Trust Modeling and Evaluation for Ad Hoc Networks

MuRI Report No. 20041017-21

Yan Lindsay Sun
Dept. of Electrical and Computer Engineering
University of Rhode Island
Wei Yu, Zhu Han, and K. J. Ray Liu
Dept. of Electrical and Computer Engineering
University of Maryland

October 17, 2004

Abstract

The performance of ad hoc networks depends on the cooperative and trust nature of the distributed nodes. To enhance security in ad hoc networks, it is important to evaluate the trustworthiness of other nodes without centralized authorities. In this paper, we present an information theoretic framework to quantitatively measure trust and model trust propagation in ad hoc networks. In the proposed information theoretic framework, the trust is a measure of uncertainty with its value represented by entropy. We develop four Axioms that address the basic understating of trust and rules for trust propagation. Based on these Axioms, we present two trust models: entropy-based model and probability-based model, which satisfy all the Axioms. Techniques of trust establishment and trust update are presented to obtain trust values from observations. The proposed trust evaluation method and trust models are employed in ad hoc networks for secure ad hoc routing and malicious node detection. A distributed scheme is designed to acquire, maintain, and update trust records associated with the behaviors of nodes' forwarding packets and the behaviors of making recommendations of other nodes. Simulations show that the proposed framework can significantly improve the network throughput as well as effectively detect malicious behaviors in ad hoc networks

I. Introduction

An ad hoc network is a group of mobile nodes without requiring a centralized administration or a fixed network infrastructure. Due to their distributed nature, ad hoc networks are vulnerable to various attacks [1]–[5]. One strategy to improve security of ad hoc networks is to develop mechanisms that allow a node to evaluate trustworthiness of other nodes. Such mechanisms not only help in malicious node detection, but also improve network performances because honest nodes can avoid working with less trustworthy nodes. The focus of this paper is to develop a framework that defines trust metrics using information theory and develops trust models of trust propagation in ad hoc networks. The proposed theoretical models are then applied to improve the performance of ad hoc network routing schemes and to perform malicious node detection.

The problem of defining trust metrics and trust relationship has been extensively studies for public key authentication [6]–[10], electronics commerce [11], as well as in P2P Networks [12], [13]. In these schemes, trust is evaluated in very different ways. Some schemes employ linguistic description of trust relationship, such as in PGP [7], [14], PolicyMaker trust management system in [15], distributed trust model in [16], trust policy language in [17], and SPKI/SDSI public-key infrastructure [18]. Based on linguistic descriptions of the trust metrics, decisions can be made based on linguistic trust policies or fuzzy logic [11]. In some other schemes, discrete or continuous numerical values are assigned to measure the level of trust [8], [9], [16]. For example, in [8], an entity's opinion about the trustworthiness of a certificate is described by a continuous value in $[0, 1]$. In [9], a triplet in $[0, 1]^3$ is assigned to measure the trustworthiness where the elements in the triplet represent believe, disbelief, and uncertainty, respectively. In [16], discrete integer numbers are used.

Before we can compare different trust evaluation methods or discuss trust models for ad hoc networks, a fundamental question needs to be answered first. What is the physical meaning of trust? The answer to this question is the critical link between observations (trust evidence) and the metrics that evaluate trustworthiness. In ad hoc networks, trust relationship can be established in two ways. The first way is through direct observations of other nodes' behavior, such as dropping packets etc. The second way is through recommendations from other nodes. Without clarifying the

meaning of trust, trustworthiness cannot be accurately determined from the observations, and the calculation/policies/rules that govern trust propagation cannot be justified.

Previous work on trust management in ad hoc networks focuses on the trustworthiness evaluation process after initial trust relationship has been established. They do not, however, address how to obtain initial trust relationship partially because the meaning of the trust metrics is not clearly defined. We approach the trust evaluation problem from a definition of trust given by Diego Gambetta in [19]. It states that trust is a level of likelihood with which an agent will perform a particular action before such action can be monitored and in a context in which it affects our own actions. It is clear that trust relationship involves two entities and a specific action. The concept of trust exists because we are not sure whether the agent will perform the action or not in some circumstances.

In the proposed information theoretic framework of trust modeling and evaluation, trust is a measure of uncertainty, as such trust values can be measured by entropy. From this understanding of trust, we developed axioms that address the basic rules for establishing trust through a third party (concatenation propagation) and through recommendations from multiple sources (multipath propagation) in ad hoc networks. Based on these axioms, we develop techniques that calculate trust values from observations and design two models that address the concatenation and multipath trust propagation problems in ad hoc networks. The proposed models are applied to improve the performance and security of ad hoc routing protocols. In particular, we investigate trust relationship associated with packet forwarding as well as making recommendations. We develop a distributed scheme to build, maintain, and update trust records in ad hoc networks. Trust records are used to assist route selection and to perform malicious node detection.

Simulations are performed to evaluate the effectiveness of the proposed models in real ad hoc networks. For malicious node detection, the proposed scheme can let individual user to obtain the trust values of forwarding packets and making recommendations in a distributed way. The malicious nodes can be detected and their types can also be identified. The proposed scheme can also track the dynamics of the networks adaptively. Compared with a base line scheme without trust models, the proposed scheme can select the route with higher recommended qualities so that the packet

dropping rates are greatly reduced. To reduce the network throughput, it takes much more number of malicious nodes for the proposed scheme than for the base line scheme.

The rest of the paper is organized as follows. The understanding of trust and basic axioms are presented in Section II. Section III describes entropy-based and probability-based trust models and proves that our models satisfy all Axioms. In Section IV, we investigate how to establish trust relationship based on observations. In Section V, the proposed models are applied in ad hoc networks to assist route selection in on-demand routing protocols and to perform malicious node detection. Simulation results are shown in Section VI. Conclusions are drawn in Section VII.

II. Basic Axioms

In this section, we will explain the meaning of trust and present four axioms for establishment of trust relationship. In this work, we relay trust as a level of uncertainty and the basic understanding of trust is summarized as follows.

1) Trust is a relationship established between two entities for a specific action. In particular, one entity trusts the other entity to perform an *action*. In this work, the first entity is called the ***subject***, the second entity is called the ***agent***. We introduce the notation $\{subject : agent, action\}$ to describe a trust relationship.
2) Trust is a function of uncertainty. In particular, if the subject believes that the agent will perform the action for sure, the subject fully "trusts" the agent to perform the action and there is no uncertainty; if the subject believes that the agent will not perform the action for sure, the subject "trusts" the agent not to perform the action, and there is no uncertainty either; if the subject does not have any idea of whether the agent will perform the action or not, the subject does not have trust in the agent. In this case, the subject has the highest uncertainty.
3) The level of trust can be measured by a continuous real number, referred to as the *trust value*. Trust value should represent uncertainty.
4) The subjects may have different trust values with the same agent for the same action. Trust is not necessarily symmetric. The fact that A trusts B does not necessarily means that B also trusts A, where A and B are two entities.

Based on our understanding of trust, we further developed basic axioms for establishing trust relationship through either direct interactions, or through recommendations without direct interactions between the agents and the subjects.

Axiom 1: Uncertainty is a measure of trust

The concept of trust is the certainty of the subject about whether or not the agent will perform an action. Let $T\{subject : agent, action\}$ denote the trust value of the trust relationship $\{subject : agent, action\}$, and $P\{subject : agent, action\}$ denote the probability that the agent will perform the action in the subject's point of view. Information theory states that entropy is a nature measure for uncertainty [20]. Thus, we define the entropy-based trust value as:

$$T\{subject : agent, action\} = \begin{cases} 1 - H(p), & \text{for } 0.5 \leq p \leq 1; \\ H(p) - 1, & \text{for } 0 \leq p < 0.5, \end{cases} \tag{1}$$

where $H(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ and $p = P\{subject : agent, action\}$. In this work, the trust value is a continuous real number in [-1,1]. This definition satisfies the following properties. When $p = 1$, the subject trusts the agent the most and the trust value is 1. When $p = 0$, the subject distrusts the agent the most and the trust value is -1. When $p = 0.5$, the subject has no trust in the agent and the trust value is 0. In general, trust value is negative for $0 \leq p < 0.5$ and positive for $0.5 < p \leq 1$. Trust value is an increasing function with p. It is noted that (1) is a one-to-one mapping between $T\{subject : agent, action\}$ and $P\{subject : agent, action\}$. In the sequel, we use both values in the description of trust relationship.

Axiom 2: Concatenation propagation of trust does not increase trust

When the subject establishes a trust relationship with the agent through the recommendation from a third party, the trust value between the subject and the agent should not be more than the trust value between the subject and the recommender as well as the trust value between the recommender and the agent. Axiom 2 states that uncertainty increases through propagation.

The trust relationship can be represented by a directional graph shown in Figure 1, where the weight of the edge is the trust value. The style of the line represents the type of the action: dashed lines represent making recommendations and solid lines represent performing the action. When relationship $\{A : B, action_r\}$ and $\{B : C, action\}$ are available, trust relationship $\{A : C, action\}$ can be established if the following two conditions are satisfied.

1. The $action_r$ is to make recommendation of other nodes about performing the $action$.
2. The trust value of $\{A : B, action_r\}$ is positive.

The first condition is necessary because the entities who performs the action do not necessarily make correct recommendations [16]. The second condition states that the recommendations from untruthful entities should not be used. The second condition makes the trust propagation in distributed networks resilient to malicious entities who can manipulate their recommendations for causing maximum damage. It is noted that the second condition is not necessary in some other situations where the malicious nodes' behavior of making recommendations is predictable.

The mathematical representation of Axiom 2 is

$$|T_{AC}| \leq \min(|R_{AB}|, |T_{BC}|), \tag{2}$$

where $T_{AC} = T\{A : C, action\}$, $R_{AB} = T\{A : B, action_r\}$ and $T_{BC} = T\{B : C, action\}$. This is similar to information processing in information theory: the information cannot be increased via propagation. In our case, the trust from others' recommendations is no more than the recommenders' trust and the trust to the recommenders.

Axiom 3: Multipath propagation of trust does not reduce trust

If the subject receives the same recommendations for the agents from multiple sources, the trust value should be no less than that in the case where the subject receives less number of recommendations.

In particular, as illustrated in Figure 2, A establishes trust with C' through one concatenation path, and A establishes trust with C through two same trust paths. Let $T_{AC} = T\{A : C, action\}$ and $T_{AC'} = T\{A : C', action\}$. The mathematical representation of Axiom 3 is

$$T_{AC} \geq T_{AC'} \geq 0, \text{ for } R_1 > 0 \text{ and } T_2 \geq 0;$$

$$T_{AC} \leq T_{AC'} \leq 0, \text{ for } R_1 > 0 \text{ and } T_2 < 0,$$

where $R_1 = T\{A : B, \text{making recommendation}\}$ and $T_2 = T\{B : C, action\}$. Axiom 3 states that multipath recommendations will not increase uncertainty. Notice that Axiom 3 holds only if multiple sources generate the same recommendations. This is because the collective combination of different recommendations is a problem in nature that can generate different trust values according to different trust models.

Axiom 4: Trust based on multiple recommendations from a single source should not be higher than that from independent sources

When the trust relationship is established jointly through concatenation and multipath trust propagation, it is possible to have multiple recommendations from a single source, as shown in Figure 3 (a). Since the recommendations from a single source are highly correlated, the trust built on those correlated recommendations should not be higher than the trust built upon recommendations from independent sources. In particular, let $T_{AC'} = T\{A : C', action\}$ denote the trust value established in Figure 3 (a), and $T_{AC} = T\{A : C, action\}$ denote the trust value established in Figure 3 (b). The Axiom 4 says that

$$T_{AC} \geq T_{AC'} \geq 0, \text{ if } T_{AC'} \geq 0;$$

$$T_{AC} \leq T_{AC'} \leq 0, \text{ if } T_{AC'} < 0,$$

where R_1, R_2, and R_3 are all positive. The physical meaning of this Axiom is that the recommendations from independent sources can reduce uncertainty more effectively than the recommendations from correlated sources.

As a summary, the above four basic Axioms address different aspects of trust relationship. Axiom 1 states the meaning of trust. Axiom 2 states the rule for concatenation trust propagation. Axiom 3 describes the rule for multipath trust propagation. Axiom 4 addresses correlation of recommendations.

III. Trust Models

The methods for calculating trust via concatenation and multipath propagation are referred to as *trust models*. In this section, we introduce entropy-based and probability-based trust models and prove that they satisfy all Axioms.

A. Entropy-based Trust Model

In this model, the trust propagations are calculated directly from trust values defined in (1). For concatenation trust propagation shown in Figure 1, node B observes the behavior of node C and makes recommendation to node A as $T_{BC} = \{B : C, action\}$. Node A trusts node B with $T\{A : B, \text{making recommendation}\} = R_{AB}$. The question is how much node A should trust node C to perform the action. To satisfy Axiom 2, one way to calculate $T_{ABC} = T\{A : C, action\}$ is

$$T_{ABC} = R_{AB} T_{BC}. \tag{3}$$

Note that if node B has no idea about node C (i.e. $T_{BC} = 0$) or if node A has no idea about node B (i.e. $R_{AB} = 0$), the trust between A and C is zero, i.e., $T_{ABC} = 0$.

For multipath trust propagation, let $R_{AB} = T\{A : B, \text{making recommendation}\}$, $T_{BC} = T\{B : C, action\}$, $R_{AD} = T\{A : D, \text{making recommendation}\}$, $T_{DC} = T\{D : C, action\}$. Thus, A can establish trust to C through two paths: $A-B-C$ and $A-D-C$. To combine the trust established through different paths, we propose to use maximal ratio combining as:

$$T\{A : C, action\} = w_1(R_{AB} T_{BC}) + w_2(R_{AD} T_{DC}), \tag{4}$$

where

$$w_1 = \frac{R_{AB}}{R_{AB} + R_{AD}}, \quad \text{and} \quad w_2 = \frac{R_{AD}}{R_{AB} + R_{AD}}. \tag{5}$$

In this model, if any path has the trust value 0, this path will not affect the final result. It is noted that the weight factors in our model are based on recommendation trust R_{AB} and R_{AD}.

Finally, we prove that (3) and (4) satisfy Axioms. Since $T \in [-1, 1]$, the multiplication in (3) will make the absolute value of $T\{A : C, action\}$ smaller or equal to $|T\{A : B, \text{making recommendation}\}|$ and $|T\{B : C, action\}|$. Thus, Axiom 2 is satisfied. When applying (3) and (4) to the special cases illustrated in Figure 2 (the third Axiom), we obtain $T_{AC} = R_1 T_2$ and $T_{AC'} = \frac{(R_1)^2 T_2 + (R_1)^2 T_2}{R_1 + R_1} = T_{AC}$. Thus, Axiom 3 is satisfied with equality. When applying the model to the cases in Figure 3, we can prove that $T_{AC} = T_{AC'} = \frac{R_1(R_2^2 T_4 + R_3^2 T_5)}{R_2 + R_3}$. Thus, Axiom 4 is satisfied with equality.

B. Probability-based Model

In the second model, we calculate concatenation and multipath trust propagation using the probability values of the trust relationship. Then, the probability values can be easily transferred back to trust values using (1).

For the concatenation in Figure 1, let p_{AB} denote the $P\{A : B, \text{make recommendation}\}$, p_{BC} denote $P\{B : C, action\}$ and p_{ABC} denote $P\{A : C, action\}$. We also define p'_B as the probability that B will make correct recommendations, $p'_{C|B=1}$ as the probability that C will perform the action if B makes correct recommendation, and $p'_{C|B=0}$ as the probability that C will perform the action if B does not make correct recommendation. Then, A can calculate p_{ABC} as:

$$p_{ABC} = p'_B \cdot p'_{C|B=1} + (1 - p'_B) \cdot p'_{C|B=0}. \tag{6}$$

Although A does not know p'_B, $p'_{C|B=1}$ and $p'_{C|B=0}$, it is reasonable for A to assume that $p'_B = p_{AB}$ and $p'_{C|B=1} = p_{BC}$. Therefore, (6) becomes

$$p_{ABC} = p_{AB} \cdot p_{BC} + (1 - p_{AB}) \cdot p'_{C|B=0}. \tag{7}$$

From Axiom 2, it is easy to see that T_{ABC} should be 0 when T_{AB} is 0. That is, p_{ABC} should be 0.5 when p_{AB} is 0.5. By using $p_{AB} = 0.5$ and $p_{ABC} = 0.5$ in (7), we can show that $p'_{C|B=0} = (1-p_{BC})$. Therefore, we calculate p_{ABC} as

$$p_{ABC} = p_{AB}p_{BC} + (1 - p_{AB})(1 - p_{BC}). \tag{8}$$

It worthy to mention that the above propagation model can also be viewed as binary symmetry channel (BSC) model [20]. The physical meaning of BSC is as follows. When node B claims 1, node A would think that the probability that 1 really happens is p and 0 happens with probability $1 - p$. The value of p is related with the uncertainty associated with the trust relationship between A and B. Similarly, when node B claims 0, A would think that 0 happens with probability p and 1 happens with probability $1-p$. The concatenation of two BSC models also generate the probability expression in (8).

For the multipath case, as shown in Figure 2, we obtain the probability value p_{ABC} through path $A - B - C$ and p_{ADC} through path $A - D - C$ using (8). The question is how to obtain the overall trust $p_{AC} = P\{A : C, action\}$ between node A and node C. This problem has similarity as the data fusion problem where observations from different sensors are combined. Thus, we use the data fusion model [21] with the assumption that the recommendations are independent. So the probability p_{AC} can be calculated as follows:

$$\frac{p_{AC}}{1 - p_{AC}} = \frac{p_{ABC}p_{ADC}}{(1 - p_{ABC})(1 - p_{ADC})}. \tag{9}$$

Note that in this model, if one path has probability value of 0.5 (i.e. no information), this path does not affect the final result of probability.

Next we show that the probability-based models satisfy the Axioms. For Axiom 2, it can be easily shown that $H(p_{ABC}) \geq H(p_{BC})$ and $H(p_{ABC}) \geq H(p_{AB})$ with equality hold if and only if $p_{AB} = 1$ and $p_{BC} = 1$, respectively. Thus, Axiom 2 holds. For Axiom 3, if both p_{ABC} and p_{ADC} are no less than 0.5, from (9), p_{AC} must be larger than both p_{ABC} and p_{ADC}. If both p_{ABC} and p_{ADC} are smaller than 0.5, p_{AC} must be smaller than both p_{ABC} and p_{ADC}. So Axiom 3 holds.

From (8) and (9), we can prove that this model also satisfies Axiom 4 and equality is achieved when any link has trust value of 0.

IV. Trust Establishment Based on Observations

The problem we address in this section is to obtain the trust value from observations. Assume that A wants to establish the trust relationship with X as $\{A : X, act\}$ based on A's previous observation about X. One typical type of observation is as follows. Node A observed that X performed the action k times upon the request of performing the action N times. For example, A asked X to forward N packets, and X in fact forwarded k packets. For this type of observation, we define random variables $V(i)$ and $n(N)$ as:

$V(i) : V(i) = 1$ means that X performs the action at the i^{th} trial;

$n(N) = \sum_{i=1}^{N} V(i) :$ the number of actions performed by X out of totally N trials;

We assume that X's behaviors in the past N trials and in the future $(N+1)^{th}$ trial are governed by the same Bernoulli distribution as

$$Pr(V(i) = 1|\theta) = \theta; \quad Pr(V(i) = 0|\theta) = 1 - \theta; \text{ for } i = 1, 2, \cdots, N+1, \tag{10}$$

where θ is the unknown parameter for the probability of X performing the action at each trial. Here, $Pr(\cdot)$ denotes the probability. We assume that $V(i)$ are independent for different i's. Then the distribution to observe $n(N) = k$ follows Binomial distribution

$$Pr(n(N) = k|\theta) = \binom{N}{k} \theta^k (1-\theta)^{N-k}. \tag{11}$$

The issue we would like to address is to estimate the probability $Pr(V(N+1) = 1)$, given the fact that k actions have been performed out of N trials. Then, we can calculate the trust value using (1). There are two possible approaches.

Approach 1: Estimate θ given the fact that k actions have been performed out of N trials.

It is well known that the minimum-variance unbiased estimator [22] for θ is $\hat{\theta} = k/N$, where $\hat{\theta}$ is the estimated value of θ. Then,

$$Pr(V(N+1)) = \hat{\theta} = k/N. \tag{12}$$

This approach is straightforward, and does not require the distribution of θ, i.e. $f(\theta)$. However, it does not accurately capture the "uncertainty" of $V(N+1)$. To see this, let one observation be $\{k = 2, N = 3\}$ and another observation be $\{k = 2000, N = 3000\}$. Obviously, A, who estimates

$Pr(V(N+1)) = 2/3$ in both cases by using (12), should be more certain about its result in the second case than that in the first case. Thus, there should be less uncertainty in the second observation than in the first observation.

Approach 2: Estimate $Pr(V(N+1) = 1|n(N) = k)$ by Bayesian approach.

From Bayesian equation, we have

$$Pr(V(N+1) = 1|n(N) = k) = \frac{Pr(V(N+1) = 1, n(N) = k)}{Pr(n(N) = k)} \tag{13}$$

where

$$Pr(n(N) = k) = \int_0^1 Pr(n(N) = k|\theta) f(\theta) d\theta. \tag{14}$$

$$Pr(V(N+1) = 1, n(N) = k) = \int_0^1 Pr(V(N+1) = 1, n(N) = k|\theta) \tag{15}$$

$$= \int_0^1 Pr(V(N+1) = 1|\theta) \cdot Pr(n(N) = k|\theta) f(\theta) d\theta \tag{16}$$

$$= \int_0^1 \theta \cdot Pr(n(N) = k|\theta) f(\theta) d\theta. \tag{17}$$

The deduction from (15) to (16) is because $V(N+1)$ and $n(N)$ are independent given θ for Binomial distribution in (11). Since there is no prior information, we assume that θ is uniformly distributed between between 0 and 1, i.e. $f(\theta) = 1$, for $\theta \in [0, 1]$. Then, using (11) we have

$$Pr(V(N+1) = 1|n(N) = k) \quad = \quad \frac{\int_0^1 \theta \times Pr(n(N) = k|\theta) f(\theta) d\theta}{\int_0^1 Pr(n(N) = k|\theta) f(\theta) d\theta} = \frac{k+1}{N+2}. \tag{18}$$

The second approach captures the uncertainty of the $V(N+1)$ given the observation. For example, the case with $k = 2000$ and $N = 3000$ will generate trust value larger than that in the case with $k = 2$ and $N = 3$. Moreover, when no observations are made, i.e. $k = 0, N = 0$, the probability value is $\frac{1}{2}$ and the trust value is 0. Clearly, if the ratio between k and N is fixed, the uncertainty is less for larger N values, which corresponds to more observations. Compared with Approach 1, Approach 2 has the advantage of capture the uncertainty more accurately for small values of k and N. In this work, we adopt Approach 2 and calculate the trust value as $T(Pr(V(N+1) = 1|n(N) = k))$, where $T(\cdot)$ is defined in (1).

In practice, node A often makes observations at different times. Let t_j denote the time when A make observations of node X, where $j = 1, 2, \cdots, I$. At time t_j, node A observes that node X performs the action k_j times upon the request of performing the action N_i times. We propose to calculate the trust value as follows:

$$P\{A:X,action\}=\frac{1+\sum_{j=1}^{I}\beta^{t_c-t_j}k_j}{2+\sum_{j=1}^{I}\beta^{t_c-t_j}N_j}, \tag{19}$$

where t_c represent the current time when this calculation is performed. We introduce $0 \leq \beta \leq 1$ as the forgetting factor, which describes that the observation made long times ago should carry less importance than the observation made more recently. The value of β depends on how fast the behavior of agents changes. When the agents' behaviors change fast, the observations made long times ago is not very useful for predicting the agents' future behaviors. In this case, β should be a small value, and vice versa. It is noted that when all observations are made long times ago, i.e. $t_c \gg t_I$, $P\{A:X,action\}$ approaches 0.5 and the trust value approaches to 0. Utilization of the forgetting factor provides a way to capture dynamic changes in the agents' behavior.

V. Security in Ad Hoc Network Routing

Securing routing protocols is a fundamental challenge for ad hoc network security [3]–[5]. Currently, most schemes that aim to secure ad hoc routing protocols focus on preventing attackers from entering the network through secure key distribution/authentication and secure neighbor discovery, such as [4], [23]. Those schemes, however, are not effective in situations where malicious nodes have gained access to the network, or some nodes in the network have been compromised. Therefore, it is important to develop mechanisms to monitor route disruption in ad hoc networks and adjust the route selection dynamically. In this section, we use the proposed trust models to improve ad hoc routing protocols and discuss their potential usage for malicious node detection.

In particular, for ad hoc routing, we investigate the trust value associated with two actions: forwarding packets and making recommendations. Briefly speaking, each node maintains its trust record associated with these two actions. When a node (source) wants to establish a route to the other node (destination), the source first tries to find multiple routes to the destination. Then the source tries to find the packet-forwarding trustworthiness of the nodes on the routes from its own trust record or through requesting recommendations. Finally the source selects the trustworthy route to transmit data. After the transmission, the source node updates the trust records based on its observation of route quality. The trust records can also be used for malicious node detection. All above should be achieved in a distributed manner.

In the rest of the section, we first address a procedure for obtaining trust recommendations in ad

hoc networks without establishing routes between the source node and the recommenders. Then, we present how to calculate and update the packet-forwarding trust and recommendation trust based on the observations. Finally, the complete scheme is described with a briefly discussion on malicious node detection and route selection.

A. Obtaining Trust Recommendations

Requiring trust recommendation in ad hoc networks often occurs in the circumstance where communication channels between arbitrary entities are not available. In this section, we will discuss the procedures for requesting trust recommendations and responding to such requests in ad hoc networks.

For requesting trust recommendations, we assume that node A wants to establish trust relationships with a set of nodes $\mathbf{B} = \{B_1, B_2, \cdots\}$ about action act, and A does not have valid trust record with $\{B_i, \forall i\}$. These trust relationships, denoted by $\{A : B_i, act\},\ \forall i$, can be established through recommendations from other nodes.

Node A first checks its trust record and selects a set of nodes, denoted by $\hat{\mathbf{Z}}$, that have the recommendation trust values larger than a threshold. Although A only needs recommendations from $\hat{\mathbf{Z}}$ to calculate trust values of $\mathbf{B}$ associated with act, A may ask for recommendations from a larger set of nodes, denoted by $\mathbf{Z}$, for two reasons. First, node A does not necessarily want to reveal the information about whom it trusts because the malicious nodes may take advantage of this information. Second, if node A establishes trust with $\mathbf{B}$ through direct interaction later, node A can use the recommendations it collects to update the recommendation trust of the nodes in $\mathbf{Z}$. This is an important way to establish or update recommendation trust. Thus, $\mathbf{Z}$ should contain not only the nodes in $\hat{\mathbf{Z}}$, but also the nodes with which A wants to update/establish recommendation trust relationship. Next, node A sends a trust recommendation request (TRR) message to its neighbors that in node A's transmission range. The TRR message should contain the IDs of nodes in set $\mathbf{B}$ and in set $\mathbf{Z}$. In order to reduce overhead, the TRR message also contains the maximal concatenation levels, denoted by Max_transit, and time-to-live (TTL). Each time a node asks further trust recommendations, the value of Max_transit is reduced by one. Node A waits time TTL for replies. In addition, $transmit\text{-}path$ is used to record delivery history of the TRR message

such that the nodes who receive the TRR message can send their recommendations back to A. Procedure 1 describes this scheme in details.

Upon receiving an unexpired TRR message, the nodes that are not in **Z** simply forward the TRR message to their neighbors; the nodes in **Z** either send trust values back to A or ask their trusted recommenders for further recommendations. In addition, the nodes in **Z** may not respond to the TRR message if they do not want to reveal their trust records to A when, for example, they believe that A is malicious. In particular, suppose node X is in **Z**. When X receives an unexpired TRR message, if X has the trust relationship with some of $\{B_i\}'s$, X sends its recommendation back to A. If X does not have trust relationship with some of $\{B_i\}'s$, X generates a new TRR message by replacing **Z** with the recommenders trusted by X and reducing the value of Max_transit by one. If Max_transit > 0, the revised TRR message is sent to X's neighbors. X also sends A corresponding recommendation trust values needed for A to establish trust propagation paths. If the original TRR message has not expired, X will also forward the original TRR message to its neighbors. By doing so, the trust concatenations can be constructed. The detailed schemes of processing TRR messages is described in Procedure 2.

The major overhead of requesting trust recommendations comes from transmitting TRR messages in the network. Let c denote the overhead of transmitting one TRR messages before it expires, and n_p denote the number of recommenders selected by each node. The overhead of transmitting TRR messages is approximately $c\sum_{k=0}^{\text{Max\_transit}} {n_p}^k$, which increases exponentially with Max_transit. In practice, Max_transit should be a small number for two reasons. First, since uncertainty increases along the trust transit path, if a trust relationship is established through many hops of trust propagation, the trust value can be very close to 0, which is not very useful anyway. The second reason is to reduce overhead that increases exponentially with Max_transit.

B. Calculation/Update of Action Trust and Recommendation Trust in Ad hoc Networks

Next, we present the procedure of utilizing Approach 2 (in Section IV) to calculate and update trust records in ad hoc networks. Assume that node A would like to ask node C to transmit packets, while A does not have trust relationship with node C.

Before the transmission

- Node A receives the recommendation from node B, and node B says that $T\{B : C, \text{forward packet}\} = T_{BC}$.
- Previously, node B has made recommendation to A for N_r times. Among those recommendations, A believes that B has made k_r "good recommendations". The definition of "good recommendations" is application dependent. Node A calculates the recommendation trust of B based on B's previous recommendations using equation (18). That is, $P\{A : B, \text{making recommendation}\} = \frac{k_r+1}{N_r+2}$ or $T\{A : B, \text{making recommendation}\} = T(\frac{k_r+1}{N_r+2})$.
- Then, A calculates the trust in C about packet forwarding through the concatenation propagation using equation (3) or (8). Let T^r_{AC} denote the calculated $\{A : C, \text{forward packet}\}$ before the transmission.

After the transmission

- Node A observes that C forwards k packets out of total N packets. A calculates $T\{A : C, \text{forward packet}\}$ using equation (18) or (19). Let T^a_{AC} denote the current trust value of $\{A : C, \text{forward packet}\}$, which is established/updated after the transmission.
- Then, node A updates the recommendation trust of node B as follows. If $|T^a_{AC} - T^r_{AC}| \leq$ threshold, node A believes that B has made good recommendation, and increases the value of k_r by 1 and increases the value of N_r by 1. If $|T^a_{AC} - T^r_{AC}| >$ threshold, node A believes that B has made bad recommendation, and increases the value of N_r by 1 while maintaining the value of k_r. A can update the recommendation trust based on the new values of k_r and N_r.

C. Proposed Scheme

In this section, we describe the details of the ad hoc routing scheme using the proposed trust models. First of all, each node in ad hoc network maintains a *trust record*, a *recommendation buffer*, and an *observation buffer*, which are described as follows.

- The entries of the trust record have the format of

$$\{subject,\ agent,\ action,\ trust\_value,\ probability\_value,\ t_{est}\}, \tag{20}$$

which describes the trust relationship $\{subject : agent, action\} = trust\_value$ established at time t_{est}, where $trust\_value = T(probability\_value)$. In the trust record of node A, the $subject$ field is always A because the trust record is established only through direction interaction.

- The entries of the recommendation buffer also have the same format as that in (20), but different meanings. The recommendation buffer of A describes that A receives the recommendation at time t_{est} from the $subject$, in which the $subject$ claimed $T\{subject : agent, action\} = trust\_value$. The $subject$ can only make recommendation based on its own trust record (i.e. directly interaction with the $agent$). In addition, when making recommendations, the $subject$ modifies trust values based on the current time and the time when its interaction with the $agent$ took place.
- Since it is not necessary to update the trust values immediately after an observation is made, each node maintains an observation buffer that contains the new observations. After an observation is used to establish/update trust relationship, it is removed from the buffer.

The flow chart of the proposed scheme is shown in Figure 4. The major blocks are explained in details as follows.

- Route discovery: Before node A can communicate with node D in ad hoc networks, routes between A and D should be established. Thus, A performs on-demand routing to find several possible routes to D. Let $\{S_i\}$ denote the nodes on all possible routes.
- Node A first checks its own trust record. If A cannot find a trust record for S_i or the trust value for S_i is below a certain threshold, node A puts S_i in set $\mathbf{B}$. Then, node A performs Procedure 1 to request recommendations for $\mathbf{B}$.
- Node A puts the received recommendations in the recommendation buffer, and constructs a trust propagation graph based on its own trust records and the recommendation buffer. Based on the trust graph, node A calculates the trust values for the nodes in $\mathbf{B}$.
- Among all possible routes, node A would like to choose a route that has the best quality. Let $\{n_i, \forall i\}$ represent the nodes on a particular route R. Let p_i represent $P\{A : n_i, \text{forward packet}\}$, where A is the source. The quality of route R is calculated as $\prod_i p_i$.
- During the transmission, node A makes the observations associated whether nodes forward packets and whether the nodes' true behaviors agree with the recommendations that A obtained from other nodes. All these observations are put into the observation buffer.
- Node A performs malicious nodes detection periodically to update its own list of malicious

nodes. In this work, we perform malicious node detection based on the trust value of two actions: forwarding packet and making recommendations. Let $P\{A : X_i, \text{forward packet}\} = P_i^f$ and $P\{A : X_i, \text{make recommendations}\} = P_i^r,\ \forall i$. On a 2D plot, each node is represented by a dot located at $[P_i^f, P_i^r]$. With enough observations, good nodes and malicious nodes should form clusters on this 2D plot, which can be used to separate good and malicious nodes. Such 2D plots will be shown in the simulation section.

- Node A monitors packet drop ratio of the entire route. When the packet drop ratio becomes smaller than a threshold, A will initiate a new round of route discovery. Before node A selects the new route, trust records are updated. Therefore, node A learns from previous experiences. If the transmission is finished, node A updates its trust record.

VI. Simulations

A. Malicious Node Detection

We first investigate the establishment of trust record in a simple system that reveals important insights of trust propagation and the effects of various attack models. The system is setup as follows. In each time interval, which is n time units long, each node selects another node to transmit packets. Assume that node A selects node X. If the trust value $\{A : X, \text{forward packet}\}$ is smaller than a threshold, node A will ask for recommendations about node X using the procedures described in Section V-A. Then, node A asks X to forward n packets and the data rate is 1 packet per time unit. In this simple system, we assume that node A can observe how many packets that X has forwarded. This assumption will be explained in the next paragraph. Next, node A updates its trust record using the procedure in Section V-B. In this system, if a malicious node decides to attack node A, it drops the packets from node A with packet drop ratio randomly selected between 0 and 40%, and/or sends recommendations to node A with trust values randomly picked from 0 to 1. Three types of malicious nodes are considered. Type 1 drops packets only, type 2 makes wrong recommendations only, and type 3 does both. No collusion is considered. For good nodes, the packet drop ratio is between 90% and 100%, and they make honest recommendations. Other simulation parameters are Max_transit $= 1$, $\mathbf{Z}$ is chosen as all nodes, and the forgetting factor is $\beta = 0.999$.

In practice, if X is A's neighbors, A can monitor X's transmission [3] and observe the number of packets forwarded by X. If X is not A's neighbor, A has to obtain this observation based on other nodes' reports. For example, when A detects abnormal route disruption, node A can ask each node on the path of packet transmission to report the number of packets that they received from the previous hop and the number of packets that they have forwarded to the next hop. If the reports are consistent, the source node believes these reports. If the reports are not consistent, the source can easily identify a small set of nodes containing the lying nodes, as long as the number of malicious nodes is not very large. The detection of fault reports is easier than the detection of malicious packet dropping. To avoid complicating this simple system, we have the assumption that A can observe the number of packets forwarded by X for this set of simulations.

We show three simulation results to demonstrate that distributed users can detect malicious nodes by using the proposed scheme. The first simulation shows the process for the malicious node detections. The second simulation shows the records of distributed users. The third simulation shows that the scheme can track the changes of the malicious behaviors and adaptively update the trust records.

In the first simulation, we have $N = 100$ total number of nodes. Among them, 24 nodes are malicious. 8 nodes for type 1, type 2, and type 3, respectively. In Figure 5, we show the trust record of one good node at different times. Here S is the simulation time. We plot the probability value of forward-packet trust vs. probability value of recommendation trust of all other nodes in this good node's trust record. When the number of observations is small, most of the nodes are with probability of 0.5 in either forward packet trust or recommendation trust. This is because this node has no experience with many others. With more observations, good nodes form a cluster that is close to the up-right corner and this cluster becomes tighter and tighter. Three types of malicious behaviors are clearly shown and can be differentiated. Type 1 nodes locate in the right-lower area, type 2 nodes locate in the left-up, and type 3 nodes are in the right-lower area.

It is important to point out that bad nodes do not necessarily form prominent clusters. There are two reasons. First, the trust values of bad nodes are reduced after they perform some malicious behaviors. With lower trust values, the chance for bad nodes to be on the routes or provide

recommendations becomes smaller. Thus, good nodes often do not have many bad experiences with malicious nodes, which is desirable because the damage caused by malicious nodes is limited. Second, malicious nodes have various behaviors. For example, some nodes may drop all packets, while others drop small potion of packets passing through them. The malicious behaviors in nature will not form very tight clusters.

In the second simulation, we have a total of 20 nodes. Among them, 3 nodes are malicious. Specifically, node 1 drops packets only, node 2 provides bad recommendations only, and node 3 does both. Figure 6 shows the trust of packet forwarding and making recommendations among distributed users for two different cases. In the first case, the malicious nodes attack all other nodes. In the second case, the bad nodes are only malicious to half of the users. In the figure, the element on the i^{th} row and j^{th} column represents the trust of the i^{th} user to the j^{th} user. The brighter the color, the higher the trust. Obviously the trust to the user itself is always 1. From Figure 6 (a), we can see that user 1, 2, and 3 are clearly differentiated from others. That is, most good nodes develop low negative trust values for user 1, 2, and 3 according to their malicious behaviors.

In the second case shown in Figure 6 (b), good nodes also develop negative trust values for malicious nodes. It is important to mention that when the malicious nodes only perform badly for half of users, the packet-forwarding trust values are similar as those in the first case. However, they can hurt others recommendation trusts. As shown in Figure 6 (b), the node 1-10 think node 11-20 do not give good recommendations and vise versa. We can make three points here. First, the recommendation trusts of malicious nodes are still significantly lower than that of good nodes. We can still perform malicious node detection. Second, node 1-10 will not give higher weights to the recommendations from node 11-20, which has positive effects on improving network throughput. Third, if good nodes can share their opinions through broadcasting (which is not discussed in this paper) , they can easily detect inconsistent behaviors of malicious nodes.

In the third simulation, we have a total of 40 nodes. At the beginning, we have 4 malicious nodes dropping packets. Every time when S increases by 3000, 4 more nodes become malicious. Here, S is the simulation time index. So we have 4, 8, 12, and 16 malicious nodes for the four stages when S equals to 3000, 6000, 9000, and 12000, respectively. In Figure 7, we show the average

packet-forward trust among users vs. user index. We highlight the changing of the trusts by drawing lines connecting the trust values in the current stage and the trust values in the previous stage. We can see that the four new malicious nodes are detected, and the proposed scheme can adaptively track network changes.

B. Network Throughput Improvement

We use an event-driven simulator to simulate mobile ad hoc networks. The physical layer assumes a fixed transmission range model, where two nodes can directly communicate with each other successfully only if they are in each other's transmission range. The MAC layer protocol simulates the IEEE 802.11 Distributed Coordination Function (DCF) [24]. DSR [25] is used as the underlying routing protocol. We use a rectangular space of size 1000m by 1000m. The total number of nodes is 50, and the maximum transmission range is 250m. There are 50 traffic pairs randomly generated for each simulation. For each traffic pair, the packet arrival time is modelled as a Poisson process, and the average packet inter-arrival time is 1 second. The size of each data packet after encryption is 512 bytes. Among all the ROUTE REQUESTs with the same ID received by a node A, A will only broadcast the first request if it is not the destination, and will send back at most 5 ROUTE REPLYs if it is the destination. The maximum number of hops on a route is restricted to be 10.

In the simulations, each node moves randomly according to the random waypoint model [25] with a slight modification: a node starts at a random position, waits for a duration called the pause time that is modeled as a random variable with exponential distribution, then randomly chooses a new location and moves towards the new location with a velocity uniformly chosen between 0 and $v_{max} = 10$ meters/second. When it arrives at the new location, it waits for another random pause time and repeats the process. The average pause time is 300 seconds.

We change the total number of malicious nodes from 1 to 11. In this implementation, the malicious nodes perform gray hole attack, i.e., randomly drop 65-75% packets passing through them. Three systems are compared: (1) baseline scheme that does not build or utilize trust record; (2) the system using entropy-based model for trust recommendations; and (3) the system using probability-based model for trust recommendations. Figure 8 shows the average packet drop ratios of good nodes. The simulation time is 1000sec. We can see that malicious nodes can significantly

degrade the performance of the baseline system. Even with 4 attackers (8% of total nodes), the packet drop ratio can be as high as 25%. Obviously, using the proposed mechanism to build and utilize trust records can greatly improve the performance. In particular, it takes more than 11 attackers (24% of total nodes) to cause 25% average packet drop ratio. In addition, the performances of probability-based and entropy-based models are similar. It is important to point out that the results shown in Figure 8 is for a very short simulation time, where the trust records are built based on very limited observations. Within such as short simulation time, the good nodes and bad nodes are not well separated on the 2D trust plots (similar as the up-left plot in Figure 5), and malicious node detection mechanism is not activated yet. Even under this condition, the proposed scheme still shows performance gain in Figure 8, which is due to the route selection mechanism based on the proposed trust models.

VII. Conclusion

In this paper, we present an information theoretic framework for trustworthiness evaluation in distributed networks. Four axioms are developed to address the meaning of trust and establish trust relationship through third parties. Based on these axioms, the level of trustworthiness can be quantitatively determined based on observations and through propagations. Two models that govern concatenation and multipath propagation of trust are developed. The proposed framework is suitable for a variety of applications in distributed networks. In this work, we demonstrate the usage of the proposed models in ad hoc network to assist malicious node detection and route selection. The simulation results demonstrate that the malicious nodes can be detected and the types of their malicious behaviors can be identified. In addition, with the trust recommendations and trust records, the chances of malicious node being on the routes are greatly reduced. As a result, the improvement in the packet drop ratio is observed. As a summary, this work provides the theoretical bases of trustworthiness evaluation as well as addresses practical implementations when applying the theories in ad hoc networks.

References

[1] L. Zhou and Z. Haas, "Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, no. 6, pp. 24–30, Nov./Dec. 1999.
[2] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proceedings of Mobicom 2000*, Aug 2000, pp. 275 – 283.

[3] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of MobiCom 2000*, August 2000, p. 255265.

[4] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *Proceedings of MobiCom 2002*, Sep 2002.

[5] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," in *Proceedings of the 2003 ACM workshop on Wireless security*, Sep 2003, pp. 30 – 40.

[6] M. K. Reiter and S. G. Stubblebine, "Resilient authentication using path independence," *IEEE Transactions on Computers*, vol. 47, no. 12, pp. 1351–1362, December 1998.

[7] W. Stallings, *Protect Your Privacy, A Guide for PGP Users*, Prentice Hall, 1995.

[8] U. Maurer, "Modelling a public-key infrastructure," in *Proceedings 1996 European Symposium on Research in Computer Security(ESORICS' 96), volume 1146 of Lecture Notes in Computer Science*, 1996, pp. 325–350.

[9] A. Jsang, "An algebra for assessing trust in certification chains," in *Proceedings of the Network and Distributed Systems Security (NDSS'99) Symposium*, 1999.

[10] R. Levien and A. Aiken, "Attack-resistant trust metrics for public key certification," in *Proceedings of the 7th USENIX Security Symposium*, January 1998, pp. 229–242.

[11] D.W. Manchala, "Trust metrics, models and protocols for electronic commerce transactions," in *Proceedings of the 18th IEEE International Conference on Distributed Computing Systems*, May 1998, pp. 312 – 321.

[12] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of 12th International World Wide Web Conferences*, May 2003.

[13] R. Guha, R. Kumar, P. Raghavan, and A.T. Propagation, "Propagation of trust and distrust," in *Proceedings of International World Wide Web Conference*, 2004.

[14] P. R. Zimmermann, *The Official PGP User's Guide*, MIT Press, 1995.

[15] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, May 1996, pp. 164–173.

[16] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *Proceedings of 1997 New Security Paradigms Workshop, ACM Press*, 1998, pp. 48–60.

[17] A. Herzberg, Y. Mass, J. Michaeli, D. Naor, and Y. Ravid, "Access control meets public key infrastructure or: Assigning roles to strangers," in *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, May 2000, pp. 2–14.

[18] D. Clarke, J.-E. Elien, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest, "Certificate chain discovery in spki/sdsi," *Journal of Computer Security*, vol. 9, no. 4, pp. 285–322, 2001.

[19] D. Gambetta, "Can we trust trust?," in *in Gambetta, Diego (ed.) Trust: Making and Breaking Cooperative Relations, electronic edition, Department of Sociology, University of Oxford*, 2000, pp. 213–237.

[20] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, 1991.

[21] D.L. Hall and S.A.H. Mcmullen, *Mathematical Techniques in Multisensor Data Fusion*, Artech Hous INC, 2004.

[22] H. Vincent Poor, *An Introduction to Signal Detection and Estimation*, Springer, 2nd edition, 1994.

[23] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," in *Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference(CNDS 2002)*, Jan 2002.

[24] IEEE Computer Society LAN MAN Standards Committee, "Wireless lan medium access control (mac) and physical layer (phy) specifications, ieee std 802.11-1007," The Institue of Electrical and Electrics Engineers.

[25] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks, mobile computing," *In Moible Computing, edited by Tomasz Imielinski and Hank Korth, Kluwer Academic Publishers*, pp. 153–181, 1996.

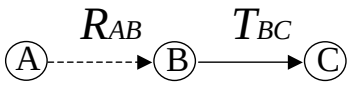


Fig. 1. Concatenation Trust Propagation

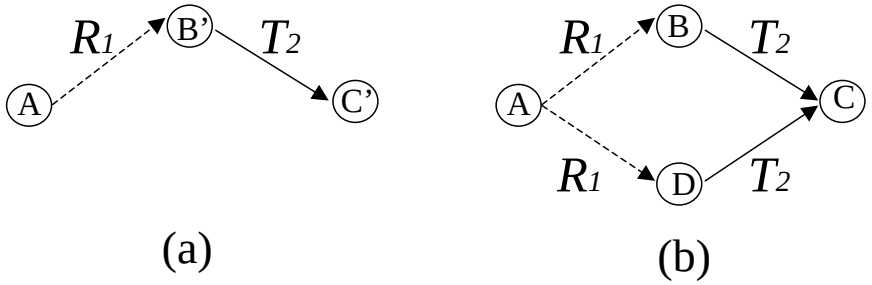


Fig. 2. Combing Trust Recommendations

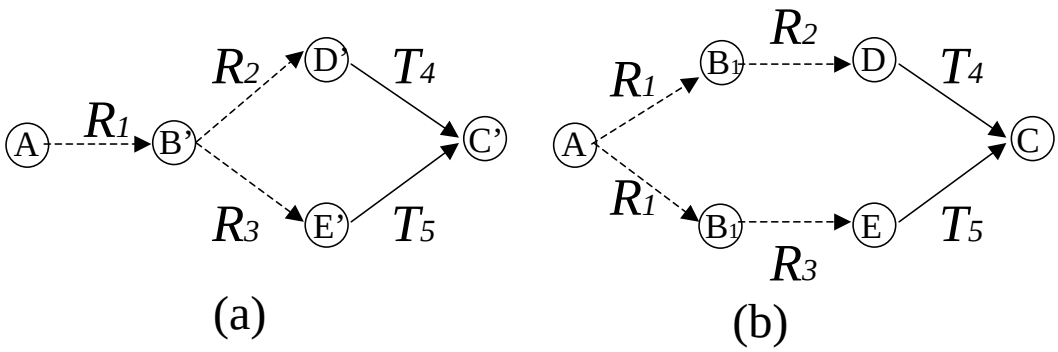


Fig. 3. One Entity Provides Multiple Recommendations

Procedure 1 Sending Trust Requesting Algorithm

1. Node A selects a set of trusted recommenders $\hat{\mathbf{Z}}$. Each node in $\hat{\mathbf{Z}}$ has recommendation trust value above a certain threshold.
2. Node A selects another set $\mathbf{Z}$. $\mathbf{Z}$ contains $\hat{\mathbf{Z}}$ and is often a larger set than $\hat{\mathbf{Z}}$.
3. Node A sends the following TRR message to its neighbors

$$\{\text{requestID}, A, \mathbf{B}, \text{act}, \mathbf{Z}, \text{Max\_transit}, \text{TTL}, \text{transmit-path}\}$$

4. Node A waits for recommendation messages until a predetermined time.

Procedure 2 Node X Processing TRR Messages

if (TRR not expired) & (X has not received this TRR before) & ($X \notin \mathbf{Z}$) **then**
  X forwards the TRR to its neighbors.
end if
if (TRR not expired) & (X has not received this TRR before) & ($X \in \mathbf{Z}$) **then**
  for every element $B_i \in \mathbf{B}$ **do**
    X checks its trust record for B_i.
    if $\{X : B_i, act\} = T_{X,B_i}$ is found in X's trust record and $|T_{X,B_i}|$ is larger than a threshold, **then**
      X sends the trust value T_{X,B_i} back to A.
    else
      X puts B_i in a set $\mathbf{B_x}$.
    end if
  end for
  if $\mathbf{B_x}$ is not empty & Max_transit> 1, **then**
    X searches its trust record for recommenders $\hat{\mathbf{Z}_\mathbf{x}} = \{Z_k^x\}$ such that $\{X : Z_k^x, act_r\} >$ threshold and $Z_k^x \notin \mathbf{Z}$. If $\hat{\mathbf{Z}_\mathbf{x}}$ is not empty, X selects a set of nodes $\mathbf{Z_x}$. The set $\mathbf{Z_x}$ contains $\hat{\mathbf{Z}_\mathbf{x}}$ and is often a larger set than $\hat{\mathbf{Z}_\mathbf{x}}$.
    X generates a new TRR message by making the following changes to the original TRR: (1) replace $\mathbf{Z}$ by $\mathbf{Z_x}$; and (2) reduce Max_transit by 1.
    X sends the new and original TRR messages to its neighbors.
    X sends its recommendation trust value of $\hat{\mathbf{Z}_\mathbf{x}}$ back to A.
  end if
end if

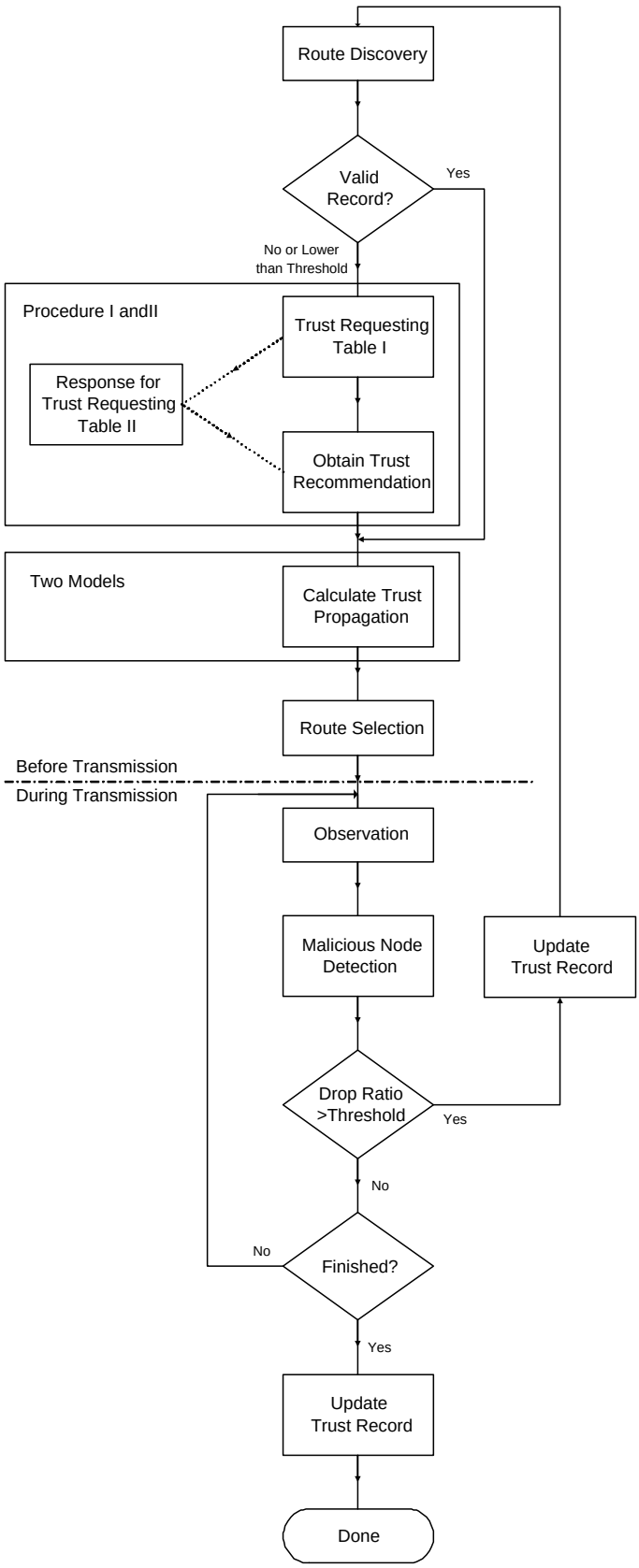


Fig. 4. Flow Chart of Proposed Scheme

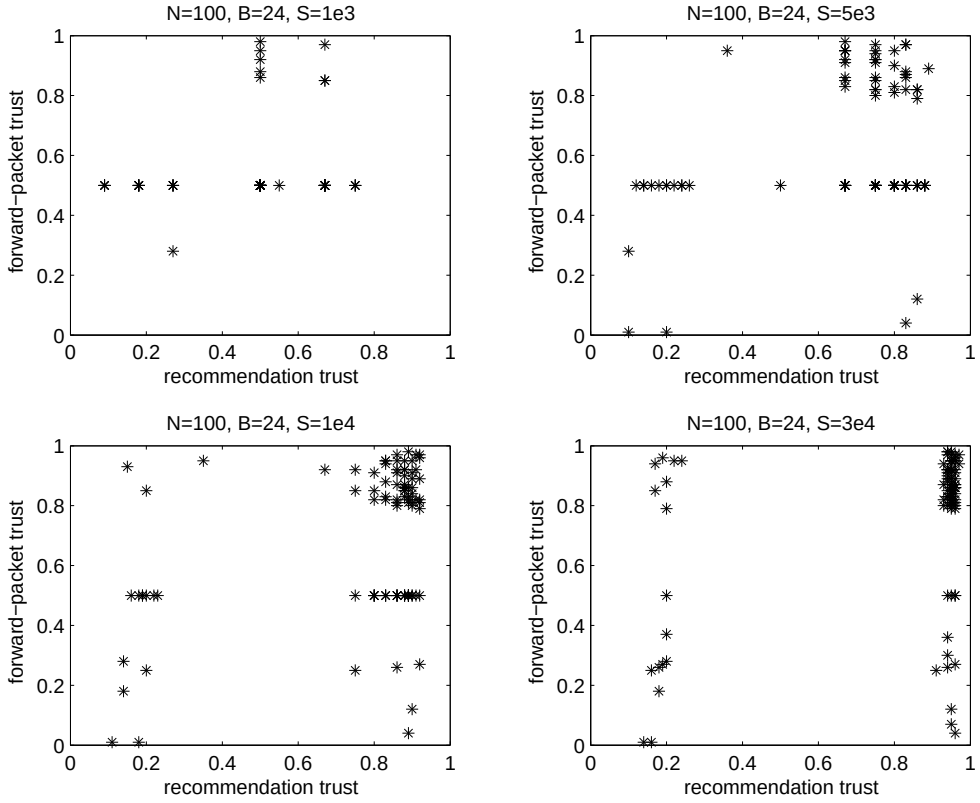
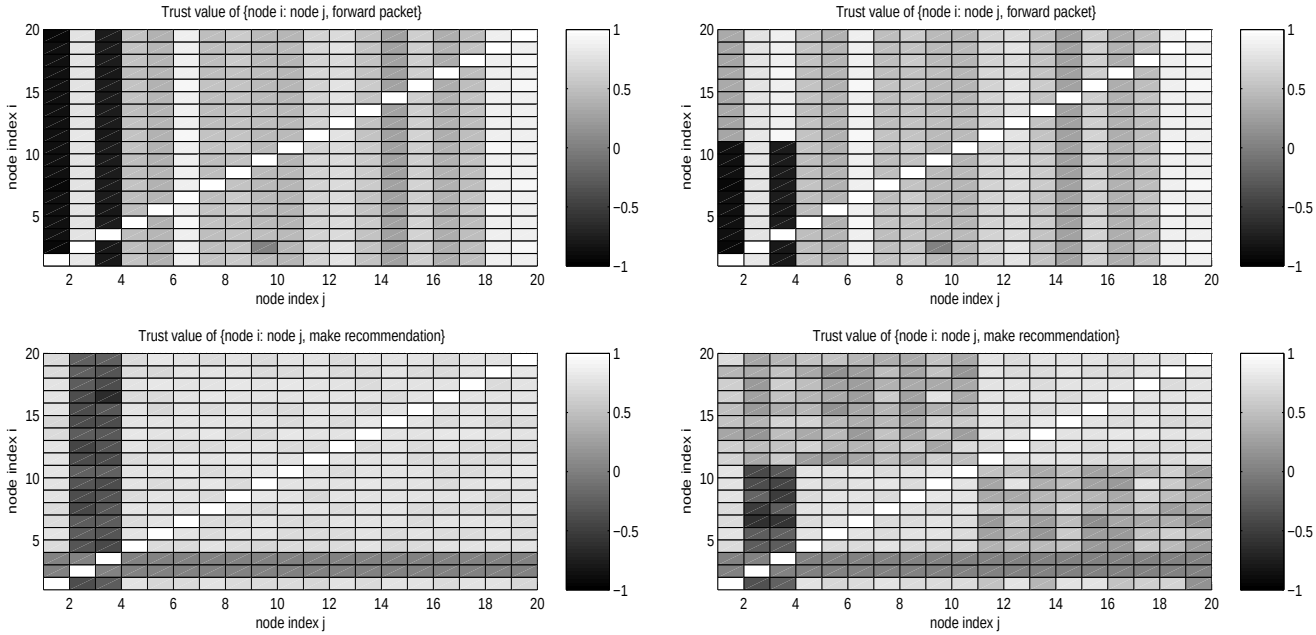


Fig. 5. Trust Record of a Good Node



(a) (b)

Fig. 6. Trust Records of 20 Nodes with 3 Malicious Nodes (a) Malicious to All Users (b) Malicious to 50% Users

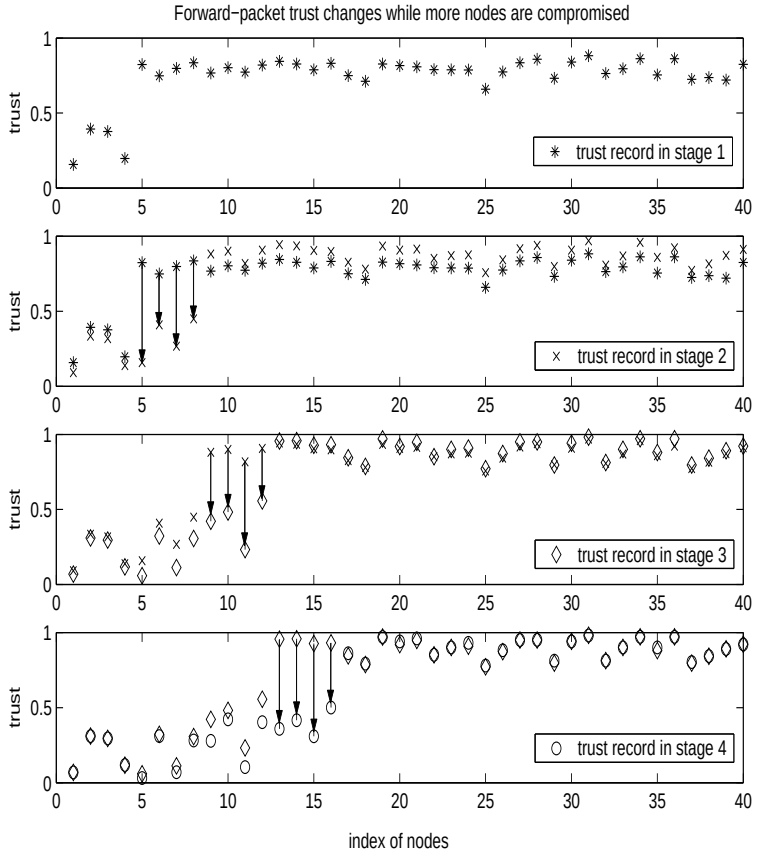


Fig. 7. Dynamic Behaviors of Malicious Node Detection

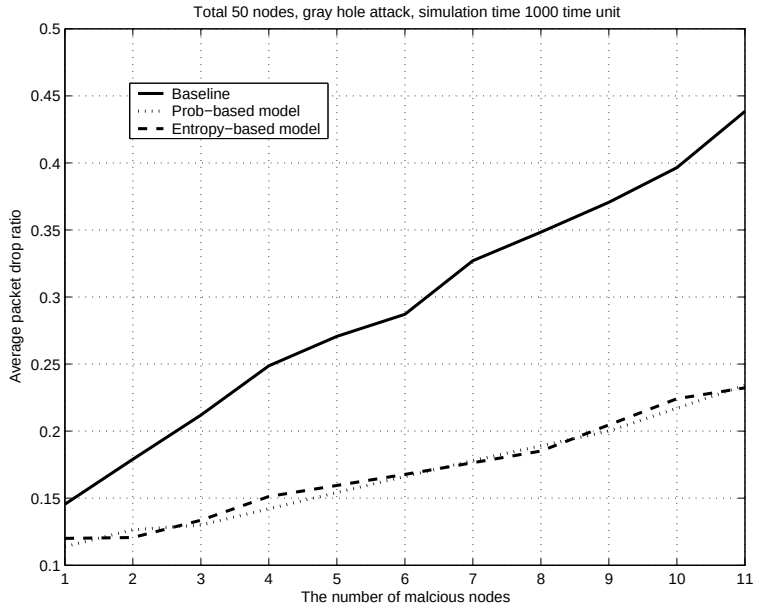


Fig. 8. Average Packet Drop Ratio with Different Number of Malicious Nodes