2011 IEEE/ACM International Conference on Green Computing and Communications

A Subjective Trust Management Model with Multiple Decision Factors for MANET based on AHP and Fuzzy Logic Rules

Hui Xia[1], *Zhiping Jia*[1], *Lei Ju*[1], *Xin Li*[1], *Youqin Zhu*[2]
(*1. School of Computer Science and Technology, Shandong University, Jinan, China, 250101;*
2. Dazhong News Group, Jinan, China, 250101)
(e-mail: sprit_xiahui@mail.sdu.edu.cn)

Abstract — **A mobile ad hoc network (MANET) is a self-organized system comprised by multiple mobile wireless nodes. Due to the openness in network topology and the absence of centralized administration in management, MANET is vulnerable to attacks from malicious nodes. In order to reduce the hazards from these malicious nodes, we incorporate the concept of trust into the MANET, and build a subjective trust management model with multiple decision factors based on the analytic hierarchy process (AHP) theory and the fuzzy logic rules prediction method – AFStrust. We consider multiple decision factors, including direct trust, recommendation trust, incentive function and active degree, in our model to reflect trust relationship's complexity and uncertainty from various aspects. It overcomes the shortage of traditional method, where the decision factors are incomplete. Moreover, the weight of classification is set up by AHP for these decision factors, which makes the model has a better rationality and a higher practicability. Compared to the existing trust management models, comprehensive experiments have been conducted to evaluate the efficiency of our trust management model in the improvement of network interaction quality, trust dynamic adaptability, malicious node identification, attack resistance and enhancements of system's security.**

Key Words: MANET; Trust Management Model; Decision Factors; Analytic Hierarchy Process; Fuzzy Logic Rules

I. INTRODUCTION

A mobile ad hoc network (MANET) is a self-organized multi-hop system comprised by multiple mobile wireless nodes with peer-to-peer relationships. The nodes in the network cannot communicate with each other via well-established infrastructure. Due to the limitation of energy, two peers out of communication range require intermediate nodes to transfer messages. Therefore, a node in this network serves as a host and a router simultaneously. Each node is assumed to relay packets for other nodes, and it works well only if the nodes in the network behave cooperatively. Due to the openness in network topology, MANET often suffers from attacks by selfish or malicious nodes, such as the on-off attack, bad-mouthing attack, conflict behavior attack, packet dropping (black-hole) attack, selective forwarding (gray-hole) attack and so on [1]. Existing security technologies are mostly based on encryption and authentication, which are unsuitable in the dynamic network topology without a trusted third-party. Moreover, the traditional cryptosystem based security mechanism is typically used to resist the external attacks. They show inefficiency in handling the attacks from the internal malicious nodes which may lead to serious influence on the security, the confidentiality, and the life cycle of the whole network.

Trust management mechanism is considered to be an effective measurement to solve these problems [2]. In the context of MANET, there are several trust management models that have been proposed in the realm of network (e.g., [3, 4, 5, 6]), where trust can be considered as the reliance of a network node on the ability to forward packets or offer services timely, integrally and reliably. In the existing models, decision factors are often incomplete in the trust derivation, which are not fully integrated with the inherent characteristics of MANET. When the factors of decision-making are given, though we know that different factors have different weights, the precise weights are difficult to determine. Existing methods in these models for weight determination are lack of rationality and practicability. As a result, they cannot calculate an accurate trust value for each node. Hence, these models are ineffective in MANET trust management, and their applications are very simple.

To address those questions, in this paper, we establish a new subjective trust management model for MANET considering the behaviors of the dynamic nodes in the open environment and the complete decision factors of nodes' trustworthiness. The nodes' trust values can be easily used in trust management strategy, which includes the applications anti-attack, decision making etc. The motivations of our work are to (*a*) obtain a more accurate node's trust value; (*b*) improve the quality of network interaction, increase the proportion number of good recommendation, raise the malicious node's correct detection ratio; (*c*) decrease the hazards from these malicious nodes and protect the network from internal attacks (e.g. mitigate cooperative denigration attacks); (*d*) enhance the network's ability of trust decisions (e.g., trusted routing decisions).

The remaining paper is organized as follows. Section 2 discusses the related work. In Section 3, we describe our trust management model, and the calculation of trust value is presented in Section 4. Section 5 presents the experimental results on the performance of our trust model. Finally, Section 6 gives the concluding remarks along with extensions and directions for future research.

II. RELATED WORK

In the context of MANET, there are multiple trust management models that have been proposed in the realm of network. In MANET, trust can be considered as the reliance of a network node on the ability of other nodes to forward packets or offer services timely,

978-0-7695-4466-3/11 $26.00 © 2011 IEEE
DOI 10.1109/GreenCom.2011.30

integrally and reliably.

From the evolutionism and sociology points of view, Mui [3] et al. firstly introduced a trust and reputation computation model for generalized networks. In the indirect trust evaluation process, they proposed a graph parallelization algorithm, which is intuitive and easy to understand. Based on the work of Mui, Durad et al. introduced a new term: trust of scaling factor (TSF2) [4], emphasizing the contribution of direct interactions and the rationality of recommendation. They also proposed a modified transformation algorithm (MTA) for TSF2 calculation.

Using the theory of semi-rings, George et al. proposed a new trust model for ad hoc networks [5]. In the model, they described trust evaluation scheme as a routing path problem in a directed weighted graph. When gathering the opinion that one entity has about another entity, they defined two binary operations to evaluate opinion values from single and multiple recommendation paths respectively.

Luo et al. [6] proposed a subjective trust management model based on certainty-factor for MANET (CFStrust) after considering fuzzy set theory and reputation model, which can be used to quantify and evaluate the nodes' credibility. In this model, the problem of trust management is modeled by fuzzy likelihood estimation and confidence estimation. The trust evaluation mechanism and the derivation rules of recommendation trust relationship are given in this model. Although two effective factors corresponding with mathematical derivation are discussed, it does not take a comprehensive account of the nodes' computing power, the instability of information transmission through multi hops and trust's attenuation problem etc.

III. TRUST MANAGEMENT MODEL

In ad hoc networks, every node acts as host and router simultaneously. Trust is a relationship between two neighbor entities. Trust value expresses the degree that one node expects another node to offer certain services.

A. Overview Our Trust Model

An ad hoc network is always comprised of many entities, and each entity is an independent node. In this section, we present our trust model from graph theory, which is denoted as M=<V, E, f>. (1) Trust entity set can be defined as $V=\{v_1, v_2, \ldots, v_n\}$, where n is the scale of the network; (2) E is a relation on V, and $|E|$ is the number of directed network links. Each e_{ij} in E represents a directed edge from node v_i to its neighbor node v_j; (3) $f : f(e_{ij}) \to R \in [0,1]$ denotes the trust value (a real number between 0 and 1) of each edge e_{ij}.

According to our definition, the trust model of an ad hoc network can be represented as a directed weighted graph. For example in Figure 1, there are five nodes in this ad hoc network. Each dashed circle represents the radiation scope of the corresponding node, where the nodes within the scope are neighbor nodes that can communicate directly.

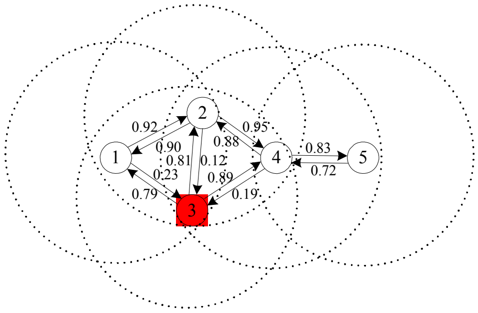


Figure 1. Trust network graph in Ad hoc networks

B. Nodes' Trust Levels

In our model, TV denotes for a node's trust value, which is defined in a continuous range between 0 and 1 (i.e. $0 \le TV_{ij} \le 1$). Let v_i and v_j represent the evaluating and evaluated nodes, respectively. The trust value 0 signifies complete distrust, while the value 1 implies absolute trust. We define simple grading criteria for trust, and an example of node's trust levels is listed in Table 1. A threshold value η, termed as the black-list trust threshold, is used to detect malicious nodes. In other words, if the trust value of a node is smaller than η, it will be regarded as a malicious node by its evaluating node.

Table 1. Trust levels of nodes

Level	***Trust Value***	***Meaning***
1	$[0, \eta)$	*Malicious node*
2	$[\eta, 0.7)$	*Low trustworthy node*
3	[0.7, 0.9)	*Trustworthy node*
4	[0.9, 1]	*Complete trustworthy node*

C. Node's Trust Table

Each node in our model additionally owns a trust table (Table 2 which bonds with Figure 1) with items defined as follows.

Table 2. Node v_1's trust table

Nb	T_{in}	T_{out}	***Black-List***
v_2	0.90	0.92	*No*
v_3	0.79	0.23	*Yes*

In each row of the table, Nb denotes node v_1's neighbor that can communicate with v_1 via a single-hop; T_{in} is the trust value that the neighbor node gets about node v_1; T_{out} is the trust value that node v_1 has about the neighbor; *Black-List* indicates whether v_1 considers this neighbor as a malicious node (e.g., the black-list trust threshold η as discussed in Table 1 is set to 0.4 in this example).

D. Classification of Trust Types

In AFStrust model, there are three types of trust value, which are historical trust value, current trust value and path trust value.

1) ***Node's historical trust***: it is estimated by the node's physical neighbors based on historical interaction information at the end of each time interval, which is calculated by four decision factors. These factors are *direct trust* denoted by DT_{ij}, *recommendation trust* denoted by RT_{ij}, *incentive function* denoted by IF_{ij} and *active degree* denoted by AD_{ij}. The historical trust value HTV_{ij} denotes node v_j's trust level from the evaluating node v_i's point of view, which can be computed by:

$$HTV_{ij} = \alpha DT_{ij} + \beta RT_{ij} + \gamma IF_{ij} + \mu AD_{ij} \qquad (1)$$

Where α, β, γ, μ are coefficients, such that $(\alpha + \beta + \gamma + \mu = 1)$. The values of α, β, γ, μ are decided by the influencing on the assessment of trust value in different environment.

2) ***Node's current trust***: as shown in Figure 2, a node's current trust value predicts the node's trust value in the next time interval *t*+1. In our proposed model, it is computed from the node's historical trust value based on the fuzzy logic rules prediction method. In this paper, we use the term '*trust value*' for a node's *current* trust value, for simplicity of representation.

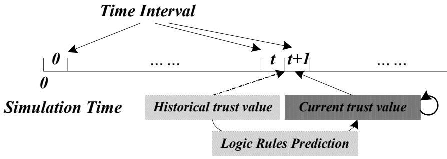


Figure 2. The types of trust value transfer graph

3) ***Path trust***: it expresses the credibility for the set of nodes on a routing path, and its value is defined as the minimum of single-hop trust values. The service (source) nodes determine service level basing on the assessment of path trust value. As a result, the path trust value can be defined as a constraint in the trusted routing decision making.

Path trust: $$PathTV_{ij} = \min_{\substack{i \le m \le j-1 \\ k=m+1}} \{ TV_{mk} \} \qquad (2)$$

From Figure 1, we easily see the examples of nodes' trust values and path trust values: TV_{12}=0.92, TV_{45}=0.83, TV_{43}=0.19; path trust value: *PathTV*$_{14}$=*Path*(1 — >2 —>4)=*min*{0.92, 0.95}=0.92. Setting η=0.4, we see that, node 3 exists in the black lists of all its neighbors, which is regarded as a malicious node by node 1, 2, 4 and will be excluded from the local network for a special time.

IV. CALCULATION OF NODE'S TRUST VALUE

Trust evaluation is the core of trust management system, including the trust definition, trust synthesis, and trust update.

A. Calculation of Historical Trust Value

Those mentioned trust management models analyze trust's decision factors, and give different models to calculate the weights of these factors.

1) The trust decision factors

There are multiple decision factors to assess a node's trust value in existing trust models [7, 8]. For example, two neighbor nodes in an ad hoc network, which can interact with each other directly, may establish a direct trust relationship. Two nodes can also establish relationship by intermediate node's (or path's) recommendation, which is usually called indirect trust or recommendation trust. Besides the above two factors, we introduce another two decision factors, which are *incentive function* and *active degree* into our trust model.

a) Direct trust evaluation rules

In our model, we consider the interaction which occurs between a node and its physical neighbor as a direct interaction, of which the evaluation is defined as direct trust evaluation; while the interaction via multi-hops is defined as indirect interaction, of which the evaluation is defined as indirect trust evaluation (Involved in the following subsection). According to the two special interaction sides (e.g., nodes v_i and v_j), node v_i make a satisfaction evaluation of each direct interaction interacted with its neighbor node v_j, denoted by $f(i,j)$ ($0 \le f(i,j) \le 1$). After evaluation step, the results are obtained in evaluating node's (i.e., v_i's) local memory. In order to obtain an accurate node's trust value, our model distinguishes the different influence of each interaction interval. Using the time stamp mechanism to analyze each interaction interval (e.g. set interval Δt =30(*s*)). Till the current interaction time, there will be *n* intervals [$t_1, t_2 \ldots t_n$] in simulation system. For the *k*-th interaction interval, there are N_{t_k} number of interactions, node v_i makes a final direct trust evaluation for node v_j with the following equation:

$$DT_{ij}^{t_k} = \frac{\sum_{m=1}^{N_{t_k}} f_m(i,j)}{N_{t_k}} \qquad (3)$$

The above equation indicates that the direct trust value of the *k*-th interval is the average of all interaction evaluations within this interval.

Definition 1 *Time decay function*: the attenuation rate function which is made by the *k*-th interaction interval compares to the latest interaction interval in the trust's calculation is defined as the time decay function.

$$f_k = \rho^{n-k}, 0 < \rho < 1, 1 \le k \le n \qquad (4)$$

The base coefficient ρ represents the attenuation factor. A smaller ρ causes a greater attenuation of f_k, and vice versa.

Finally, node v_i calculates a direct trust value for node v_j according to their history interaction evaluations using the following equation.

$$DT_{ij} = \frac{\sum_{k=1}^{n} f_k \times R_{ij}^{t_k} \times (1 - e^{-\frac{N_{t_k}}{\sigma}})}{\sum_{k=1}^{n} f_k \times (1 - e^{-\frac{N_{t_k}}{\sigma}})} \qquad (5)$$

Regulatory factor σ is used to scale the impact of number of interactions on the direct trust computation, and the interaction factor is denoted by $(1 - e^{-\frac{N_{t_k}}{\sigma}})$. The concrete value of σ can be adjusted based on the environment and characteristics of the application. The interaction factor has a negative exponential growth to the number of interactions in a given interval. This factor is used to emphasize the importance of the transaction number. As is shown in above equation, the direct trust value is the weighted average of all interaction evaluations in different interaction intervals.

b) Recommendation trust evaluation rules

Definition 2 *Recommending credibility*: it represents the credibility degree of the recommending node (or recommending path) to provide recommendation experience, which is denoted by *RC*.

We set a recommending credibility threshold d_1, if a recommending node's recommending credibility is less than this threshold ($RC<d_1$), the recommendation experience provided by this node is not considered.

Suppose node v_i gets indirect recommendation trust value of node v_j by the intermediate recommendation of node v_m (a single node). Setting $DT_{im}=p$, $DT_{mj}=q$, we can calculate:

$$RT^*_{imj} = RC_{im} \times DT_{mj} = DT_{im} \times DT_{mj} = p * q \tag{6}$$

This can be shown in Figure 3(*a*).

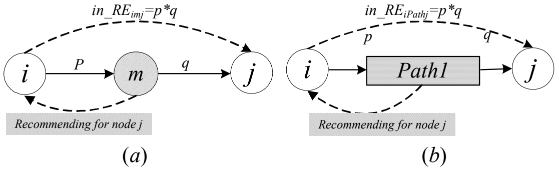


Figure 3. Node's Expurgating Rules

Suppose node v_i gets a recommendation trust value of node v_j by the intermediate recommendation of path *P*. Setting $DR_{iPathP}=p$, $R_{nj}=q$, then we have:

$$RT_{iPathj} = RC_{iPathP} \times R_{nj} = p * q \tag{7}$$

Node v_n is the Path *P*'s terminal node. This can be shown in Figure 3(*b*). The single intermediate node's recommendation style can be seen as a special form of path's recommendation.

Assume node v_i gets recommendation experience RT^*_{ij} by trust propagation along a directed recommending path $P=<v_i, v_1, v_2, \ldots, v_n, v_j>$. Currently we can't say the value RT^*_{ij} got from trust attenuation rules is a reasonable value, because the recommending nodes in the path is not absolutely trustworthy from node *i*'s point of view. In recommendation experience evaluation rules, the node v_m's recommending credibility in this recommending path is denoted by RC_{im}.

$$RC_{im} = DT_{i1} * DT_{12} * \ldots * DT_{(m-1)m} \tag{8}$$

Capturing the notion of social networks, node v_i should give an objective estimate to all recommending nodes in a recommending path. We obtain a recommending credibility value RC_{iPathP} for path *P*'s recommending credibility, which can be calculated with the following equation:

$$RC_{iPathP} = \frac{\sum_{k=1}^{n} RC_{iv_k}}{n} = \frac{DT_{i1} + \sum_{k=2}^{n} [DT_{i1} * DT_{12} * \ldots * DT_{(k-1)k}]}{n} \tag{9}$$

Where RC_{ik_v} is the recommending credibility that node v_i has about node v_k. Therefore, the path *P* makes recommendation experience as RT_{iPj} :

$$RT_{iPj} = DR_{iPathP} * R_{nj} \tag{10}$$

Node v_n is path *P*'s terminal node.

If there are *n* recommending paths $P_1, P_2, \ldots, P_n$ between node v_i and v_j, node v_i should give an objective estimate to all recommending paths. Each path has its local weight recommending credibility W_{Path}. However, when comes to all, each path plays different role which based on their local weight. Then node v_i should give an objective estimate for each path. Thus we define path P_k's general weight W_{P_k} as follows:

$$W_{P_k} = \frac{RC_{iP_k}}{\sum_{k=1}^{n} RC_{iP_k}} \tag{11}$$

With $W_{P_k} \in [0,1]$ and $\sum_{k=1}^{n} W_{P_k} = 1$.

Assuming the in-degree of node v_j is *n*, it means there are *n* paths from node v_i to node v_j. Using trust attenuation rules, we can get *n* recommendation experience: RT_{iP1j}, RT_{iP2j}, …, RT_{iPnj}. In terms of recommending path weighting rules for multiple paths above, we can calculate these recommending paths' general weights and get W_{P_k} ($k \in \{1,2,\ldots n\}$). Then node v_i can calculate the RT_{ij} for node v_j as follows:

$$RT_{ij} = \sum_{k=1}^{n} W_{P_k} * RT_{iP_k j} \tag{12}$$

c) Incentive function evaluation rules

Definition 3 *Incentive function*: this function reflects the incentive for cooperative entities.

Because of the cooperative entities often have fewer bad interactions and less interaction failure rates, while malicious or uncooperative entities often refuse to or interrupt service. Incentive function also reflects that the system would make some punishment to the uncooperative entities. This function is denoted by IF_{ij}, which is calculated with following equation:

$$IF_{ij} = 1 - \phi(n) \tag{13}$$

$\Phi(n)$ represents penalty factor, which is used to indicate that the node does not fulfill its responsibility, and do harm to evaluating a node's trust value. When the satisfaction evaluations of interactions are less than a threshold *d*, we call those interactions as malicious interactions. Till the current interaction time (*T*) between node v_i and node v_j, the total number of malicious interactions is denoted by m_T, the total number of interactions is denoted by N_T. The penalty factor is calculated with the following equation:

$$\phi(n) = \varphi^{\frac{N_T - m_T}{N_T}} \quad (\varphi < 1) \tag{14}$$

d) Active degree evaluation rules

Definition 4 *Active degree*: this decision factor reflects the level of activity of an entity in a network.

It is used to indicate the credibility of evaluated entity. If an (evaluated) entity has a higher active degree, other (evaluating) entities is willing to interact with it due to its expected higher trust level. An evaluating node v_i records the cumulative number of entities interacting with an evaluated node v_j, and calculates the active degree of the evaluated node as follows:

$$AD_{ij} = 1 - \frac{\eta}{L+1}, L \geq 0 \quad (15)$$

L represents for the cumulative number of entities interacted with the evaluated node v_j. η termed as the black-list trust threshold (discussed in Subsection 3.2).

2) AHP theory

Analytic Hierarchy Process (AHP) [9] theory was proposed by Saaty in the 1970s, which has become one of the essential multi-criteria decision making methods. It combines qualitative and quantitative factors in the analysis via a multi-index synthetic assessment algorithm. The analysis can be divided into the following four steps: (1) construct a hierarchical structure model; (2) form basic judgment matrix; (3) calculate the weight vector of decision factors; (4) make a consistency test for the judgment matrix.

We use AHP to precisely determine the weights of trust's decision factors based on the nodes' historical behaviors. Node's historical trust value:

$$HTV_{ij}{=}0.56DT_{ij}{+}0.264RT_{ij}{+}\ 0.122IF_{ij}{+}0.054AD_{ij}$$

B. Calculation of Current Trust Value

When the requested (transmitter) node receives a packet transmission request, it's hard for requesting node to evaluate whether the requested node is willing or not to provide the service. However, its historical interactions can be recorded and the node's capability level can be monitored, therefore, we can model these factors as follows [10]: Let $TV(t)$ represents a evaluated (requested or transmitter) node's historical trust level at the end of time interval t, its record of historical behaviors on offering certain services in the latest time interval, which has been measured in Subsection 4.1 using AHP theory. Let $C(t{+}1)$ represents the same node's capability level on providing service level for the next time interval $t{+}1$ (prediction time interval), which includes the remnant utilization ratio of battery, local memory, CPU cycle, and bandwidth at that point. Let $TV(t{+}1)$ refers to the same node's current trust level for the next time interval $t{+}1$. Assume the fuzzy membership function of $TV(t)$ or $TV(t{+}1)$ consists of four fuzzy sets VeryLow(*VL*-Malicious node), Low(*L*-Low trustworthy node), Medial(*M*-Trustworthy node) and High(*H*-Complete trustworthy node), and the fuzzy member function of $C(t{+}1)$ consists of three fuzzy sets LOW(*L*-Low capability level), Medial(*M*-Medium capability level) and High(*H*-High capability level), respectively. Combined with social control theory, we give the fuzzy inference rules as follows (Table 3):

Table 3. Logical rules prediction on trust levels

TE (t) / ***C (t+1)***	*VL*	*L*	*M*	*H*
L	*VL*			*L*
M	*VL*		*M*	
H	*VL*	*L*	*M*	*H*

The rules in the above table actually establish a mapping function from $TV(t){\times}C(t{+}1)$ to $TV(t{+}1)$, which is based on the analysis of the node's historical behaviors and current conditions. For example, when an overloaded node lacks the CPU cycles, buffer space or available network bandwidth to forward packets, with such a low capability level, even if its historical trust level is very high, it's also untrustworthy in next time. This only shows the first rule from above table. Corresponding with each rule, there is an inference relationship R_l:

$$R_l = TV_t \times C_{t+1} \times TV_{t+1} \quad (23)$$

That is for $\forall h \in TV(t), c \in C(t+1), u \in TV(t+1)$, we have

$$R_l(h,c,u) = TV(h) \wedge C(c) \wedge TV(u) \quad (24)$$

For all the n rules we have the fuzzy inference relationship

$$R(h,c,u) = \overset{n}{\underset{l=1}{\vee}} R_l(h,c,u) \quad (25)$$

For each pair of given ($TV(t)^*, C(t+1)^*$), using the general total relationship R, we can obtain the output:

$$TV(t+1)^* = (TV(t)^* \times C(t+1)^*) \circ R \quad (26)$$

Then with the help of the maximum membership degree approach, we can get an explicitly node's current trust value (a complex representation for node's trust value) $u^* \in [0,1]$ by defuzzification. For each time interval (Δt), at the end of sub-time interval (e.g. one fifth of the time interval, $\Delta t/5$), making the latest node's current trust value as node's historical value and taking this value as the input, we recycle the fuzzy logic rules prediction method to update the node's rust value.

V. EXPERIMENTAL RESULTS

All experiments are carried out on a PC machine with a Pentium 4 processor (2.4 GHz) and 2GB main memory. To evaluate the performance and validity of Durad's TSF2 [4], Luo's CFStrust [6] and our management model AFStrust, we have conducted a comprehensive test using Netlogo simulator [11].

A. Experiment Setup

We use Netlogo platform to construct a simulated environment for MANET, and make a comparative analysis for the performance of the related models. The platform initializes a MANET with 40 nodes, which contains malicious nodes. Basic experimental parameters are set as follows [12].

Table 4. Parameters setting

N	T	Δt	ρ	σ	φ	d	η
40	300(s)	30(s)	0.9	5	0.9	0.7	0.4

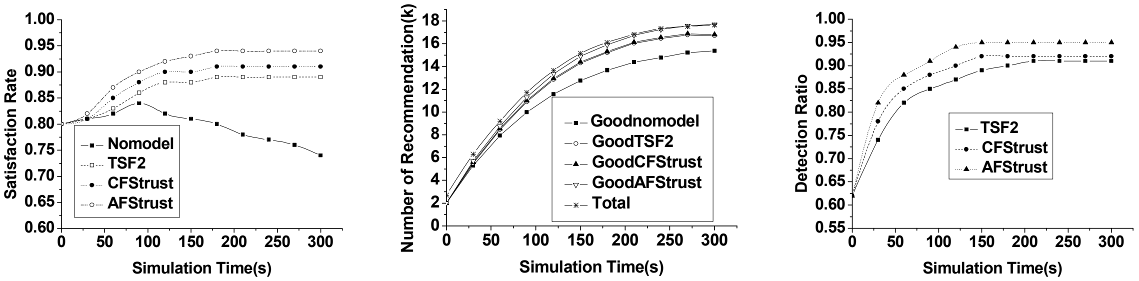
In Table 4, the network size denoted by N, the simulation time denoted by T, the duration of times tamp denoted by Δt, the base number of decay function denoted by ρ, the adjustable parameters of the factor for the number of transactions denoted by σ, the base number of the penalty factor denoted by φ, the threshold value for the evaluation of malicious interaction denoted by d, and the black-list trust threshold denoted by η.

B. Contributions to Network

1) Ad hoc Network with 10% malicious nodes

The platform initializes a MANET which contains with 10% percentage of malicious nodes. We make a comparison in terms of performance between TSF2, CFStrust and our trust model as shown in Figure 4.

Figure 4(*a*) demonstrates the comparison of the satisfaction rates of network interaction with no-model, TSF2, CFStrust and AFStrust models. From this figure, we see that the satisfaction rates of network interaction with trust management models rise with the increase of simulation time, while the satisfaction rate with no-model decreases gently, and the satisfaction rates with trust models are significantly higher than that with no trust model. This advancement with trust models can be attributed to the node's trust mechanism, which elevates the probability of successful (or good) service to a trustworthy node. If one node is thought to be as a malicious node by its neighbors who will not continue the interactions with it, so the interactions will be only happened among normal nodes, leading to that the satisfaction rates of network interaction rise. While with no-model, malicious nodes longer exist, the greater damage on the network, the satisfaction rate of network interaction reduces with the operation of system. The performance with AFStrust is better than those with TSF2 and CFStrust. Taking simulation time 120(*s*) for example, with no-model and three trust models (TSF2, CFStrust and AFStrust), we get the satisfaction rates of network interaction as 0.81, 0.87, 0.88, 0.92 respectively. Comparing with CFStrust, the satisfaction rate based on AHPSLtrust is increased by 4%, comparing with no-model that is increased by 11%.



(*a*) (*b*) (*c*)

Figure 4. Simulating Ad hoc Network with malicious nodes

The comparison of the numbers of good recommendation with different trust models are shown in Figure 4(*b*). The curves of GoodnoModel, GoodTSF2, GoodCFStrust and GoodAFStrust respectively represent the number of good recommendation in network with no model, TSF2, CFStrust and AFStrust model. From this figure, we see that with the increase of total number of recommendation in network, the proportion numbers of good recommendation with trust models rise, while the number with no-model decreases gently. The proportion number with trust model is significantly higher than with no-model. Due to the help of trust management mechanism, the network could identify its inherent malicious nodes. When calculating the nodes' trust values, the proportion of good recommendation experience provided by malicious nodes will be neglected, and the malicious nodes will be slowly removed from the set of recommending nodes. The proportion number of bad recommendation gets smaller with the operation of system, so good recommendation occupies total recommendation ascend. Due to the detailed and complete recommendation trust evaluation mechanism in AFStrust model, the performance with this model is better than TSF2 and CFStrust. Taking simulation time 120(*s*) for example, the proportion number of good recommendation with AFStrust is 5 percentage points higher than that with CFStrust.

From Figure 4(*c*), we see that the comparison of the correct detection ratio of malicious nodes. Trust evaluation is an attractive target for those malicious adversaries. The results of trust evaluation can be used to detect malicious nodes in a network. The curves of TSF2, CFStrust and AFStrust respectively represent the correct detection ratio of malicious nodes with TSF2, CFStrust and AFStrust models. This figure obviously shows that, the correct detection ratio of malicious nodes with trust management models increases sharply with the increase of simulation time (0~60*s*), then these values are stable, finally the best one (AFStrust) reaches 0.95. Due to the detailed and complete trust evaluation mechanism (e.g. based on multiple decision factors) and fuzzy logic rules prediction method in AFStrust model, this model makes a more accurate detection ratio of malicious nodes and its performance is better than TSF2 and CFStrust models. Comparing with CFStrust, at the time 120*s*, the correct detection ratio of malicious nodes is increased by 3%.

2) Prevent from cheating attacks

In order to cheat a high reputation, a malicious node may make good performance (or disguise itself to be a good node) in a special time interval, and then behave badly. Proposed trust models are not successful in prevent from this attack, and the node's trust value based on those models has big volatilities. Based on two novel trust's decision factors (i.e., detailed recommendation trust and incentive function) and fuzzy logical rules prediction method, our trust model has ability to prevent malicious nodes from cheating attacks. The comparison performance with different trust models is shown in Figure 5.

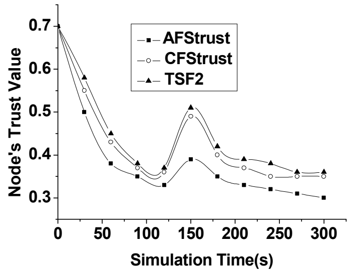


Figure 5. Trust value fluctuation curve of malicious node

Figure 5 shows that, the effect of attack on our model is less than that on TSF2 and CFStrust, and the malicious nodes' trust value with AFStrust is smaller than that with TSF2 and CFStrust. At time 0*s*, an unknown (malicious) node's trust value is initialized to 0.7. With the increase of the simulation time, this malicious node is found by the system respectively based on different trust models, and the AFStrust has a better performance than TSF2 and CFStrust in malicious node early recognition. In order to cheat a high trust value, a malicious node makes good performance in time interval (120*s*~150*s*). From Figure 5, we easily see that AFStrust model effectively mitigate the harm caused by such attacks, while this node's trust values based on TSF2 and CFStrust have big volatilities. Due to the special decision factor (i.e., incentive function), fuzzy logic rules and black-list mechanism in our trust model, in the middle and later periods, this malicious node's trust values based on AFStrust decrease gently while that with the other two models maintain a regular value. This experiment also hints that the trust obtains hardly while it loses easily.

VI. CONCLUSIONS AND FUTURE WORKS

This article studies the questions related with the definition and synthesis of nodes' trust value for MANET. Basing on the inherent characteristics of this network, a subjective trust management model for MANET (AFStrust) based on the analytic hierarchy process theory and fuzzy logic rules prediction method is proposed. Building weight mechanism based on the AHP theory to calculate the weights of multiple trust decision factors and handle fuzzy logic rules prediction method to predict the node's trust value make our model more stable, adaptive and robust, which consequently enhances network's security and performance. The simulation results analyze the effectiveness of our trust management model. This model works as an intuitive and effective evaluation, analysis and derivation tool, it could provide effective support for the trust decisions and against attacks.

We would continue our work in the following three directions: 1. Make a further improvement for the trust model proposed in this paper, we plan to incorporate other decision factors to this model; 2. We will consider an adaptive trust level classification of nodes taking into account the average trust value of all nodes. The problem of dynamic behavior modification will also be considered; Moreover, 3. Propose a detailed trust-based on-demand multi-path routing for MANET in which the node's trust value is calculated by our trust model.

ACKNOWLEDGMENT

This research is sponsored by the Natural Science Foundation of China (NSFC) under grant No. 61070022, 60903031, Shandong Provincial Natural Science Foundation under grant no. ZR2010FM015.

REFERENCES

[1] Yan Lindsay Sun, Zhu Han, Wei Yu, and K J.Ray Liu.: 'Attacks on trust valuation in distributed networks', Proc. of the 40th Annual Conference on Information Sciences and Systems, March 2006, pp. 1461-1466.

[2] A.A. Pirzada, and C. McDonald.: 'Trust establishment in pure ad-hoc networks', Wireless Personal Communications, 2006, 37, pp. 39-168.

[3] L. Mui.: 'Computational models of trust and reputation: agents, evolutionary games, and social networks', Ph.D. Thesis. MIT. Massachusetts, 2003.

[4] M.H. Durad, Yuanda Cao, Liehuang Zhu.: 'Two novel trust evaluation algorithm', Proc. of the Communications Circuits and Systems, June 2006, Vol.3, pp. 1641-1646.

[5] George.: 'Distributed trust evaluation in ad hoc networks', Proc. of the 3rd ACM workshop in Wireless Security, Oct. 2004, pp.1-10.

[6] Junhai Luo, Mingyu Fan.: 'A subjectiveive trust management model based on certainty-factor for MANETs', Chinese Journal of Computer Research and Development, 2010, 47(3), pp. 515-523.

[7] Hui Xia, Zhiping Jia, Xin Li, Feng Zhang.: 'A Subjectiveive Trust Management Model based on AHP for MANET', Accessed by The 2011 International Conference on Network Computing and Information Security (NCIS'11).

[8] Xiaoyong Li, Xiaolin Gui.: 'Trust Quantitative Model with Multiple Decision Factors in Trusted Network', Chinese Journal of Computers, 32, 2009, pp. 405-416.

[9] T.L. Satty.: 'The analytic hierarchy process', New York: McGraw-Hill, 1980.

[10] Timothy J. Ross.: 'Fuzzy Logic with Engineering Applications', McGraw Hill International Editions, International Editions, 2000.

[11] U. Wilensky, Netlogo, 1999, http://ccl. northwestern. edu/netlogo/.

[12] Liangchen Wen, Xuefeng Zhang, Limei Zhu.: 'Method of ameliorative multi-objective synthetic evaluation based on entropy weight and its application', Control and Decision Conference, 2009. CCDC '09. Chinese, pp. 1538-1541.