessment index of concrete damage after fire. Liu and Wen [11] proposed a continuum topology optimization method that can consider the uncertainty of load location. In their work, the cloud model has been employed to depict the uncertainties in the loading locations. In [12], a novel integrated FMEA model based on the cloud model and hierarchical technique was developed to assess and rank the risk of failure modes. Based on the cloud model, Gao *et al.* [13] proposed an intelligent lateral control algorithm, which was designed to calculate intelligent vehicle lateral offsets. Peng and Wang [14] proposed a multicriteria group decision-making method based on the normal cloud model with Zadeh's Z-numbers. In their paper, the normal cloud model has been employed to analyze the Z-number construct. Xu *et al.* [15] put forward a safety assessment method to prevent petrochemical enterprise accidents by proposing a composite safety assessment approach based on the cloud model, preliminary hazard analysis–layer of protection analysis and the bow-tie model. The petrochemical enterprise and its relevant indicators were evaluated based on the cloud model. To implement human knowledge more effectively in the field of human-machine cooperative path planning, a fast human-in-the-loop path planning strategy based on the cloud model was proposed in [16]. In their paper, the cloud model was used to allow human's fuzzy decision about path direction and trending. Wu *et al.* [17] introduced a method based on cloud model and the automatic threshold algorithm for range-constrained thresholding, and used the cloud model to represent various visual properties of the images.

B. FPN IMPROVEMENTS

Petri nets (PNs) are the mathematical representation of parallel discrete systems, which are suitable for describing asynchronous and concurrent system models and have been applied in many fields [18]–[20]. For example, Li *et al.* [21] developed a deadlock prevention method based on structure reuse of Petri net supervisors, which can lead to a nearly optimal for flexible manufacturing systems (FMS). Based on Labelled Petri net, Liu and Jiang [22] proposed the concept of secure bisimulation to solve the problem that the classic bisimulation theory is not suitable for the security-oriented interactive systems. Du *et al.* [23] proposed a web service substitution method based on Petri nets, which can be applied to ecommerce service substitution to meet the business automation needs. FPNs are the modification of classical PNs for dealing with imprecise, vague or fuzzy information in knowledge-based systems. In view of its existing problems, a variety of extended FPN methods have been proposed. Specifically, to precisely express the experience of domain experts, Liu *et al.* [24] presented a linguistic Petri net and a matrix operation-based reasoning algorithm, and the cloud model was used to manage the fuzziness and randomness of knowledge assessments. Chang *et al.* [25] proposed a methodology based on FPN to evaluate the comprehensive risk of deepwater drilling risers, and by using the fuzzy reasoning algorithm based on FPN, risk values of risk factors at different levels and the integrated system were gained by iteration of state matrix. In [26], a hybrid fault location method for smart distribution systems was proposed by using FPN and available multi-source data. Furthermore, a fault diagnosis model based on FPN technique was developed, which employed discrete evidences to estimate the faulted section. Li *et al.* [27] developed a theoretical model based on linguistic interval 2-tuples and interval-valued intuitionistic FPNs for acquiring and representing tacit knowledge, which can be used to increase and sustain the competitive advantages of knowledge intensive organizations. Shi *et al.* [28] presented a novel classical failure mode and effects analysis (FMEA) method based on FPN and fuzzy evidential reasoning to improve the accuracy and reliability of FMEA.

C. TRUST-BASED ROUTING MECHANISMS FOR MANET

Trust-based routing mechanisms include two aspects: First of all, using the trust model we can determine which nodes are trusted and which are not according to the performance of the nodes. Then we need to design the trust-based routing protocol, and establish the routing table composed of trusted nodes based on this protocol for packet transmission. In [29], a trust-based on-demand multipath routing scheme was proposed to find the trust-based secure route from source nodes to destination nodes. Based on the weighted binary relational fuzzy trust model, Jain *et al.* [30] presented an approach of security enhancement in MANETs to mitigate blackhole attacks in ad hoc on-demand distance vector (AODV) protocol. They used a trust computing approach to determine malicious nodes and safe routes in MANETs. In order to embody trust and energy in routing protocol, Sethuraman and Kannan [31] proposed a refined trust energy-ad hoc on demand distance vector (ReTE-AODV) routing algorithm. Bayesian probability was used in their paper for trust management. The proposed algorithm routes the packets from the source nodes to destination nodes not through the shortest route but by selecting a reliable route which consumes low energy for sending the packets. Rajesh *et al.* [32] proposed a subjective logic-based trust model that integrates the behavioral trust with the context-based trust, where the behavioral-based trust incorporates subjective logic-based evidence fusion in indirect trust evaluation. What's more, this model assigns a weight for both behavior and context-based trust to efficiently calculate total trustworthiness of a node. Thorat and Kulkarni [33] proposed the uncertainty analysis framework (UAF) for trust-based routing in MANET. The UAF was integrated into different trust variants of AODV protocol, and used to calculate the network belief, disbelief, and uncertainty (BDU)

values. Wang *et al.* [34] proposed a trust-based QoS routing algorithm and used it to enhance the security of network in the presence of malicious nodes. In their paper, the routing algorithm ensures the forwarding of packets through the trusted and least link delay routes by monitoring the behavior of each node. Once a malicious node is discovered, it is isolated from the network so that no packet is forwarded through it. Zhang *et al.* [35] proposed a novel approach of the distributed and adaptive trust metrics for MANET based on the one-hop module and multi-hop module. In their paper, the one-hop module was used to calculate direct trust and recommendation trust, and the multi-hop module was used to calculate indirect trust. However, some trust mechanisms in MANET add the path with the highest credibility to the routing table. Since the hop counts of route is not taken into account, the route with large hop counts is generated, which reduces the QoS of the routing. Besides, in the trust model based on FPN, there is a proposition about whether routing operation routing operation of the node is normal or not. We need to judge whether the routing operation of the node is normal according to the number of TC messages sent by the node, and then calculate the truth degree of this proposition. Specifically, in order to calculate the truth of this proposition, we tried to set a threshold in the FPN model. We assume that if the number of normal TC messages sent by the node is higher than the threshold, then the routing operation of the node is completely normal, thus the truth degree of the proposition is set to 1. And if the number of normal TC messages sent by the node is lower than the threshold, then the routing operation of the node is completely abnormal, thus the truth degree of the proposition is set to 0. But we found that when the number of normal TC messages sent by the node is lower than the threshold, the routing operation of these nodes was normal. This is because the increase of malicious nodes leads to network congestion, which leads to the loss of TC messages sent by the nodes. In this case, we cannot think that the routing operation of these nodes is completely abnormal. A better expression is to indicate the performance of routing operation of the node according to the number of TC messages sent by the node, that is, the performance of routing operation of the node is very poor, poor, good, very good, etc. To solve these two problems, we propose a novel trust-based routing mechanism. Firstly, a trust reasoning model based on cloud model and FPN is presented to evaluate the credibility of a mobile node. Secondly, in order to reflect the comprehensive effect of route hops and node's trust values on route selection, a routing algorithm based on trust entropy is proposed. This routing algorithm selects the route with the minimum trust entropy. Finally, we extend the OLSR by using the trust entropy routing algorithm, called TUE-OLSR. The simulation experiments have been conducted to present the effectiveness of this new protocol, which show that TUE-OLSR protocol performs better than existing trust-based OLSR protocols in terms of packet delivery ratio and average latency.

Algorithm 1 Algorithm for Calculating the Truth Degrees of Condition Propositions

Input: $load(i)$, $rce(i)$, $fwd(i)$, $ftc(i)$ and $delay(i)$
Output: the truth degrees of condition propositions of node V_i
Setp1: while ($t \leq g$) // Execute this loop during g period
{
if $packet \rightarrow V_i$ //V_i received a packet
$load(i) = load(i) + length(packet)$
$rce(i)++$
end if
if $V_i \rightarrow packet$ //V_i transmitted a packet
$fwd(i)++$
end if
if ($packet \rightarrow V_i$ && $V_i \rightarrow packet$) //V_i has //received and transmitted a packet
$delay(i) = delay(i) + time(g_f) - time(g_r)$
end if
if $V_i \rightarrow$ TC//V_i sent a TC message
$ftc(i)++$
end if
}
Setp2: begin

$$\widetilde{\alpha_1}^{(0)} = ((E_T - E_i)/E_T, (E_T - E_i)/E_T, 0, 0)$$
$$\widetilde{\alpha_2}^{(0)} = (1 - delay(i)/(fwd(i) \cdot s), 1 - delay(i)/(fwd(i) \cdot s), 0, 0)$$
$$\widetilde{\alpha_3}^{(0)} = (fwd(i)/rce(i), fwd(i)/rce(i), 0, 0)$$
$$\widetilde{\alpha_4}^{(0)} = (1 - load(i))/(g \cdot e), 1 - load(i))/(g \cdot e), 0, 0)$$

Switch ($ftc(i)$)
{
case $[g/u] : \widetilde{\alpha_5}^{(0)} = (1, 1, 0.119, 0.007)$;
case $[g/u] - 1 : \widetilde{\alpha_5}^{(0)} = (0.596, 0.596, 0.045, 0.003)$;
case $[g/u] - 2 : \widetilde{\alpha_5}^{(0)} = (0.5, 0.5, 0.028, 0.001)$;
case $[g/u] - 3 : \widetilde{\alpha_5}^{(0)} = (0.405, 0.405, 0.045, 0.003)$;
default: $\widetilde{\alpha_5}^{(0)} = (0, 0, 0.119, 0.007)$;
}

$$\widetilde{\alpha_6}^{(0)} = \widetilde{\alpha_7}^{(0)} = \widetilde{\alpha_8}^{(0)} = \widetilde{\alpha_9}^{(0)} = (0, 0, 0, 0)$$
$$M_0 = (\widetilde{\alpha_1}^{(0)}, \widetilde{\alpha_2}^{(0)}, \ldots, \widetilde{\alpha_9}^{(0)})^T$$

end

III. BASIC CONCEPTS

A. CLOUD MODEL THEORY

The cloud model can synthetically describe the randomness and fuzziness of concepts and implement the uncertain transformation between a qualitative concept and its quantitative instantiations [36]. A cloud $\gamma = (Ex, En, He)$ is described by three numerical characteristics, namely, expectation (Ex), entropy (En) and hyper entropy (He). x is called a cloud

Algorithm 2 Cloud-Based Trust Reasoning Algorithm

Input: *I, O, W, CF and TH* are n × m-dimensional cloud matrices, M_0 is n × 1-dimensional cloud matrix.

Output: M_k is n × 1-dimensional cloud matrix, representing

the truth degrees of all propositions.

Step 1: $k = 1$

The parameter k records the number of iterations.

Step 2: $X_i^{(k)} = W^T M_{(k-1)} \quad (i = 1, 2, 3, \ldots, n)$

Calculate the equivalent input value of each transition.

Step 3:

$$N^{(k)} = [X_1^{(k)}, X_2^{(k)}, \ldots, X_n^{(k)}]_{n\times m}^T \odot O$$
$$Y^{(k)} = (y_{ij}^{(k)})_{nxm} = N^{(k)} \copyright TH$$

$Y^{(k)}$ indicates the enabled transitions of output places. For the kth iteration, the elements in $Y^{(k)}$ are obtained by comparing the input value of the transition with its corresponding threshold value.

Step 4: $Z^{(k)} = Y^{(k)} \odot CF$

Calculate the output certainty factors of the enabled transitions.

Step 5: $Q^{(k)} = Z^{(k)} \otimes X_1^{(k)}$

Calculate the truth degrees of output places of the enabled transitions.

Step 6: $M_k = M_{k-1} \oplus Q^{(k)}$

Calculate the truth degrees of all places.

Step 7: if $M_k \neq M_{k-1}$

$k = k + 1$

jump to Step 2

end if

Step 8: End reasoning.

Algorithm 3 Trust Entropy-Based Routing Algorithm

Input: node V_x and its trust value T_x

Output: route R where the source node is node V_s and the destination node is node V_j

int V_x, $i = 1$

float T_x

$V_x = V_s$// Access the source node V_x at first

$TUE_R = 5.89$// Let the initial trust entropy of route R be //the maximum trust entropy of route R

void search (int V_x)

{

visit (V_x) // Access all nodes that can act as the //intermediate nodes in route R_i

if (! check (R_i, V_x) && $hop(R_i) < 10$) // Function //*hop* is used to calculate the hop counts of the route R_i. If //$V_x$is not in route R_i and the hop counts of route R_i is //smaller than 10, then add node V_x to route R_i

add (R_i, V_x)

if ($V_x = V_j$) // Node $V_x$is the destination node

$i++$

if ($TUE_R > TUE_{R_{i-1}}$) // If the trust entropy of //route R is greater than the trust entropy of route R_{i-1}

$TUE_R = TUE_{R_{i-1}}$// Let the trust entropy of route //R be the trust entropy of route R_{i-1}

$R = R_{i-1}$// Let route R be route R_{i-1}

end if

R_i = delete(R_{i-1}, V_x)// Delete node V_x from //routeR_{i-1} to get route R_i

end if

end if

for(V_w = firstneighbor(V_x); V_w >= 0; V_w = nextneighbor(V_x, V_w))// Search the adjacent points of node V_x. If all of these //points are accessed, then end the for loop

{

search (V_w)

}

delete(R_i, V_x) // Delete V_x from route R_i

}

drop, which is a random instantiation of a qualitative concept. *Ex* denotes the mathematical expectation of a cloud drop belonging to a qualitative concept. *En* is the uncertainty measurement of a qualitative concept. *He* is the uncertainty degree of *En*, which can be regarded as the second-order entropy of *En* [24].

Definition 1 (Interval Cloud): Let U be the universe of discourse and Q be a qualitative concept in U. If $x \in U$ is a random instantiation of concept Q, which satisfies $x \sim N(Ex, En'^2)$ and $En' \sim N(Ex, He^2)$, and the certainty degree of x belonging to concept Q satisfies $y = e^{-\frac{(x-Ex)^2}{2(En)'^2}}$, then the distribution of x on the universe U is said to be a normal cloud. When the expectation Ex is expanded to an interval value $[\underline{Ex}, \overline{Ex}]$, the cloud is called an interval cloud, $\widetilde{\gamma} = ([\underline{Ex}, \overline{Ex}], En, He)$ [24].

Definition 2 (Constant Cloud): To carry out fuzzy reasoning operations on the fuzzy concepts and the definite concepts in the same environment, constant cloud is defined in this paper. For cloud $\widetilde{\gamma} = ([\underline{Ex}, \overline{Ex}], En, He)$, if $\underline{Ex} = \overline{Ex} = a \in [0, 1]$ and $En = He = 0$, then the cloud $\widetilde{\gamma} = ([a, a], 0, 0)$ is called a constant cloud. Constant cloud is the expression of real number in cloud model, and is a special kind of interval cloud.

Definition 3: Given two clouds $\gamma_1 = ([\underline{Ex_1}, \overline{Ex_1}], En_1, He_1)$ and $\gamma_2 = ([\underline{Ex_2}, \overline{Ex_2}], En_2, He_2)$, the arithmetic operations of γ_1 and γ_2 can be summarized as follows [37], [38]:

(1) $\gamma_1 + \gamma_2 = (\underline{Ex_1} + \underline{Ex_2}, \overline{Ex_1} + \overline{Ex_2},$

$$\sqrt{En_1^2 + En_2^2}, \sqrt{He_1^2 + He_2^2});$$

(2) $\gamma_1 - \gamma_2 = (\underline{Ex_1} - \underline{Ex_2}, \overline{Ex_1} - \overline{Ex_2},$

$$\sqrt{En_1^2 + En_2^2}, \sqrt{He_1^2 + He_2^2});$$

(3) $\gamma_1 \times \gamma_2 = (\underline{Ex_1}\,\underline{Ex_2}, \overline{Ex_1}\,\overline{Ex_2}, \sqrt{(En_1Ex_2)^2 + (En_2Ex_1)^2}, \sqrt{(He_1Ex_2)^2 + (He_2Ex_1)^2})$, where $Ex_1 = (\underline{Ex_1} + \overline{Ex_1})/2$ and $Ex_2 = (\underline{Ex_2} + \overline{Ex_2})/2$;
(4) $\lambda\gamma = (\lambda\underline{Ex}, \lambda\overline{Ex}, \sqrt{\lambda}En, \sqrt{\lambda}He)$, $\lambda \geq 0$.

Definition 4 [37]: Given two clouds $\gamma_1 = ([\underline{Ex_1}, \overline{Ex_1}], En_1, He_1)$ and $\gamma_2 = ([\underline{Ex_2}, \overline{Ex_2}], En_2, He_2)$, the possibility degree for the comparison between them can be represented as (1), as shown at the bottom of this page.

Unlike [37], in order to compare constant cloud with interval cloud, we set $s_1 = 1.1 - \frac{\sqrt{En_1^2+He_1^2}}{\sqrt{En_1^2+He_1^2}+\sqrt{En_2^2+He_2^2}}$ and $s_2 = 1.1 - \frac{\sqrt{En_2^2+He_2^2}}{\sqrt{En_1^2+He_1^2}+\sqrt{En_2^2+He_2^2}}$. If $En_1 = En_2 = He_1 = He_2 = 0$, then $\frac{\sqrt{En_1^2+He_1^2}}{\sqrt{En_1^2+He_1^2}+\sqrt{En_2^2+He_2^2}} = \frac{\sqrt{En_2^2+He_2^2}}{\sqrt{En_1^2+He_1^2}+\sqrt{En_2^2+He_2^2}} = 0$.

Theorem 1 [37]: Given two clouds $\gamma_1 = ([\underline{Ex_1}, \overline{Ex_1}], En_1, He_1)$ and $\gamma_2 = ([\underline{Ex_2}, \overline{Ex_2}], En_2, He_2)$, then the followings are true.

(1) $0 \leq p(\gamma_1 \geq \gamma_2) \leq 1$;
(2) $p(\gamma_1 \geq \gamma_2) = 1 \Leftrightarrow s_2\overline{Ex_2} \leq s_1\underline{Ex_1}$;
(3) $p(\gamma_1 \geq \gamma_2) = 0 \Leftrightarrow s_1\overline{Ex_1} \leq s_2\underline{Ex_2}$;
(4) $p(\gamma_1 \geq \gamma_2) \geq 1/2$;
(5) if $s_2\underline{Ex_2} + s_2\overline{Ex_2} \leq s_1\underline{Ex_1} + s_1\overline{Ex_1}$, then $p(\gamma_1 \geq \gamma_2) \geq 1/2$, and especially if $s_2\underline{Ex_2} + s_2\overline{Ex_2} = s_1\underline{Ex_1} + s_1\overline{Ex_1}$, then $p(\gamma_1 \geq \gamma_2) = 1/2$.

Definition 5 (Priority Indice): Assume that there are n clouds $\gamma_i (i = 1, 2, 3, \ldots, n)$, and each cloud γ_i is compared with all clouds $\gamma_i (i = 1, 2, 3, \ldots, n)$ by using Eq. (1). Then, a complementary matrix can be constructed as

$$p = \begin{bmatrix} p_{11} & p_{12} & & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ & & \vdots & \\ p_{n1} & p_{n2} & & p_{nn} \end{bmatrix}.$$

The priority indice of cloud is defined as

$$v_i = \frac{1}{n(n-1)}\left(\sum_{j=1}^{n} p_{ij} + \frac{n}{2} - 1\right).$$

Then we can rank the n clouds $\gamma_i (i = 1, 2, 3, \ldots, n)$ in descending order of the values $v_i (i = 1, 2, 3, \ldots, n)$ [37].

Theorem 2 [37]: Given two clouds $\gamma_1 = ([\underline{Ex_1}, \overline{Ex_1}], En_1, He_1)$ and $\gamma_2 = ([\underline{Ex_2}, \overline{Ex_2}], En_2, He_2)$, whose priority indices are v_1 and v_2, then the followings are true.

(1) if $v_1 \geq v_2$, then $\gamma_1 \geq \gamma_2$;
(2) if $v_1 = v_2$, then $\gamma_1 = \gamma_2$;
(3) if $v_1 \leq v_2$, then $\gamma_1 \leq \gamma_2$;
(4) $\gamma_1 \geq \gamma_2 \Leftrightarrow p(\gamma_1 \geq \gamma_2) \geq p(\gamma_2 \geq \gamma_1)$,
$\gamma_1 = \gamma_2 \Leftrightarrow p(\gamma_1 \geq \gamma_2) = p(\gamma_2 \geq \gamma_1) = 1/2$.

Definition 6 (Cloud Matrix): To implement fuzzy reasoning in the cloud model environment, cloud matrix is defined in this paper. The cloud matrix is composed of n rows and m columns of cloud $\gamma_{ij} (i = 1, 2, 3, \ldots, n; j = 1, 2, 3, \ldots, m)$, then a cloud matrix can be constructed as

$$A = \begin{bmatrix} \gamma_{11} & \gamma_{12} & & \gamma_{1m} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2m} \\ & & \vdots & \\ \gamma_{n1} & \gamma_{n2} & & \gamma_{nm} \end{bmatrix}.$$

Definition 7: To implement the reasoning of the cloud matrix, the product operation of a real number $\lambda (\lambda \geq 0)$ and a cloud matrix A is defined as

$$\lambda A = \begin{bmatrix} \lambda\gamma_{11} & \lambda\gamma_{12} & & \lambda\gamma_{1m} \\ \lambda\gamma_{21} & \lambda\gamma_{22} & \cdots & \lambda\gamma_{2m} \\ & & \vdots & \\ \lambda\gamma_{n1} & \lambda\gamma_{n2} & & \lambda\gamma_{nm} \end{bmatrix}.$$

The operation of $\lambda\gamma_{ij} (i = 1, 2, 3, \ldots, n; j = 1, 2, 3, \ldots, m)$ obeys Definition 3.

Definition 8: Let $A = (a_{ik}) (i = 1, 2, 3, \ldots, n; k = 1, 2, 3, \ldots, s;\ j = 1, 2, 3, \ldots, m)$ be an n × s-dimensional cloud matrix or real matrix, and let $B = (b_{kj})$ be an s × m-dimensional cloud matrix, then the product operation of matrix A and matrix B is defined as $C = (c_{ij})$, where $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{is}b_{sj} = \sum_{k=1}^{s} a_{ik}b_{kj}$, and the operation of $\sum_{k=1}^{s} a_{ik}b_{kj}$ obeys Definition 3.

B. CONVERSION BETWEEN LINGUISTIC TERMS AND CLOUDS

In many situations, linguistic terms are more suitable than precise values in describing the qualitative evaluation information elicited from decision makers [24].

Let $H = \{h_i | i = 0, 1, \ldots, 2t, t \in N^*\}$ be a finite and linguistic term set, where h_i represents a possible value for a linguistic variable, and it should satisfy the following characteristics [35].

(1) The set is ordered: $h_i > h_j$, if $i > j$;
(2) There is the negation operator: $Neg(h_i) = h_j$, where $j = 2t - i$.

For example, a set of seven linguistic terms H can be defined as $H = \{h_0 = extremely\ low, h_1 = very\ low, h_2 = low, h_3 = medium, h_4 = high, h_5 = very\ high, h_6 = extremely\ high\}$, then seven basic clouds can be generated for the corresponding linguistic terms. These clouds can be denoted as: $\gamma_0 = (Ex_0, En_0, He_0)$, $\gamma_1 = (Ex_1, En_1, He_1)$,..., $\gamma_6 =$

$$p(\gamma_1 \geq \gamma_2) = \frac{\min\left\{s_1(\overline{Ex_1} - \underline{Ex_1}) + s_2(\overline{Ex_2} - \underline{Ex_2}), \max(s_1\overline{Ex_1} - s_2\underline{Ex_2}, 0)\right\}}{s_1(\overline{Ex_1} - \underline{Ex_1}) + s_2(\overline{Ex_2} - \underline{Ex_2})}. \quad (1)$$

(Ex_6, En_6, He_6). Using the golden segmentation method [38], their numerical characteristics are computed as:

$$\begin{aligned}
&Ex_0 = X_{min}, Ex_6 = X_{max}, Ex_3 = (X_{max} + X_{min})/2,\\
&Ex_2 = Ex_3 - 0.382(X_{max} + X_{min})/4,\\
&Ex_4 = Ex_3 + 0.382(X_{max} - X_{min})/4,\\
&Ex_1 = Ex_3 - (X_{max} - X_{min})/4,\\
&Ex_5 = Ex_3 + (X_{max} - X_{min})/4;\\
&En_2 = En_4 = 0.382(X_{max} - X_{min})/12,\\
&En_3 = 0.618En_4,\\
&En_1 = En_5 = En_4/0.618, En_0 = En_6 = En_5/0.618;\\
&He_2 = He_4 = He_3/0.618, He_1 = He_5 = He_4/0.618,\\
&He_0 = He_6 = He_5/0.618.
\end{aligned}$$

$[X_{min}, X_{max}]$ is the effective domain of x, and He_3 is given by experience. In this paper, we set $X_{min} = 0, X_{max} = 1$ and $He_3 = 0.001$. Therefore, the above seven basic clouds can be expressed as:

$$\begin{aligned}
&\gamma_0 = (0, 0.084, 0.005), \quad \gamma_1 = (0.25, 0.052, 0.003),\\
&\gamma_2 = (0.405, 0.032, 0.002), \quad \gamma_3 = (0.5, 0.02, 0.001),\\
&\gamma_4 = (0.596, 0.032, 0.002), \gamma_5 = (0.75, 0.052, 0.003) \quad \text{and}\\
&\gamma_6 = (1, 0.084, 0.005).
\end{aligned}$$

Definition 9[39]: Let the uncertain linguistic value be $[h_i, h_j]$. The lower limit h_i and upper limit h_j can be converted into two clouds $\gamma_i = (Ex_i, En_i, He_i)$ and $\gamma_j = (Ex_j, En_j, He_j)$ respectively. Then an interval cloud $\widetilde{\gamma} = ([\underline{Ex}, \overline{Ex},], En, He)$ is obtained, where $\underline{Ex} = \min\{Ex_i, Ex_j\}$, $\overline{Ex} = \max\{Ex_i, Ex_j\}, En = \sqrt{En_i^2 + En_j^2}$ and $He = \sqrt{He_i^2 + He_j^2}$.

C. DEFINITION OF CFPNs

FPNs are a modification of classical Petri nets (PNs) for dealing with imprecise, vague or fuzzy information in knowledge-based systems. The main characteristics of an FPN are that it supports structural organization of information, provides visualization of knowledge reasoning, and facilitates design of efficient fuzzy inference algorithms [40]. The truth degrees of input places are generally given as a series of definite real numbers. However, in some cases the truth degrees of input places are random and fuzzy, they are easier to be described qualitatively than to be expressed quantitatively. The cloud model can synthetically describe the randomness and fuzziness of concepts and implement the uncertain transformation between a qualitative concept and its quantitative instantiations. Therefore, in this paper, we use the cloud model to deal with the fuzziness and randomness of the truth degrees of input places. Furthermore, a new type of FPNs based on cloud model theory is proposed for knowledge representation and reasoning, namely CFPNs.

Definition 10 (CFPN): A CFPN is a 9-tuple: $CFPN = (\widetilde{\alpha}, P, T, M, I, O, W, TH, CF)$, where

(1) $P = \{p_1, p_2, \ldots, p_n\}$ denotes a finite nonempty set of places or propositions.

(2) $\widetilde{\alpha}_i(P_i) : P_i \to \widetilde{\gamma}_i$ is an association function which maps from a place p_i to a cloud $\widetilde{\gamma}_i$. The token value of a place $p_i (p_i \in p)$ is expressed by a cloud $\widetilde{\alpha}_i$ which can represent the truth degree of the place p_i.

(3) $T = \{t_1, t_2, \ldots, t_m\}$ denotes a finite nonempty set of transitions or rules.

(4) $I : P \times T \to (I(p_i, t_j))_{n\times m}$ is an n × m-dimensional input incidence matrix defining the directed arcs from places to transitions $(i = 1, 2, 3, \ldots, n; j = 1, 2, 3, \ldots, m)$.

$$I(p_i, t_j) = \left\{ \begin{array}{l} 1, \ \textit{if there is a directed arc from } p_i \textit{ to } t_j \\ 0, \ \textit{otherwise} \end{array} \right\}$$

(5) $O : T \times P \to (O(t_i, p_j))_{n\times m}$ is an n × m-dimensional ou-tput incidence matrix defining the directed arcs from transitions to places $(i = 1, 2, 3, \ldots, n; j = 1, 2, 3, \ldots, m)$.

$$O(t_j, p_i) = \left\{ \begin{array}{l} 1, \ \textit{if there is a directed arc from } t_j \textit{ to } p_i \\ 0, \ \textit{otherwise} \end{array} \right\}$$

(6) $M = (\widetilde{\alpha_1}, \widetilde{\alpha_2}, \ldots, \widetilde{\alpha_n})^T$ is a making of the CFPN, where $\widetilde{\alpha_i} (i = 1, 2, 3, \ldots, n)$ is a cloud. M indicates the truth degrees of places. Moreover, $M_{(k)} = (\widetilde{\alpha_1}^{(k)}, \widetilde{\alpha_2}^{(k)}, \ldots, \widetilde{\alpha_n}^{(k)})^{\mathrm{T}}$ represents the truth degrees of places after k times of reasoning.

(7) $TH = (\widetilde{\alpha}(\tau_{ij}))_{n\times m}$ denotes the threshold of transition t_j, and the threshold can be represented by cloud $\widetilde{\alpha}(\tau_{ij}) (i = 1, 2, 3, \ldots, n; j = 1, 2, 3, \ldots, m)$.

$$TH_{ij} = \left\{ \begin{array}{ll} \widetilde{\alpha}(\tau_{ij}), & \textit{if there is a directed arc from } t_j \textit{ to } p_i \\ (1, 1, 0, 0), & \textit{otherwise} \end{array} \right\}$$

(8) $W = (w_{ij})_{n\times m} (i = 1, 2, 3, \ldots, n; j = 1, 2, 3, \ldots, m)$ is the weight coefficient of the place p_i, which indicates how much a place p_i, impacts a transition t_j.

$$W(p_i, t_j) = \left\{ \begin{array}{ll} w_{ij}, & \textit{if there is a directed arc from } p_i \textit{ to } t_j \\ 0, & \textit{otherwise} \end{array} \right\}$$

(9) $CF = (\widetilde{\alpha}(\mu_{ij}))_{n\times m}$ is the output certainty factor of the transition t_j, and can be represented by cloud $\widetilde{\alpha}(\mu_{ij})$ $(i = 1, 2, 3, \ldots, n; j = 1, 2, 3, \ldots, m)$.$CF_{ij}$ indicates how much a transition t_j impacts a place p_i.

$$CF_{ij} = \left\{ \begin{array}{ll} \widetilde{\alpha}(\mu_{ij}), & \textit{if there is a directed arc from } t_j \textit{ to } p_i \\ (0, 0, 0, 0), & \textit{otherwise} \end{array} \right\}$$

IV. CFPN-BASED TRUST REASONING MECHANISM

In this section, based on CFPN, we propose a trust reasoning mechanism to deal with the fuzziness and randomness of the truth degrees of propositions and calculate the trust values of propositions. This mechanism has four aspects: cloud-based fuzzy production rules, cloud-based rule representations for MANET, calculation of the truth degrees of condition propositions, and cloud-based trust reasoning algorithm.

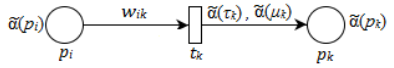


FIGURE 1. CFPR model of Case 1.

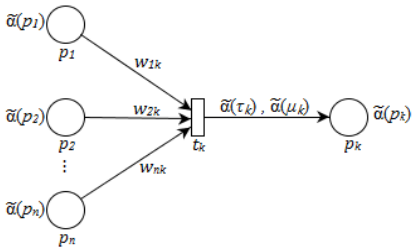


FIGURE 2. CFPR model of Case 2.

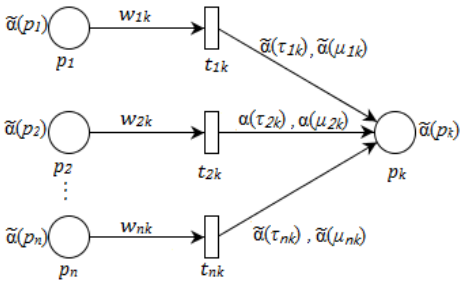


FIGURE 3. CFPR model of Case 3.

A. CLOUD-BASED FUZZY PRODUCTION RULES

Fuzzy production rules (FPRs) are used as a tool of knowledge expression and reasoning for uncertain and fuzzy knowledge. The fuzzy concepts in propositions and rules of FPRs are usually represented by real numbers, but sometimes we have to utilize linguistic terms to state our judgments about knowledge representation parameters (e.g., TC message mentioned in Section I). Therefore, we extend FPRs to the linguistic environment and propose the cloud-based fuzzy production rules (CFPRs).

The formal definition of CFPRs is as follows:

If p_i then $p_k(t_j, \widetilde{\alpha}(p_i), w_{ij}, \widetilde{\alpha}(\tau_{kj}), \widetilde{\alpha}(\mu_{kj}))$

(1) p_i is the antecedence proposition, $i = 1, 2, 3, \ldots, n$;
(2) p_k is the consequence proposition, k is constant;
(3) t_j is the rule of proposition p_i, $j = 1, 2, 3, \ldots, m$;
(4) $\widetilde{\alpha}(p_i)$ is the truth degree of proposition p_i;
(5) $\widetilde{\alpha}(\mu_{kj})$ is the output certainty factor of the rule t_j;
(6) $\widetilde{\alpha}(\tau_{kj})$ is the threshold of rule t_j;
(7) w_{ij} is the weight of proposition p_i.

The CFPRs can be divided into three cases, and all the rules can be represented in accordance with the CFPR model as shown in Figs. 1-3, respectively.

Case 1: A simple CFPR

if p_i then $p_k(t_k, \widetilde{\alpha}(p_i), w_{ik} = 1, \widetilde{\alpha}(\tau_k), \widetilde{\alpha}(\mu_k))$

If $\widetilde{\alpha}(p_i) \geq \widetilde{\alpha}(\tau_k)$, then t_k is fired, the truth degree of consequence proposition p_k can be expressed as

$$\widetilde{\alpha}(p_k) = \widetilde{\alpha}(p_i) \cdot w_{ik} \cdot \widetilde{\alpha}(\mu_k) = \widetilde{\alpha}(p_i) \cdot \widetilde{\alpha}(\mu_k).$$

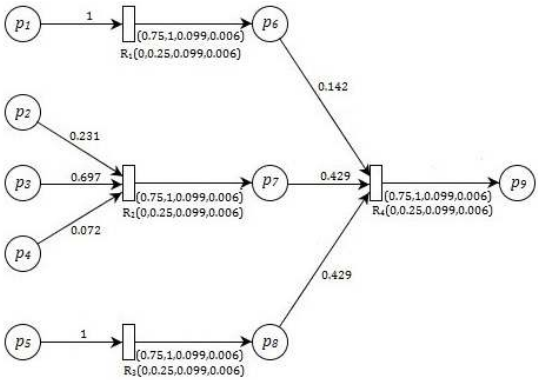


FIGURE 4. Cloud-based rule representations model for MANET.

Case 2: A compound cloud conjunctive rule in the antecedent if p_1 and p_2 and, ..., and p_n then $p_k(t_k, \widetilde{\alpha}(p_i), \sum_{i=0}^{n} w_{ik} = 1, \widetilde{\alpha}(\tau_k), \widetilde{\alpha}(\mu_k))(i = 1, 2, 3, \ldots, n)$

If $\sum_{i=0}^{n} w_{ik} \cdot \widetilde{\alpha}(p_i) \geq \widetilde{\alpha}(\tau_k)$, then t_k is fired, the truth degree of consequence proposition p_k can be expressed as $\widetilde{\alpha}(p_k) = \widetilde{\alpha}(\mu_k) \cdot \sum_{i=0}^{n} w_{ik} \cdot \widetilde{\alpha}(p_i)$.

Case 3: A compound cloud disjunctive rule in the antecedent if p_1 or p_2 or, ..., or p_n then $p_k(t_{ik}, \widetilde{\alpha}(p_i), w_{ik} = 1, \widetilde{\alpha}(\tau_{ik}), \widetilde{\alpha}(\mu_{ik}))(i = 1, 2, 3, \ldots, n)$.

If $\exists w_{ik} \cdot \widetilde{\alpha}(p_i) \geq \widetilde{\alpha}(\tau_{ik})$, then t_{ik} is fired, the truth degree of consequence proposition p_k can be expressed as

$$\widetilde{\alpha}(p_k) = \widetilde{\alpha}(\mu_{ik}) \cdot Max(\widetilde{\alpha}(p_i) \cdot w_{ik}) = \widetilde{\alpha}(\mu_{ik}) \cdot Max(\widetilde{\alpha}(p_i)).$$

B. CLOUD-BASED RULE REPRESENTATIONS FOR MANET

Based on the CFPRs, we build a cloud-based rule representations model for MANET by taking the performance metrics of a node as the condition propositions and the credibility of a node as a conclusion proposition. Specifically, in this subsection, we define 9 propositions and 4 rules, and the cloud-based rule representations model for MANET is shown in Fig. 4 where rule parameters are given by experience. The rules and propositions are as follows:

Rule1: IF p_1 then p_6

p_1 : The node has low energy consumption.

p_6 : The residual energy of the node is high.

Rule2: IF p_2 and p_3 and p_4 then p_7

p_2 : The average packet forwarding delay of the node is low.

p_3 : The packet forwarding rate of the node is high.

p_4 : The load of the node is low.

p_7 : The performance of the node in data plane is normal.

Rule3: IF p_5 then p_8

p_5 : The routing operations of the node are normal.

p_8 : The performance of the node in routing plane is normal.

Rule4: If p_6 and p_7 and p_8 then p_9

p_6 : The residual energy of the node is high.

p_7 : The performance of the node in data plane is normal.

p_8 : The performance of the node in routing plane is normal.

p_9 : The node can be trusted.

In this paper, we use the Analytic Hierarchy Process (AHP) to determine the weight coefficient of a proposition. Moreover, based on the transformation rules between clouds and linguistic terms described in Section III.B, we set $TH_{ij} = \widetilde{\alpha}(\tau_{ij}) = (0, 0.25, 0.099, 0.006)$ if and only if there is a directed arc from t_j to p_i, and $CF_{ij} = \widetilde{\alpha}(\mu_{ij}) = (0.75, 1, 0.099, 0.006)$ if and only if there is a directed arc from t_j to $p_i (i = 1, 2, 3, \ldots, n; j = 1, 2, 3, \ldots, m)$.

C. CALCULATION OF THE TRUTH DEGREES OF CONDITION PROPOSITIONS

In order to implement the trust reasoning algorithm to calculate the credibility of nodes, we have to calculate the truth degrees of condition propositions. In this paper, the truth degrees of condition propositions are determined by the performance of the nodes. Thus, trust factors are defined to evaluate the performance of the nodes, then an algorithm is proposed for calculating the truth degrees of condition propositions.

Definition 10 (Trust Factors): Let g denote the trust update period for updating trust reasoning value.

$load(i)$: the load of the node V_i during g period.

$rce(i)$: number of packets received by the node V_i during g period.

$fwd(i)$: number of packets transmitted by the node V_i during g period.

$ftc(i)$: number of TC messages sent by the node V_i during g period.

$delay(i)$: the forwarding delay of the node V_i during g period.

These trust factors are cleared every g period.

(1) Let the initial energy of node V_i be E_T, the energy consumed of node V_i be E_i, the energy consumed by receiving a packet be E_{rx}, the energy consumed by transmitting a packet be E_{tx}, the receiving power be P_{rx}, the transmitting power be P_{tx}, and the bandwidth be e. The energy consumption calculation method in [41] is adopted in this paper, according to [41], we can get $E_{tx} = P_{tx} \cdot 8 \cdot packetsize/bandwidth = P_{tx} \cdot 8 \cdot packetsize/e$ and $E_{rx} = P_{rx} \cdot 8 \cdot packetsize/bandwidth = P_{rx} \cdot 8 \cdot packetsize/e$. Then we can derive $E_i = rce(i) \cdot E_{rx} + fwd(i) \cdot E_{tx}$. The truth degree of proposition p_1 can be expressed as $(E_T - E_i)/E_T$. In the cloud model, it is expressed as

$$\widetilde{\alpha_1}^{(0)} = ((E_T - E_i)/E_T, (E_T - E_i)/E_T, 0, 0).$$

(2) The packet receiving process is marked as g_r, and the packet transmitting process is marked as g_f. Let the packet forwarding delay of node V_i during g period be $delay(i)$, where $delay\,(i) = delay\,(i) + time\,(g_f) - time\,(g_r)$. The packet forwarding delay of node V_i during g period is expressed as $delay(i)/fwd(i)$, where $fwd(i) \neq 0$. Let delay tolerance be s, the truth degree of proposition p_2 can be expressed as $1 - delay(i)/(fwd(i) \cdot s)$. In the cloud model, it is expressed as

$$\widetilde{\alpha_2}^{(0)} = (1 - delay(i)/(fwd(i) \cdot s), 1 - delay(i)/(fwd(i) \cdot s), 0, 0).$$

(3) If V_i received a packet, then $rce(i)++$. If V_i transmitted a packet, then $fwd(i)++$. The packet forwarding rate of node V_i can be expressed as $fwd(i)/rce(i)$, where $rce(i) \neq 0$. In the cloud model, it is expressed as

$$\widetilde{\alpha_3}^{(0)} = (fwd(i)/rce(i), fwd(i)/rce(i), 0, 0).$$

(4) If V_i received a packet, then $load(i) = load(i) + length(\text{packet})$. $length$ is a function, which is used to compute the data bits in the packet. The load of node V_i can be expressed as $load(i)/g$. The truth degree of proposition p_4 can be expressed as $1 - load(i))/(g \cdot e)$. In the cloud model, it is expressed as

$$\widetilde{\alpha_4}^{(0)} = (1 - load(i)/(g \cdot e), 1 - load(i)/(g \cdot e), 0, 0).$$

(5) If V_i sent a TC message, then $ftc(i) + +$. Let the time interval of TC messages transmission be u. Theoretically, the number of TC messages transmission during g period is $[g/u]([g/u] > 4)$.$[g/u]$ is the largest integer less than $[g/u]$.

If $ftc(i) = [g/u]$, then the truth degree of proposition p_5 is extremely high. In the cloud model, it is expressed as

$$\widetilde{\alpha_5}^{(0)} = (1, 1, 0.119, 0.007).$$

If $ftc(i) = [g/u] - 1$, then the truth degree of proposition p_5 is high. In the cloud model, it is expressed as

$$\widetilde{\alpha_5}^{(0)} = (0.596, 0.596, 0.045, 0.003).$$

If $ftc(i) = [g/u] - 2$, then the truth degree of proposition p_5 is medium. In the cloud model, it is expressed as

$$\widetilde{\alpha_5}^{(0)} = (0.5, 0.5, 0.028, 0.001).$$

If $ftc(i) = [g/u] - 3$, then the truth degree of proposition p_5 is low. In the cloud model, it is expressed as

$$\widetilde{\alpha_5}^{(0)} = (0.405, 0.405, 0.045, 0.003).$$

If $ftc(i) = [g/u] - 4$, then the truth degree of proposition p_5 is extremely low. In the cloud model, it is expressed as

$$\widetilde{\alpha_5}^{(0)} = (0, 0, 0.119, 0.007).$$

In this paper, trust factors are collected by monitoring. We use MPRs to monitor and evaluate their selectors. Since a node has at least one MPR, all nodes can be evaluated.

D. CLOUD-BASED TRUST REASONING ALGORITHM

Based on the calculation method of conditional propositions in Section IV.C, we can calculate the credibility of nodes by using the trust reasoning algorithm in this subsection. In order to formally describe the trust reasoning algorithm, some matrix operators are introduced first.

1) Operator $\oplus$: Let x_{ij}, y_{ij} and z_{ij} be three clouds ($i = 1, 2, 3, \ldots, n$; $j = 1, 2, 3, \ldots, m$). $X \oplus Y = Z$, where $X = (x_{ij})_{n\times m}$, $Y = (y_{ij})_{n\times m}$, $Z = (z_{ij})_{n\times m}$ and $z_{ij} = \max(x_{ij}, y_{ij})$.
2) Operator $\copyright$: Let x_{ij}, y_{ij} and z_{ij} be three clouds ($i = 1, 2, 3, \ldots, n$; $j = 1, 2, 3, \ldots, m$). $X \copyright Y = Z$, where $X = (x_{ij})_{n\times m}$, $Y = (y_{ij})_{n\times m}$ and $Z = (z_{ij})_{n\times m}$. If $x_{ij} \geq y_{ij}$, then $z_{ij} = 1$; otherwise, $z_{ij} = 0$.
3) Operator $\otimes$: Let x_{ik}, y_{kj} and z_{ij} be three clouds ($i = 1, 2, 3, \ldots, n$; $k = 1, 2, 3, \ldots, s$; $j = 1, 2, 3, \ldots, m$). $X \otimes Y = Z$, where $X = (x_{ik})_{n\times k}$, $Y = (y_{kj})_{k\times m}$, $Z = (z_{ij})_{n\times m}$ and $z_{ij} = \max(x_{ik} \cdot y_{kj})$.
4) Operator $\odot$: Let x_{ij}, y_{ij} and z_{ij} be three clouds ($i = 1, 2, 3, \ldots, n$; $j = 1, 2, 3, \ldots, m$). $X \odot Y = Z$, where $X = (x_{ij})_{n\times m}$, $Y = (y_{ij})_{n\times m}$, $Z = (z_{ij})_{n\times m}$ and $z_{ij} = x_{ij} \cdot y_{ij}$.

Let $n = 9$, $m = 4$. The trust reasoning algorithm with matrix operations is described in Algorithm 2.

Now we give an instance of the trust reasoning algorithm. If we monitored the truth degrees of condition propositions of node V_i during g period are as follows: $\widetilde{\alpha}_1^{(0)} = (0.6, 0.6, 0, 0)$, $\widetilde{\alpha}_2^{(0)} = (0.75, 0.75, 0, 0)$, $\widetilde{\alpha}_3^{(0)} = (0.8, 0.8, 0, 0)$, $\widetilde{\alpha}_4^{(0)} = (0.7, 0.7, 0, 0)$ and $\widetilde{\alpha}_5^{(0)} = (1, 1, 0.119, 0.007)$. Then the trust reasoning process is shown below.

According to the above, M_0 can be expressed as

$$M_0 = \begin{bmatrix} (0.6, 0.6, 0, 0) \\ (0.75, 0.75, 0, 0) \\ (0.8, 0.8, 0, 0) \\ (0.7, 0.7, 0, 0) \\ (1, 1, 0.119, 0.007) \\ (0, 0, 0, 0) \\ (0, 0, 0, 0) \\ (0, 0, 0, 0) \\ (0, 0, 0, 0) \end{bmatrix}$$

According to the definition of CFPN and Fig. 4, we can obtain, I, TH, and CF, as shown at the bottom of the next page.

(1) Set $k = 1$, calculate the input value $X_i^{(1)} (i = 1, 2, 3, \ldots, n)$.

$$X_i^{(1)} = W^T M_0 = \begin{bmatrix} (0.6, 0.6, 0, 0) \\ (0.781, 0, 781, 0, 0) \\ (1, 1, 0.119, 0.007) \\ (0, 0, 0, 0) \end{bmatrix}$$

(2) Calculate $N^{(1)}$, as shown at the bottom of the next page.

(3) Calculate $Y^{(1)}$.

$$Y^{(1)} = (y_{ij}^{(1)})_{n\times m} = N^{(1)} \copyright TH = \begin{bmatrix} 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,0 \end{bmatrix}$$

(4) Calculate the output certainty factors of the enabled transitions, $Z^{(1)} = Y^{(1)} \odot CF$, as shown at the bottom of 12 page.

(5) Calculate the truth degrees of output places of the enabled transitions.

$$Q^{(1)} = Z^{(1)} \otimes X_1^{(1)} = \begin{bmatrix} (0, 0, 0, 0) \\ (0, 0, 0, 0) \\ (0, 0, 0, 0) \\ (0, 0, 0, 0) \\ (0, 0, 0, 0) \\ (0.45, 0.6, 0.059, 0.004) \\ (0.586, 0.781, 0.007, 0.005) \\ (0.75, 1, 0.144, 0.009) \\ (0, 0, 0, 0) \end{bmatrix}$$

(6) Calculate the truth degrees of all places.

$$M_1 = M_0 \oplus Q^{(1)} = \begin{bmatrix} (0.6, 0.6, 0, 0) \\ (0.75, 0.75, 0, 0) \\ (0.8, 0.8, 0, 0) \\ (0.7, 0.7, 0, 0) \\ (1, 1, 0.119, 0.007) \\ (0.45, 0.6, 0.059, 0.004) \\ (0.586, 0.781, 0.007, 0.005) \\ (0.75, 1, 0.144, 0.009) \\ (0, 0, 0, 0) \end{bmatrix}$$

(7) Since $M_1 \neq M_0$, let $k = 2$. After the second iteration, we have

$$M_2 = \begin{bmatrix} (0.6, 0.6, 0, 0) \\ (0.75, 0.75, 0, 0) \\ (0.8, 0.8, 0, 0) \\ (0.7, 0.7, 0, 0) \\ (1, 1, 0.119, 0.007) \\ (0.45, 0.6, 0.059, 0.004) \\ (0.586, 0.781, 0.007, 0.005) \\ (0.75, 1, 0.144, 0.009) \\ (0.478, 0.849, 0.112, 0.008) \end{bmatrix}$$

After the third iteration, we get $M_3 = (\widetilde{\alpha}_1^{(3)}, \widetilde{\alpha}_2^{(3)}, \ldots, \widetilde{\alpha}_9^{(3)})^T = M_2$, thus the reasoning progress will stop after three iterations. Then the trust value of node V_i can be expressed as $\widetilde{\alpha}_9^{(3)} = (0.478, 0.849, 0.112, 0.008)$.

V. TRUST ENTROPY-BASED ROUTING MECHANISM

In this section, to solve the problem of high load on some nodes with high credibility, we divide the intervals of the trust values, the nodes belonging to the same interval have the same trust value. Then, based on the concept of trust entropy, we propose a trust entropy-based routing algorithm. Routes with the minimum trust entropy are selected to add to the routing table. This routing algorithm can not only reflect the credibility of nodes, but also take into account the influence of route hops and link load on routing selection.

A. PARTITIONING THE INTERVALS OF THE TRUST VALUES

Some trust mechanisms of routing protocols in MANET add the path with the highest credibility to the routing table. Since the hop counts of route is not taken into account, the route with large hop counts is generated, which reduces QoS of the routing. Fig. 5 illustrates this problem, where network topology is showed and the trust values of nodes are marked. Assume that the source node is node a and the destination node is node h, the traditional trust-based algorithms will select path $a \rightarrow b \rightarrow d \rightarrow e \rightarrow f \rightarrow g \rightarrow h$ as the route between node a and node h. Although this path has the highest reliability among all possible paths between node a and node h, the number of hop counts of this path is one of the largest among these paths. Usually, when the truth values of the nodes are not very different, the hop counts of route play a decisive role in route selection. In this case, the ideal path is path $a \rightarrow i \rightarrow h$, which has the smallest

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad O = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.231 & 0 & 0 \\ 0 & 0.697 & 0 & 0 \\ 0 & 0.072 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.142 \\ 0 & 0 & 0 & 0.429 \\ 0 & 0 & 0 & 0.429 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$TH = \begin{bmatrix} (1,1,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (1,1,0,0) & (1,1,0,0) & (1,1,0,0) & (1,1,0,0) \\ (1,1,0,0) & (1,1,0,0) & (1,1,0,0) & (1,1,0,0) \\ (1,1,0,0) & (1,1,0,0) & (1,1,0,0) & (1,1,0,0) \\ (1,1,0,0) & (1,1,0,0) & (1,1,0,0) & (1,1,0,0) \\ (0,0.25,0.099,0.006) & (1,1,0,0) & (1,1,0,0) & (1,1,0,0) \\ (1,1,0,0) & (0,0.25,0.099,0.006) & (1,1,0,0) & (1,1,0,0) \\ (1,1,0,0) & (1,1,0,0) & (0,0.25,0.099,0.006) & (1,1,0,0) \\ (1,1,0,0) & (1,1,0,0) & (1,1,0,0) & (0,0.25,0.099,0.006) \end{bmatrix}$$

$$CF = \begin{bmatrix} (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0.75,1,0.099,0.006) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0.75,1,0.099,0.006) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0.75,1,0.099,0.006) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0.75,1,0.099,0.006) \end{bmatrix}$$

$$N^{(1)} = [X_1^{(1)}, X_2^{(1)}, \ldots, X_n^{(1)}]_{n\times m}^T \odot O$$

$$= \begin{bmatrix} (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0.6,0.6,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0.781,0.781,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (1,1,0.119,0.007) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \end{bmatrix}$$

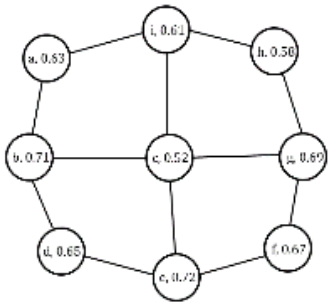


FIGURE 5. Network topology map.

FIGURE 6. Number of forwarded packets.

hop counts of route among all paths between node a and node h.

At the same time, relying on the trust values of the nodes will lead to high load on some nodes with high credibility. As shown in Fig. 6, the arrows represent packets. On the one hand, since the trust value of node a is greater than that of node b, the number of routes that select node a as the intermediate node is relatively large, which increases the load of node a. However, most trust models choose node load and packet loss rate as trust evaluation indexes. Therefore, the credibility of node a will be reduced in these models. Then the route containing node a will be deleted, which will cause the fluctuate of network performance to MANET.

In order not to excessively pursue the high trust values of the nodes in route selection, we classify the trust values of all nodes into different levels according to the differences in these trust values. Specifically, according to the linguistic terms H mentioned in Section III.B, we partition the intervals of the trust values as shown in Eq. (2), the nodes belonging to the same interval have the same trust value. Then we get a new trust value T_x by mapping the trust value of node V_x to Eq. (2). What's more, T_x is given by experience. The larger T_x, the higher the credibility of the node V_x. If $\widetilde{\alpha}_9^{(3)} < (0.25, 0.25, 0.052, 0.003)$, then the reliability of the node is too low to be used as a routing node, (2), as shown at the bottom of this page.

B. TRUST ENTROPY-BASED ROUTING ALGORITHM

In this subsection, we propose a routing algorithm based on trust entropy. Routes with the minimum trust entropy are selected to add to the routing table. This routing algorithm can reflect the comprehensive effect of route hops and node's trust values on route selection. At first, we put forward the concept of trust entropy.

Definition 11 (Trust Entropy): For a route R composed of n nodes, the trust entropy of route R is defined as

$$TUE_R = \sum_{k=1}^{n} (T_k \log_{1/2} T_k^{-1} + 1)(T_k > 0) \tag{3}$$

In the MANET, due to the error and loss of packets in the process of packet forwarding, it is generally considered that it is invalid to deliver packets with more than 10 hops. Therefore, the maximum number of hops for a route is set as 10 in this paper.

Theorem 3: For a route R with no more than 10 hops, the trust entropy of route R is less than or equal to 5.89.

We will prove this theorem in the next subsection. Then the trust routing algorithm based on trust entropy is as follows.

As shown in Fig. 7, assume the source node is node a and the destination node is node h. Access the source node a at first, since node a is not in route R_1, add node a to route R_1. Then access the first adjacency point i of node a.

$$Z^{(1)} = Y^{(1)} \odot CF = \begin{bmatrix} (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0.75,1,0.099,0.006) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0.75,1,0.099,0.006) & (0,0,0,0) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0.75,1,0.099,0.006) & (0,0,0,0) \\ (0,0,0,0) & (0,0,0,0) & (0,0,0,0) & (0,0,0,0) \end{bmatrix}$$

$$T_x = \begin{cases} 0.35 & \widetilde{\alpha}_9^{(3)} \in [(0.75,0.75,0.052,0.003),(1,1,0.084,0.005)] \\ 0.3 & \widetilde{\alpha}_9^{(3)} \in [(0.596,0.596,0.032,0.002),(0.75,0.75,0.052,0.003)] \\ 0.25 & \widetilde{\alpha}_9^{(3)} \in [(0.5,0.5,0.02,0.001),(0.596,0.596,0.032,0.002)] \\ 0.2 & \widetilde{\alpha}_9^{(3)} \in [(0.405,0.405,0.032,0.002),(0.5,0.5,0.02,0.001)] \\ 0.15 & \widetilde{\alpha}_9^{(3)} \in [(0.25,0.25,0.052,0.003),(0.405,0.405,0.032,0.002)] \end{cases} \tag{2}$$

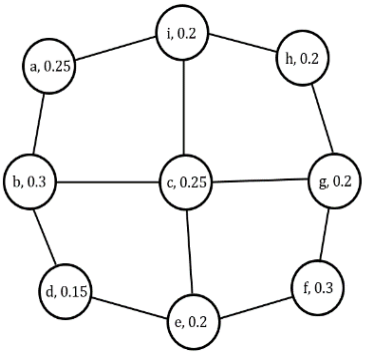


FIGURE 7. Network topology map based on trust entropy.

Because node i is not in route R_1, add node i to route R_1. Access the first adjacency point h of node i, since node h is the destination node, route R_1 completes the routing establishment process. Ultimately route $R_1 = (a, i, h)$ and $TUE_{R_1} = (a, i, h) = 1.572$. Since $TUE_R = 5.89 > TUE_{R_1}$, set $TUE_R = 1.572$ and $R = R_1 = (a, i, h)$. And then set $R_2 = (a, i)$, access the next adjacency point c of i, since node c is not in route R_2, add node c to route R_2. Next, access the first adjacency point g of node c. Because node g is not in route R_2, add node g to route R_2. Then access the first adjacency point h of node g, since node h is the destination node, route R_2 completes the routing establishment process. Thus, route $R_2 = (\mathrm{a}, i, c, g, h)$ and $TUE_{R_2}(a, i, c, g, h) = 2.608$. Since $TUE_R = 1.572 < TUE_{R_2}$, TUE_R and R do not change. Similarly, all routes and their trust entropy are generated according to algorithm 3, as follows:

$$\begin{aligned}
TUE_{R_1}(a, i, h) &= 1.572,\\
TUE_{R_2}(a, i, c, g, h) &= 2.608,\\
TUE_{R_3}(a, i, c, b, d, e, f, g, h) &= 4.691,\\
TUE_{R_4}(a, i, c, e, f, g, h) &= 3.623,\\
TUE_{R_5}(a, b, c, i, h) &= 2.551,\\
TUE_{R_6}(a, b, c, g, h) &= 2.551,\\
TUE_{R_7}(a, b, c, e, f, g, h) &= 3.566,\\
TUE_{R_8}(a, b, d, e, c, g, h) &= 3.676,\\
TUE_{R_9}(a, b, d, e, c, i, h) &= 3.676,\\
TUE_{R_{10}}(a, b, d, e, f, g, h) &= 3.655,\\
TUE_{R_{11}}(a, b, d, e, f, g, c, i, h) &= 4.691.
\end{aligned}$$

Ultimately $TUE_R = \min\left\{TUE_{R_1}, TUE_{R_2}, \ldots, TUE_{R_{11}}\right\} = TUE_{R_1}$ and $R = R_1 = (a, i, h)$. Therefore, add $R = (\mathrm{a}, i, h)$ to the routing table.

C. PROVING THE CORRECTNESS OF THE ROUTING ALGORITHM

Proving the correctness of the proposed routing algorithm is equivalent to proving the following theorems.

Theorem 4: Assume that routeR_a consists of m nodes, where $\sum_{k=1}^{m} T_k = T$. For $\forall V_i, V_j \in R_a$, TUE_{R_a} approaches the minimum value as $T_i - T_j$ approaches zero.

Proof: Denote the multivariable function by $F = \sum_{k=1}^{m}(T_k \log_{1/2} T_k^{-1} + 1)$ and let the constraints be given by $\sum_{k=1}^{m} T_k - T = 0$. The problem is finding the extremum of function$F$ with conditions $\sum_{k=1}^{m} T_k - T = 0$. We first construct the Lagrange Function: $L = \sum_{k=1}^{m}(T_k \log_{1/2} T_k^{-1} + 1) + \lambda(\sum_{k=1}^{m} T_k - T)$and take the partial derivatives of $T_1, T_2, \ldots, T_m$ andλ, then make them be 0, as shown in Eq. (4).

$$\begin{cases}
\dfrac{\partial L}{\partial T_1} = -\log_{1/2} T_1 + 1/\ln 2 + \lambda = 0\\
\dfrac{\partial L}{\partial T_2} = -\log_{1/2} T_2 + 1/\ln 2 + \lambda = 0\\
\qquad\vdots\\
\dfrac{\partial L}{\partial T_m} = -\log_{1/2} T_m + 1/\ln 2 + \lambda = 0\\
\dfrac{\partial L}{\partial \lambda} = \sum_{k=1}^{m} T_k - T = 0
\end{cases} \tag{4}$$

Solve this equation, then we get $T_1 = T_2 = \cdots = T_m = \frac{T}{\mathrm{m}}$. That means when $T_1 = T_2 = \cdots = T_m$, the function F takes the extreme value with conditions $\sum_{k=1}^{m} T_k - T = 0$. Let the extreme value of function F be F_e. Next, we prove that F_e is the minimum value by an example.

To prove that F_e is the minimum value, it is only necessary to prove the following conclusion: For route R_a that consists of m nodes, where $\sum_{k=1}^{m} T_k = T$, let the trust entropy of route R_a be F_a. If $\exists V_i, V_j \in R_a$ and $T_i \neq T_j$, then $F_a > F_e$.

Now we prove this conclusion. Assume $m = 4$ and $T = 1$. Then $F_e = 2$ as $T_1 = T_2 = T_3 = T_4 = 0.25$. For route R_a that consists of 4 nodes, Where $T_1 = 0.2$, $T_2 = 0.3$ and $T_3 = T_4 = 0.25$, $F_a = 2.0145 > F_e = 2$. Thus this theorem is proved. In other words, the more uniform the distribution of node's trust values, the smaller the route's trust entropy. The route has a minimum trust entropy if and only if all nodes in the route have the same trust value.

Theorem 5: Assume that route R_a consists of m nodes and route R_b consists of n nodes. For $\forall V_i \in R_a$ and $\forall V_j \in R_b$, if $T_i = T_j$ and $hop(m) < hop(n)$, then $TUE_{R_a} < TUE_{R_b}$.

Proof: Set $T_i = T_j = T$ and $f(T) = T\log_{1/2} T^{-1} + 1$. Thus $TUE_{R_a} = mf(T)$ and $TUE_{R_b} = nf(T)$. Since $hop(m) < hop(n)$, $m < n$. Since $0 < T \leq 0.35$ and $f(T)$ decreases monotonously in the interval $[0, 0.368]$, $f(T) > f(0.35) > 0$. Thus $mf(T) < nf(T)$, that is, $TUE_{R_a} < TUE_{R_b}$. Therefore, this theorem is proved. In other words, When the trust value distribution of the nodes are uniform, the smaller the hop counts of the route, the smaller the trust entropy of the route.

*Theorem 6:*Assume that route R_a consists of m nodes and route R_b consists of n nodes. For $\forall V_i, V_j \in R_a$ and $\forall V_p, V_q \in R_b$, we have $T_i = T_j$, $T_p = T_q$ and $hop(m) = hop(n)$. If $T_i > T_p$, then $TUE_{R_a} < TUE_{R_b}$.

Proof: Set $T_i = T_j = T_a$ and $T_p = T_q = T_b$. According to this theorem, we get $0 < T_b < T_a \leq 0.35$. Since $f(T) = T\log_{1/2} T^{-1} + 1$ decreases monotonously in the interval $[0, 0.368]$ and $T_b < T_a$, $f(T_b) < f(T_a)$. Since $hop(m) = hop(n)$, $m = n$. Therefore, $mf(T_b) < nf(T_a)$, that is, $TUE_{R_a} < TUE_{R_b}$. Therefore, this theorem is proved. In other words,

Reserved		Htime	Willingness
Link code	Reserved	Link message size	
The Trust Value T_s of the Originator			
Neighbor Interface Address 1			
Neighbor Interface Address 2			
.....			

FIGURE 8. Format of HELLO message.

ANSN	Reserved
Advertised Neighbor Main Address A	
T_A	
Advertised Neighbor Main Address B	
T_B	
.....	

FIGURE 9. Format of TC message.

When the trust value distribution of the nodes are uniform, the higher the trust values of the nodes, the smaller the trust entropy of the route.

In summary, if the trust entropy of the route is small, the trust value distribution of the nodes are uniform, the trust values of the nodes are high and the hop counts of the route is small. In other words, if the hop counts of the route is high and the trust values of the nodes are low, then the trust entropy of the route is high. Thus, for a route R with no more than 10 hops, the trust entropy of route R has the maximum value as $hop(R) = 10$ and $T_x = 0.35$ for all nodes. Then according to Eq. (3), the maximum value of route R can be expressed as $10\left(0.35\log_{1/2}0.35^{-1}+1\right) = 5.89$. Therefore, Theorem 3 is proved.

VI. SIMULATION RESULTS AND ANALYSIS

As is known, HELLO message is used to discover 1-hop and 2-hop neighborhoods. And TC message that contain certain link information is flooded to the entire network. For the TUE-OLSR, where the trust value of each node is collected, the HELLO messages and the TC messages of it need to be modified based on the OLSR. Specifically, in order to obtain the trust values of all nodes in the network, for a network node in MANET, the trust value of this node is added to the HELLO message of it, then the trust values of MPR selectors of this node are added to the TC message of it. The extended HELLO message and TC message are shown in Fig. 8 and Fig. 9, respectively.

To verify the effectiveness of TUE-OLSR, this paper compares TUE-OLSR with OLSR and FPNT-OLSR [9] in terms of the QoS of the routings. FPNT-OLSR is a trust-based routing protocol based on fuzzy Petri net. The idea of FPNT-OLSR is to collect the trust values of the nodes and add the path with the highest credibility to the routing table.

TABLE 1. Fixed simulation parameters.

Parameter	Value
Topology area	1000m× 1000m
Number of nodes	40
Node moving speed	0-10m/s
Transmission radius	250m
Bandwidth	2MHz
Packet size	1024 bytes
Date rate	2pkts/s
Mobility model	VECTOR
Number of malicious nodes	0-10
Simulation time	600s

A. SIMULATION ENVIRONMENT

In this paper, OPNET Modeler 14.5 is used to evaluate the performance of these three routing protocols in different conditions. Our simulations are based on the IEEE 802.11b of MAC layer. Simulation parameters are listed in Table 1. 40 mobile nodes are randomly distributed in a 1000m × 1000m rectangular area, where the nodes move at the speed up to 10m/s. The node mobility uses the VECTOR model and the radio propagation range for each node is 250 meters. The size of packet is 1024 bytes and the network bandwidth is 2MHz. Each simulation executes for 600s of simulation time. Black hole attacks and grey hole attacks are deployed to simulate the environment of malicious attacks.

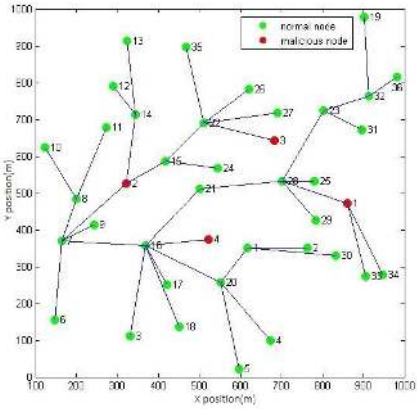
We use three metrics to evaluate the performance of these three routing protocols.

1) Packet delivery ratio: the ratio of the number of received packets to the total number of transmitted packets.
2) Average end-to-end latency: the average time taken by the data packets from source node to destination node.
3) Routing packet overhead: the ratio of the number of control packets to the number of data packets.

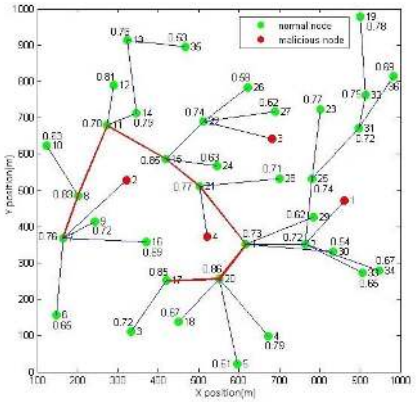
B. TEST 1: ROUTE SELECTION

In this test, we select 40 nodes for simulation experiments, and configure four malicious nodes, two of which simulate black hole attack and two of which simulate grey hole attack. Routing tables of the node 7 and the node 22 are selected to qualitatively analyze the differences of route selection between TUE-OLSR, FPNT-OLSR and OLSR. The simulation results are shown in Fig. 10 and Fig. 11.

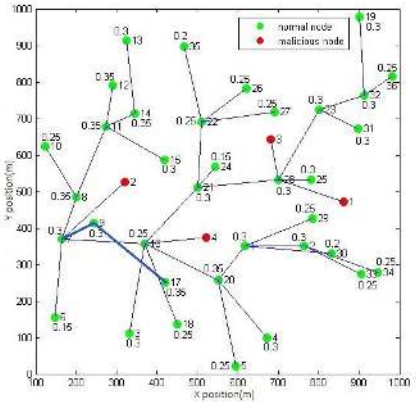
After finishing routing table calculation, we extract details of routes consisting of more than two hops from node 7's routing table, as is shown in Fig. 10. Fig. 10 (a) illustrates that two malicious nodes out of four are selected as intermediate nodes in OLSR. On the contrary, Fig. 10 (b) and (c) demonstrate that FPNT-OLSR and TUE-OLSR prevent all malicious nodes from acting as the intermediate nodes. By taking destination node 17 as an example, we can specifically analyze the differences of route selection between FPNT-OLSR and TUE-OLSR. For FPNT-OLSR, it takes 7 hops for node 7 to send a message to node 17 as shown in the red lines in Fig. 10 (b). However, according to the trust entropy-based routing algorithm we proposed in Section V.B, it takes only 2 hops for



(a) OLSR

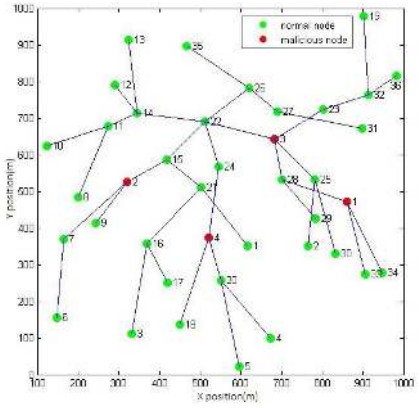


(b) FPNT-OLSR

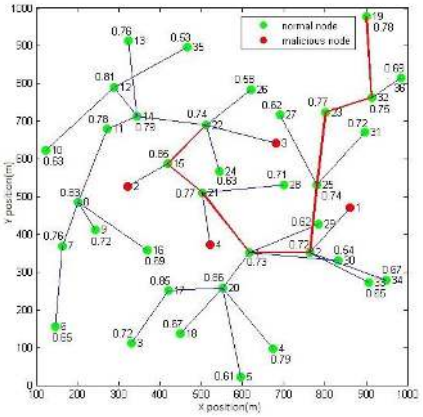


(c) TUE-OLSR

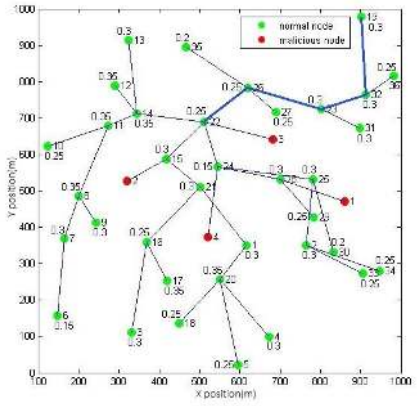
FIGURE 10. Routing table of node 7.



(a)OLSR



(b) FPNT-OLSR



(c) TUE-OLSR

FIGURE 11. Routing table of node 22.

node 7 to send a message to node 17 as shown in the blue lines in Fig. 10 (c). This is because the TUE-OLSR adopts the trust routing algorithm based on trust entropy, which can select routes with small hop counts and high trust values of the nodes. However, the FPNT-OLSR only considers the trust values of the nodes and adds the path with the highest credibility to the routing table, which makes it difficult to choose the route with small hop counts at the same time.

Fig. 11 shows node 22's routes whose distances are more than one hop. Fig. 11(a) indicates that all four malicious nodes are selected as intermediate nodes in OLSR. Whereas, FPNT-OLSR and TUE-OLSR both avoid selecting all malicious nodes, as shown in Fig. 11(b) and (c) respectively. For FPNT-OLSR, it takes 8 hops for node 22 to send messages to node 19 as shown in Fig. 11 (b). However, using our routing algorithm, it only takes 4 hops for node 22 to send messages to node 19 as shown in Fig. 11(c). What's more, as shown in Fig. 11(c), the trust values of the intermediate nodes 26, 23 and 32 are represented as 0.25, 0.3 and 0.3 respectively. According to the rating criteria for the trust values as shown in Eq. (2), these trust values are relatively high.

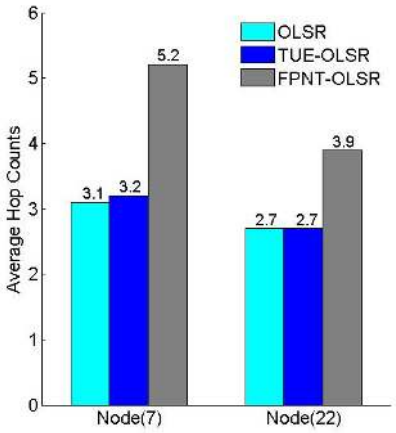


FIGURE 12. The average hop counts of the route.

TABLE 2. Varying simulation parameters.

Test no.	Malicious nodes	Max speed
2	5	0-10m/s
3	0-10	2m/s

Fig. 12 analyzes the average hop counts required by OLSR, FPNT-OLSR and TUE-OLSR to send messages to other nodes in the network. It indicates that the average hops generated by TUE-OLSR and OLSR are almost the same, while the average hops generated by FPNT-OLSR is higher than the former two protocols.

Fig. 10, 11 and 12 indicate that TUE-OLSR can effectively prevent malicious nodes from forwarding messages as intermediate nodes. What's more, this protocol can reflect the comprehensive effect of route hops and node's trust values on route selection, where the hop counts of the routes are small and the trust values of the intermediate nodes are relatively high.

To quantitatively test the performance of these three protocols, we configure varying simulation parameters and simulate the following tests under different conditions as shown in Table 2.

C. TEST 2: VARYING NODE SPEEDS

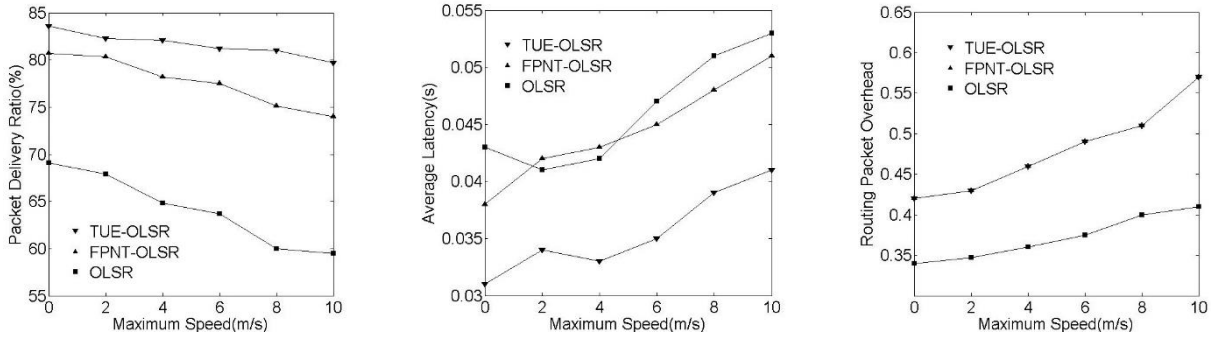
This test compares the performance of OLSR, FPNT-OLSR and TUE-OLSR with varying node speeds. Fig. 13 (a) illustrates that the packet delivery ratios of the three routing protocols all decreases with the increase of the moving speed of nodes. The packet delivery ratios of OLSR decreases significantly, while the packet delivery ratios of TUE-OLSR and FPNT-OLSR decreases steadily. This is attributed to the trust mechanism adopted by TUE-OLSR and FPNT-OLSR, which can add nodes with high packet delivery ratios to the route, so as to prevent malicious nodes or nodes with poor performance of service to forward packets. However, OLSR selects the route with the least number of hop counts without considering the performance of the nodes, which causes malicious nodes or nodes with poor performance of service to forward packets as intermediate nodes. The packet delivery ratios of TUE-OLSR is higher than that of FPNT-OLSR, which is due to the trust entropy-based routing algorithm adopted by TUE-OLSR. The FPNT-OLSR relies on the credibility of the nodes when choosing the path, resulting in high load on some nodes with high credibility. However, FPNT-OLSR chooses node load as trust evaluation index. Therefore, the credibility of these nodes will be reduced. Then the routes containing these nodes will be deleted, which makes the frequency of link broken of FPNT-OLSR higher than that of TUE-OLSR. Therefore, the routing stability of FPNT-OLSR is lower than that of TUE-OLSR, which causes FPNT-OLSR to lose more packets compared with FPNT-OLSR.

For the three protocols, the average end-to-end latency all rises with the increase of node speeds as shown in Fig. 13 (b). This is because the route break down easily as the nodes speed up. Thus, the source nodes have to initiate more route rediscoveries before sending packets, which increases the average end-to-end latency of these three protocols. Compared with the other two protocols, the average end-to-end delay of TUE-OLSR is relatively low. The reasons are as follows. Since OLSR is unable to monitor the performance of nodes, malicious nodes or nodes with poor performance of service are added to the route, which increases the average end-to-end delay. Meanwhile, FPNT-OLSR does not consider the number of hop counts of the route in route selection, and chooses the route with large hop counts, which leads to the increase of average end-to-end delay. On the contrary, TUE-OLSR can reflect the comprehensive effect of route hops and node's trust values on route selection. Specifically, based on the trust routing algorithm we proposed, the nodes with high trust values are added to the route and then the routes with small hop counts are generated. Therefore, the average end-to-end delay of TUE-OLSR is the lowest among the three protocols.

The routing packet overhead of the three protocols all in- creases with the increase of node speeds as shown in Fig. 13 (c). This is because the faster the node moves, the easier the route will break down, and route reconstructions will generate control packets, which will increase the routing packet overhead. TUE-OLSR and FPNT-OLSR have the same routing packet overhead due to the same trust broadcast mechanism adopted by these two protocols. Moreover, the routing overhead of TUE-OLSR and FPNT-OLSR is higher than that of OLSR due to the need to broadcast the trust value of the node.

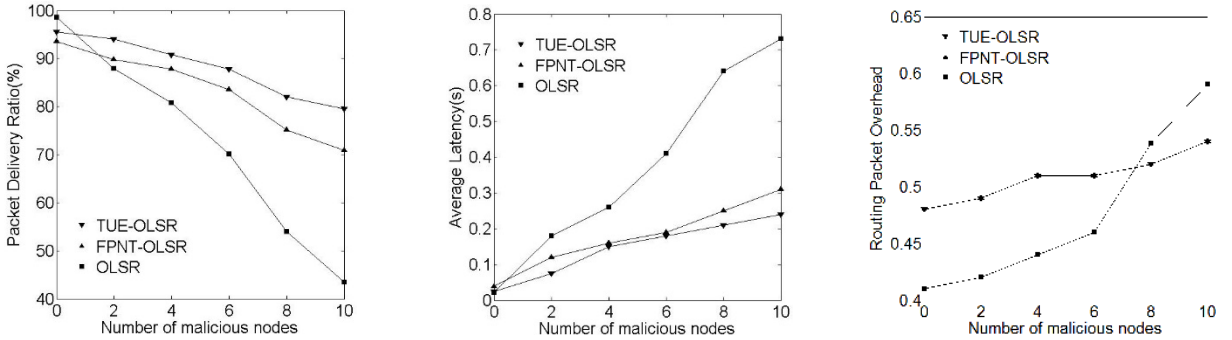
D. TEST 3: VARYING NUMBER OF MALICIOUS NODES

This test compares the performance of OLSR, FPNT-OLSR and TUE-OLSR with varying number of malicious nodes. Fig. 14 (a) shows that the packet delivery ratios of the three routing protocols decreases significantly with the increase of the number of malicious nodes. Since it is impossible to prevent malicious nodes to forward packets as intermediate



(a) Packet Delivery Ratio (b) Average Latency (c) Routing Packet Overhead

FIGURE 13. Performance comparison with varying node speed.



(a) Packet Delivery Ratio (b) Average Latency (c) Routing Packet Overhead

FIGURE 14. Performance comparison with varying number of malicious nodes.

nodes, the packet delivery ratios of OLSR decreases the most. With the increase of the number of malicious nodes, TUE-OLSR and FPNT-OLSR cannot completely stop malicious nodes forwarding packets as intermediate nodes, which leads to a significant decrease in the packet delivery ratios of these two protocols. As mentioned in the first paragraph of Section VI.C, since the routing stability of TUE-OLSR is better than that of FPNT-OLSR, the packet delivery ratios of TUE-OLSR decreases the least among the three protocols.

With the increase of the number of malicious nodes, the average end-to-end delay of the three routing protocols all ascends, as shown in Fig. 14 (b). Due to the lack of the consideration of the malicious nodes in route selection, the average end-to-end delay of OLSR increases sharply. TUE-OLSR and FPNT-OLSR can prevent malicious nodes or nodes with poor performance of service to forward packets, thus the average end-to-end delay of these two protocols is lower than that of OLSR. The average end-to-end delay of TUE-OLSR is lower than that of FPNT-OLSR, because the route hops selected by TUE-OLSR are small. The smaller the hop counts of the routes, the lower the end-to-end delay of the protocol.

Fig. 14 (c) illustrates that the routing overhead of the three routing protocols increases with the increase of the number of malicious nodes. TUE-OLSR and FPNT-OLSR have the same routing packet overhead, which is due to the same trust broadcast mechanism adopted by these two protocols. When the number of malicious nodes in the network is smaller than 7, the routing packet overhead of TUE-OLSR and FPNT-OLSR is larger than that of OLSR. This is because TUE-OLSR and FPNT-OLSR need to broadcast the trust values of nodes to the network, which increases the routing packet overhead. However, when the number of malicious nodes is bigger than 8, the routing packet overhead of OLSR is larger than that of TUE-OLSR and FPNT-OLSR. This is because OLSR cannot prevent malicious nodes or nodes with poor performance of service to forward packets as intermediate nodes. Therefore, more and more malicious nodes are added to the route, resulting in a large routing packet overhead.

VII. CONCLUSION

MANETs are self-organized network without an absolute control center, which makes them vulnerable to a variety of attacks. To improve the security of MANET, the trust-based routing protocols are proposed. In order to solve the problems that FPNT-OLSR routing protocols can not accurately express the truth degree of the proposition of routing operation, and choose routes with a large number of hop counts, we put forward a series of improvement measures. Firstly, we establish a CFPN-based trust reasoning mechanism based

on cloud model and fuzzy Petri net to calculate the reliability of nodes. Concretely speaking, in order to accurately express the truth degree of the propositions, we use cloud model to deal with the fuzziness of the propositions. To calculate the initial truth degrees of propositions, we define the concept of trust factors and calculate these truth degrees using trust factors. Secondly, we propose the concept of trust entropy and design a trust entropy-based routing algorithm which can select routes with high reliability and small number of hop counts. Finally, according to CFPN-based trust reasoning mechanism and trust entropy-based routing algorithm, we establish TUE-OLSR routing protocol based on OLSR protocol. The simulation results show that TUE-OLSR routing protocol performs better than the FPN-OLSR and the OLSR protocols in terms of average delay and packet delivery ratio.

In future work, more effective trust factors can be added to the rule expressions model of MANET to improve the accuracy of fuzzy reasoning.

REFERENCES

[1] H. Xia, Z. Jia, L. Ju, X. Li, and E. H.-M. Sha, "Impact of trust model on on-demand multi-path routing in mobile ad hoc networks," *Comput. Commun.*, vol. 36, no. 9, pp. 1078–1093, May 2013.

[2] N. Sharma and A. Sharma, "The black-hole node attack in MANET," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Technol.*, Rohtak, India, Jan. 2012, pp. 546–550.

[3] M. Mohanapriya and I. Krishnamurthi, "Modified DSR protocol for detection and removal of selective black hole attack in MANET," *Comput. Electr. Eng.*, vol. 40, no. 2, pp. 530–538, Feb. 2014.

[4] G. S. Bindra, A. Kapoor, A. Narang, and A. Agrawal, "Detection and removal of co-operative blackhole and grayhole attacks in MANETs," in *Proc. Int. Conf. Syst. Eng. Technol. (ICSET)*, Bandung, Indonesia, Sep. 2012, pp. 1–5.

[5] Y. Yang, "Broadcast encryption based non-interactive key distribution in MANETs," *J. Comput. Syst. Sci.*, vol. 80, no. 3, pp. 533–545, May 2014.

[6] E. S. Babu, C. Nagaraju, and M. K. Prasad, "Efficient DNA-based cryptographic mechanism to defend and detect blackhole attack in MANETs," in *Proc. Int. Conf. ICT Sustain. Develop.*, Panaji, India, 2016, pp. 695–706.

[7] O. Singh, J. Singh, and R. Singh, "Multi-level trust based intelligence intrusion detection system to detect the malicious nodes using elliptic curve cryptography in MANET," *Cluster Comput.*, vol. 21, no. 1, pp. 51–63, Mar. 2018.

[8] H. Xia, J. Yu, C.-L. Tian, Z.-K. Pan, and E. Sha, "Light-weight trust-enhanced on-demand multi-path routing in mobile ad hoc networks," *J. Netw. Comput. Appl.*, vol. 62, pp. 112–127, Feb. 2016.

[9] S. Tan, X. Li, and Q. Dong, "Trust based routing mechanism for securing OSLR-based MANET," *Ad Hoc Netw.*, vol. 30, pp. 84–98, Jul. 2015.

[10] Z. Gao, D. Ma, X. Guo, W. Wang, and Z. Wang, "The comprehensive assessment method of concrete damage after disastrous fire based on game theory-normal cloud model," *Math. Problems Eng.*, vol. 2019, pp. 1–9, Feb. 2019.

[11] J. Liu and G. Wen, "Continuum topology optimization considering uncertainties in load locations based on the cloud model," *Eng. Optim.*, vol. 50, no. 6, pp. 1041–1060, Aug. 2017.

[12] H.-C. Liu, L.-E. Wang, Z. Li, and Y.-P. Hu, "Improving risk evaluation in FMEA with cloud model and hierarchical TOPSIS method," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 1, pp. 84–95, Jan. 2019.

[13] H. Gao, X. Zhang, Y. Liu, and D. Li, "Cloud model approach for lateral control of intelligent vehicle systems," *Sci. Program.*, vol. 2016, pp. 1–12, Sep. 2016.

[14] H.-G. Peng and J.-Q. Wang, "A multicriteria group decision-making method based on the normal cloud model with Zadeh's z -Numbers," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3246–3260, Dec. 2018.

[15] Q. Xu, K. Xu, L. Li, and X. Yao, "Safety assessment of petrochemical enterprise using the cloud model, PHA–LOPA and the bow-tie model," *Roy. Soc. Open Sci.*, vol. 5, no. 7, Jul. 2018, Art. no. 180212.

[16] X. Sun, C. Cai, and X. Shen, "A new cloud model based human-machine cooperative path planning method," *J. Intell. Robotic Syst.*, vol. 79, no. 1, pp. 3–19, Aug. 2014.

[17] T. Wu, J. Xiao, K. Qin, and Y. Chen, "Cloud model-based method for range-constrained thresholding," *Comput. Electr. Eng.*, vol. 42, pp. 33–48, Feb. 2015.

[18] X. Guo, S. Wang, D. You, Z. Li, and X. Jiang, "A siphon-based deadlock prevention strategy for S^3PR," *IEEE Access*, vol. 7, pp. 86863–86873, 2019.

[19] W. Duo, X. Jiang, O. Karoui, X. Guo, D. You, S. Wang, and Y. Ruan, "A deadlock prevention policy for a class of multithreaded software," *IEEE Access*, vol. 8, pp. 16676–16688, 2020.

[20] S. Wang, D. You, and M. Zhou, "A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in s 4PR," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4173–4179, Aug. 2017.

[21] Z. Li, G. Liu, H.-M. Hanisch, and M. Zhou, "Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 1, pp. 178–191, Jan. 2012.

[22] G. Liu and C. Jiang, "Behavioral equivalence of security-oriented interactive systems," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 8, pp. 2061–2068, Aug. 2016.

[23] Y. Du, J. Gai, and M. Zhou, "A Web service substitution method based on service cluster nets," *Enterprise Inf. Syst.*, vol. 11, no. 10, pp. 1535–1551, Apr. 2016.

[24] H.-C. Liu, X. Luan, Z. Li, and J. Wu, "Linguistic Petri nets based on cloud model theory for knowledge representation and reasoning," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 4, pp. 717–728, Apr. 2018.

[25] Y. Chang, X. Wu, G. Chen, J. Ye, B. Chen, L. Xu, J. Zhou, Z. Yin, and K. Ren, "Comprehensive risk assessment of deepwater drilling riser using fuzzy Petri net model," *Process Saf. Environ. Protection*, vol. 117, pp. 483–497, Jul. 2018.

[26] I. Kiaei and S. Lotfifard, "Fault section identification in smart distribution systems using multi-source data based on fuzzy Petri nets," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 74–83, Jan. 2020.

[27] H. Li, J.-X. You, H.-C. Liu, and G. Tian, "Acquiring and sharing tacit knowledge based on interval 2-Tuple linguistic assessments and extended fuzzy Petri nets," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 26, no. 1, pp. 43–65, Jan. 2018.

[28] H. Shi, L. Wang, X.-Y. Li, and H.-C. Liu, "A novel method for failure mode and effects analysis using fuzzy evidential reasoning and fuzzy Petri nets," *J. Ambient Intell. Humanized Comput.*, pp. 1–15, Mar. 2019.

[29] G. M. Borkar and A. R. Mahajan, "A secure and trust based on-demand multipath routing scheme for self-organized mobile ad-hoc networks," *Wireless Netw.*, vol. 23, no. 8, pp. 2455–2472, Nov. 2017.

[30] A. K. Jain, V. Tokekar, and S. Shrivastava, "Security enhancement in MANETs using fuzzy-based trust computation against black hole attacks," in *Proc. 2nd Int. Congr. Inf. Commun. Technol.*, Bangkok, Thailand, 2016, pp. 39–47.

[31] P. Sethuraman and N. Kannan, "Refined trust energy-ad hoc on demand distance vector (ReTE-AODV) routing algorithm for secured routing in MANET," *Wireless Netw.*, vol. 23, no. 7, pp. 2227–2237, May 2016.

[32] A. Rajesh, V. Raji, and N. M. Kumar, "Subjective logic based trust model for geographic routing in mobile ad hoc networks," *Tehnicki Vjesnik-Tech. Gazette.*, vol. 23, no. 5, pp. 1357–1364, Oct. 2016.

[33] S. A. Thorat and P. J. Kulkarni, "Uncertainty analysis framework for trust based routing in MANET," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 4, pp. 1101–1111, Jul. 2017.

[34] B. Wang, X. Chen, and W. Chang, "A light-weight trust-based QoS routing algorithm for ad hoc networks," *Pervas. Mobile Comput.*, vol. 13, pp. 164–180, Aug. 2014.

[35] D. Zhang, J. Gao, X. Liu, T. Zhang, and D. Zhao, "Novel approach of distributed & adaptive trust metrics for MANET," *Wireless Netw.*, vol. 25, no. 6, pp. 3587–3603, Aug. 2019.

[36] D. Li, C. Liu, and W. Gan, "A new cognitive model: Cloud model," *Int. J. Intell. Syst.*, vol. 24, no. 3, pp. 357–375, Mar. 2009.

[37] J.-Q. Wang, P. Wang, J. Wang, H.-Y. Zhang, and X.-H. Chen, "Atanassov's interval-valued intuitionistic linguistic multicriteria group decision-making method based on the trapezium cloud model," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 3, pp. 542–554, Jun. 2015.

[38] C. Zhu, L. Zhu, and X. Zhang, "Linguistic hesitant fuzzy power aggregation operators and their applications in multiple attribute decision-making," *Inf. Sci.*, vols. 367–368, pp. 809–826, Nov. 2016.

[39] H. Shi, H.-C. Liu, P. Li, and X.-G. Xu, "An integrated decision making approach for assessing healthcare waste treatment technologies from a multiple stakeholder," *Waste Manage.*, vol. 59, pp. 508–517, Jan. 2017.
[40] H.-C. Liu, J.-X. You, Z. Li, and G. Tian, "Fuzzy Petri nets for knowledge representation and reasoning: A literature review," *Eng. Appl. Artif. Intell.*, vol. 60, pp. 45–56, Apr. 2017.
[41] M. Yuan, L. Zhang, and Y. Shu, "An analysis of energy consumption for ad hoc network routing protocols," *Comput. Eng. Appl.*, vol. 39, 2003.

XIAOLIANG WANG received the B.S. degree from the Shandong University of Science and Technology, Qingdao, China, in 2016, where he is currently pursuing the M.S. degree with the College of Computer Science and Engineering. His current research interests are fuzzy Petri nets, trust management, and mobile ad hoc network routing protocol.

PENG ZHANG received the B.S. and M.S. degrees from the Shandong University of Science and Technology, Qingdao, China, in 1997 and 2000, respectively, and the Ph.D. degree in computer application from Tongji University, Shanghai, China, in 2006. He is currently an Associate Professor with the Shandong University of Science and Technology, Qingdao, China. His research interests are in formal engineering methods, Petri nets, and web services. He has participated in many projects supported by the National Natural Science Foundation, the National Key Basic Research Development Program, and other important and key projects at provincial levels. He has authored over 20 papers in journals and conference proceedings, such as *Acta Electronica Sinica* and the *Journal of Computers*. He was a recipient of the Science and Technology Advancement Awards at the province level.

YUYUE DU received the B.S. degree from Shandong University, Jinan, China, in 1982, the M.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1991, and the Ph.D. degree in computer application from Tongji University, Shanghai, China, in 2003. He is currently a Professor with the Shandong University of Science and Technology, Qingdao, China. He has authored over 170 papers in journals and conference proceedings, such as the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART A, the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART C, *Information Sciences*, and *Enterprise Information Systems*. He has led or participated in over 15 projects supported by the National Natural Science Foundation, the National Key Basic Research Development Program, and other important and key projects at provincial levels. His research interests are in formal engineering methods, Petri nets, real-time systems, web services, and workflows. He is a member of the Professional Committee of Petri Nets of the China Computer Federation. He was a recipient of two prizes of science and technology advancement awards at the province level and the Excellent Doctoral Dissertation Award in Shanghai, China.

MEI QI received the B.S. and M.S. degrees from the Shandong University of Science and Technology, Qingdao, China, in 1998 and 2005, respectively. Her current research interests are in Petri nets, workflow and computer architecture.