Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks

Yan Lindsay Sun, *Member, IEEE*, Wei Yu, *Student Member, IEEE*, Zhu Han, *Member, IEEE*, and K. J. Ray Liu, *Fellow, IEEE*

***Abstract*—The performance of ad hoc networks depends on cooperation and trust among distributed nodes. To enhance security in ad hoc networks, it is important to evaluate 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 framework, trust is a measure of uncertainty with its value represented by entropy. We develop four Axioms that address the basic understanding of trust and the 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 observation. 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 about other nodes. Simulations show that the proposed trust evaluation system can significantly improve the network throughput as well as effectively detect malicious behaviors in ad hoc networks.**

***Index Terms*—Ad hoc networks, security, trust modeling and evaluation.**

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 performance 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 routing schemes and to perform malicious node detection.

The research on trust evaluation has been extensively performed for a wide range of applications, including public key authentication [6]–[15], electronics commerce [16]–[18], peer-to-peer networks [19], [20], and ad hoc and sensor networks [21]–[23]. However, there are still many challenges need to be addressed.

Trust Definition: Although definitions of trust have been borrowed from the social science literature, there is no clear consensus on the definition of trust in distributed computer networks. Trust has been interpreted as reputation, trusting opinion, probability [24], etc.

Trust Metrics: As a nature consequence of the confusion in trust definition, trust has been evaluated in very different ways. Some schemes employ linguistic descriptions of trust relationship, such as in PGP [19], PolicyMaker [12], distributed trust model [14], trust policy language [15], and SPKI/SDSI public-key infrastructure [13]. In some other schemes, continuous or discrete numerical values are assigned to measure the level of trustworthiness. For example, in [6], an entity's opinion about the trustworthiness of a certificate is described by a continuous value in [0, 1]. In [23], a two-tuple in $[0,1]^2$ describes the trust opinion. In [8], the metric is a triplet in $[0,1]^3$, where the elements in the triplet represent belief, disbelief, and uncertainty, respectively. In [14], discrete integer numbers are used.

Currently, it is very difficult to compare or validate these trust metrics because a fundamental question has not been well understood. What is the physical meaning of trust? We need trust metrics to have clear physical meanings, for establishing the connection between trust metrics and observation (trust evidence) and justifying calculation/policies/rules that govern calculations performed upon trust values.

Quantitative Trust Models: Many trust models have been developed to model trust transit through third parties. For example, the simplest method is to sum the number of positive ratings and negative ratings separately and keep a total score as the positive score minus the negative score. This method is used in eBay's reputation forum [17]. In [8], subjective logics is used to assess trust values based on the triplet representation of trust. In [16], fuzzy logic provides rules for reasoning with linguistic trust metrics. In the context of the "Web of Trust," many trust models are built upon a graph where the resources/entities are nodes and trust relationships are edges, such as in [6] and [7]. Then, simple mathematic, such as minimum, maximum, and weighted average, is used to calculate unknown trust values through concatenation and multipath trust propagation. In [25]–[27], a Bayesian model is used to take binary ratings as input and compute reputation scores by statistically updating beta probability density functions.

Although a variety of trust models are available, it is still not well understood what fundamental rules the trust models must

Manuscript received October 14, 2004; revised August 15, 2005.

Y. L. Sun is with the Department of Electrical and Computer Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: yansun@ele.uri.edu).

W. Yu, Z. Han, and K. J. R. Liu are with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20740 USA (e-mail: weiyu@glue.umd.edu; hanzhu@glue.umd.edu; kjrliu@umd.edu).

Digital Object Identifier 10.1109/JSAC.2005.861389

follow. Without a good answer to this question, the design of trust models is still at the empirical stage.

We approach the trust evaluation problem from a definition of trust given by Gambetta in [24]. 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 develop axioms that address the basic rules for establishing trust through a third party (concatenation propagation) and through recommendations from multiple sources (multipath propagation). Based on these axioms, we develop techniques that calculate trust values from observation and design two models that address the concatenation and multipath trust propagation problems in ad hoc networks. The proposed models are then 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 ad hoc networks. Individual users 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 baseline scheme without trust evaluation, the proposed scheme can select the route with higher recommended quality so that the packet dropping rates are greatly reduced.

In this paper, we also briefly discuss various attacks on trust evaluation systems. This discussion includes some well-known attacks as presented in [28]–[31] and a new attack strategy resulting from the study in this paper. In addition, tradeoffs among implementation overhead, nodes mobility, and effectiveness of recommendation mechanism in trust evaluation are discussed.

The rest of this 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 these two models satisfy all the axioms. In Section IV, we investigate how to establish trust relationship based on observation. In Section V, the proposed models are applied in ad hoc networks to assist route selection in routing protocols and to perform malicious node detection. Simulation results are shown in Section VI, followed by discussion in Section VII. Conclusions are drawn in Section VIII.

II. Basic Axioms

In this section, we will explain the meaning of trust and present four axioms for establishment of trust relationship. In this paper, we interpret 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 [redacted], the second entity is called the [redacted]. 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 agent and the subject.

Axiom 1: Uncertainty is a Measure of Trust: The concept of trust describes the certainty of whether the agent will perform an action in the subject's point of view. Let $T\{subject : agent, action\}$ denote the trust value of the 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. It is important to note that this probability is not absolute, but the opinion of a specific subject. Thus, different subjects can assign different probability values for the same agent and the same action. Information theory states that entropy is a nature measure for uncertainty [32]. Thus, we define the entropy-based trust value as

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

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 $0 \le p < 0.5$ and positive for $0.5 < p \le 1$. 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 $P\{subject : agent, action\}$.

considering

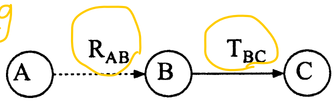

Fig. 1. Concatenation trust propagation.

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.

Trust relationship can be represented by a directional graph shown in Fig. 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 that perform the action do not necessarily make correct recommendations [14]. The second condition states that the recommendations from entities with low trust values should not be used. The second condition makes the trust propagation in distributed networks resilient to malicious entities who can manipulate their recommendations in order to cause maximal 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 built upon others' recommendations is no more than the recommenders' trust and the trust in 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 Fig. 2, node A_1 establishes trust with C_1 through one concatenation path, and A_2 establishes trust with C_2 through two same paths. Let $T_{A_1C_1} = T\{A_1 : C_1, action\}$ and $T_{A_2C_2} = T\{A_2 : C_2, action\}$. The mathematical representation of Axiom 3 is

$$T_{A_2C_2} \geq T_{A_1C_1} \geq 0, \text{ for } R_1 > 0 \text{ and } T_2 \geq 0$$
$$T_{A_2C_2} \leq T_{A_1C_1} \leq 0, \text{ for } R_1 > 0 \text{ and } T_2 < 0$$

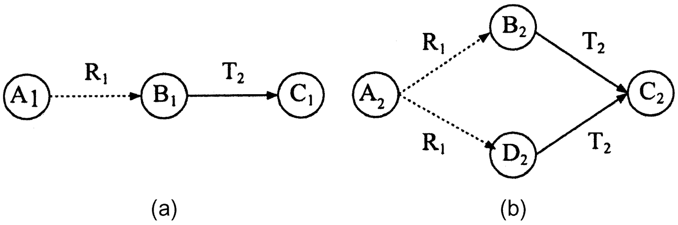

Fig. 2. Combining trust recommendations.

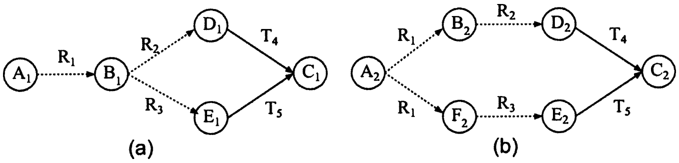

Fig. 3. One entity provides multiple recommendations.

where $R_1 = T\{A_1; B_1, action_r\} = T\{A_2 : B_2, action_r\} = T\{A_2 : D_2, action_r\}$ and $T_2 = T\{B_1, C_1, action\} = T\{B_2 : C_2, action\} = T\{D_2 : C_2, action\}$. 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 Fig. 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 $T_{A_1C_1} = T\{A_1 : C_1, action\}$ denote the trust value established in Fig. 3(a), and $T_{A_2C_2} = T\{A_2 : C_2, action\}$ denote the trust value established in Fig. 3(b). The Axiom 4 says that

$$T_{A_2C_2} \geq T_{A_1C_1} \geq 0, \quad \text{if } T_{A_1C_1} \geq 0$$
$$T_{A_2C_2} \leq T_{A_1C_1} \leq 0, \quad \text{if } T_{A_1C_1} < 0$$

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 the 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 Fig. 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 Fig. 2 (the third Axiom), we obtain $T_{AC} = T_{AC'} = R_1 T_2$. Thus, Axiom 3 is satisfied with equality. When applying the model to the cases in Fig. 3, we can prove that $T_{AC} = T_{AC'} = R_1(R_2^2 T_4 + R_3^2 T_5)/(R_2 + R_3)$. 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 Fig. 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, node A can calculate p_{ABC} as

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

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

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

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 is worth mentioning that the above propagation model can also be viewed as binary symmetry channel (BSC) model [32], The physical meaning of BSC is as follows. When node B claims 1, node A would think that 1 happens with probability 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, node A would think that 0 happens with probability p and 1 happens with probability $1 - p$. The concatenation of two BSC models also generates the probability expression in (8).

For the multipath case, as shown in Fig. 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 in [33] 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 Observation

The problem we address in this section is to obtain the trust value from observation. 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, node 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 ith trial;

$n(N) = \sum_{i=1}^{N} V(i)$ — the number of actions performed by X out of total N trials.

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

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

where θ, an unknown parameter, is 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:

$$Pr\,(n(N) = k|\theta) = \binom{N}{k} \theta^k (1-\theta)^{N-k}. \tag{10}$$

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. Here, we assume that every action leads to the same consequence. 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 [34] for θ is $\hat{\theta} = k/N$, where $\hat{\theta}$ is the estimated value of θ. Then

$$Pr\,(V(N+1)) = \hat{\theta} = \frac{k}{N}. \tag{11}$$

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's exam two cases; (1) $\{k = 2, N = 3\}$; and (2) $\{k = 2000, N = 3000\}$. When using (11), $Pr(V(N+1))$ is estimated as 2/3 for both cases. Intuitively, if the ratio between k and N is fixed, the uncertainty should be less for larger N values. The subject who had made more observation should be more certain about the agent than in the case that the subject had made less observation. Thus, there, should be less uncertainty in the second case than in the first case.

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

From Bayesian equation, we have

$$Pr\,(V(N{+}1){=}1|n(N){=}k){=}\frac{Pr\,(V(N{+}1){=}1, n(N){=}k)}{Pr\,(n(N){=}k)} \tag{12}$$

where

$$Pr\,(n(N) = k) = \int_0^1 Pr\,(n(N) = k|\theta)\, f(\theta) d\theta. \tag{13}$$

$$\begin{aligned} &Pr\,(V(N+1) = 1, n(N) = k) \\ &= \int_0^1 Pr\,(V(N+1) = 1, n(N) = k|\theta) \\ &= \int_0^1 Pr\,(V(N+1) = 1|\theta) \cdot Pr\,(n(N) = k|\theta)\, f(\theta) d\theta \\ &= \int_0^1 \theta \cdot Pr\,(n(N) = k|\theta)\, f(\theta) d\theta. \end{aligned} \tag{14}$$

In the above derivation, we use the assumption that $V(N+1)$ and $n(N)$ are independent given θ for Binomial distribution in (10). Since there is no prior information about θ, we assume that θ is uniformly distributed between 0 and 1, i.e., $f(\theta) = 1$, for $\theta \in [0, 1]$. Then, using (10), we have

$$\begin{aligned} &Pr\,(V(N+1) = 1|n(N) = k) \\ &= \frac{\int_0^1 \theta \times Pr\,(n(N) = k|\theta)\, f(\theta) d\theta}{\int_0^1 Pr\,(n(N) = k|\theta)\, f(\theta) d\theta} \\ &= \frac{k+1}{N+2}. \end{aligned} \tag{15}$$

When using the second approach, the case $\{k = 2000, N = 3000\}$ will generate trust value 0.6666 , which is a little bit smaller than the trust value for the case $\{k = 2, N = 3\}$. Moreover, when no observation is made, i.e., $k = 0, N = 0$, the probability value is 1/2 and the trust value is 0, which is also very reasonable. Compared with Approach 1, Approach 2 has the advantage of capturing the uncertainty more accurately, especially 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 observation at different time instances. Let t_j denote the time when A make observation 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:

$$P\{A : X, action\} = \frac{1 + \sum_{j=1}^{I} \beta^{t_c - t_j} k_j}{2 + \sum_{j=1}^{I} \beta^{t_c - t_j} N_j} \tag{16}$$

where t_c represents the current time when this calculation is performed. We introduce $0 \leq \beta \leq 1$ as the remembering factor, which describes that the observation made long times, ago should carry less importance than the observation made recently. The value of β depends on how fast the behavior of agents changes. When the agents' behaviors change fast, the observation 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 observation is made long times ago, i.e., $t_c \gg t_1$, $P\{A : X, action\}$ approaches 0.5 and the trust, value approaches to 0. Utilization of the remembering 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] and [35]. 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 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 trust values 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 a trustworthy route to transmit data. After the transmission, the source node updates the trust record based on its observation of route quality. The trust record can also be used for malicious node detection. All above should be achieved in a distributed manner.

Next UP

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 observation. Finally, the complete scheme is described with a brief discussion on malicious node detection and route selection.

A. Obtaining Trust Recommendations

Requesting 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 $\mathbf{B} = \{B_i, B_2, \cdots\}$ 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 ID's 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-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.

```
Procedure 1 Sending Trust Requesting Algorithm
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.
Node $A$ selects another set $\mathbf{Z}$. $\mathbf{Z}$ contains $\hat{\mathbf{Z}}$ and is often a larger set than $\hat{\mathbf{Z}}$.
Node $A$ sends the following TRR message to its neighbors
    $\{\text{requestID}, A, B, \text{act}, \mathbf{Z}, \text{Max\_transit}, \text{TTL}, \text{transmit\_path}\}$.
Node $A$ waits for recommendation messages until a predetermined time.
```

Upon receiving an unexpired TRR message, the nodes that are not in $\mathbf{Z}$ simply forward the TRR message to their neighbors; the nodes in $\mathbf{Z}$ either send trust values back to A or ask their trusted recommenders for further recommendations. In addition, the nodes in $\mathbf{Z}$ 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 $\mathbf{Z}$. 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 $\mathbf{Z}$ with the recommenders trusted by X and reducing the value of Max_transit by one. If $\text{Max\_transit} > 0$, 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 scheme of processing TRR messages is described in Procedure 2.

```
Procedure 2 Node $X$ Processing TRR Messages
if (TRR not expired) and ($X$ has not received this TRR before) and
```

```
    ($X \in \mathbf{Z}$) then
    $X$ forward the TRR to its
    neighbors.
else if (TRR not expired) and ($X$
    has not received this TRR before)
    and ($X \in \mathbf{Z}$) 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 and
    $\text{Max\_transit} > 1$, then
        $X$ searches its trust record
        for recommenders $\hat{\mathbf{Z}}_\mathbf{x} = \{Z^x_k\}$ such
        that $\{X : Z^x_k, act_r\} >$ threshold
        and $Z^x_k \notin \mathbf{Z}$. 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
```

The major overhead of requesting trust recommendations comes from transmitting TRR messages in the network. Let c denote the overhead of transmitting one TRR message 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 recommendations 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 (15). That is, $P\{A : B, \text{making recommendation}\}= (k_r + 1)/(N_r + 2)$ or $T\{A : B, \text{making recommendation}\}= T(k_r + 1/N_r + 2)$.
- Then, A calculates the trust in C about packet forwarding through the concatenation propagation using (3) or (8). Let T^r_{AC} denote the calculated $T\{A : C, \text{forward packet}\}$ before the transmission.

After the Transmission:

- Node A observes that C forward k packets out of total N packets. Node A calculates $T\{A : C, \text{forward packet}\}$ using (15) or (16). 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. Node 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 in the trust record have the format of
$$\{subject,\ agent,\ action\ trust\_value,\ probability\_value,\ t_{est}\}$$
which describes the trust relationship $\{subject : agent, action\}$ established at time t_{est}, with trust value $trust\_value$ and $trust\_value = T(probability\_value)$. In the trust record of node A, the $subject$ field is always A because the trust record is established only through direct interaction.
- The entries in the recommendation buffer have the same format as those in the trust record, 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., direct 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 observation is made, each node maintains an observation buffer that contains the new observation. After an observation entry is used to establish/update trust relationship, it is removed from the buffer.

The flow chart of the proposed scheme is shown in Fig. 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 record 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 observation associated with whether nodes forward packets and whether the nodes' true behaviors agree with the recommendations that A obtained from other nodes. All observation is 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 values 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 two-dimensional (2-D) plot, each node is represented by a dot located at, $[P_i^f, P_i^r]$. With enough observation, good nodes and malicious nodes should form clusters on this 2-D plot, which can be used to separate good and malicious nodes. Such 2-D 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 record is updated: Therefore node A learns from previous experience. After the transmission is finished, node A updates its trust record.

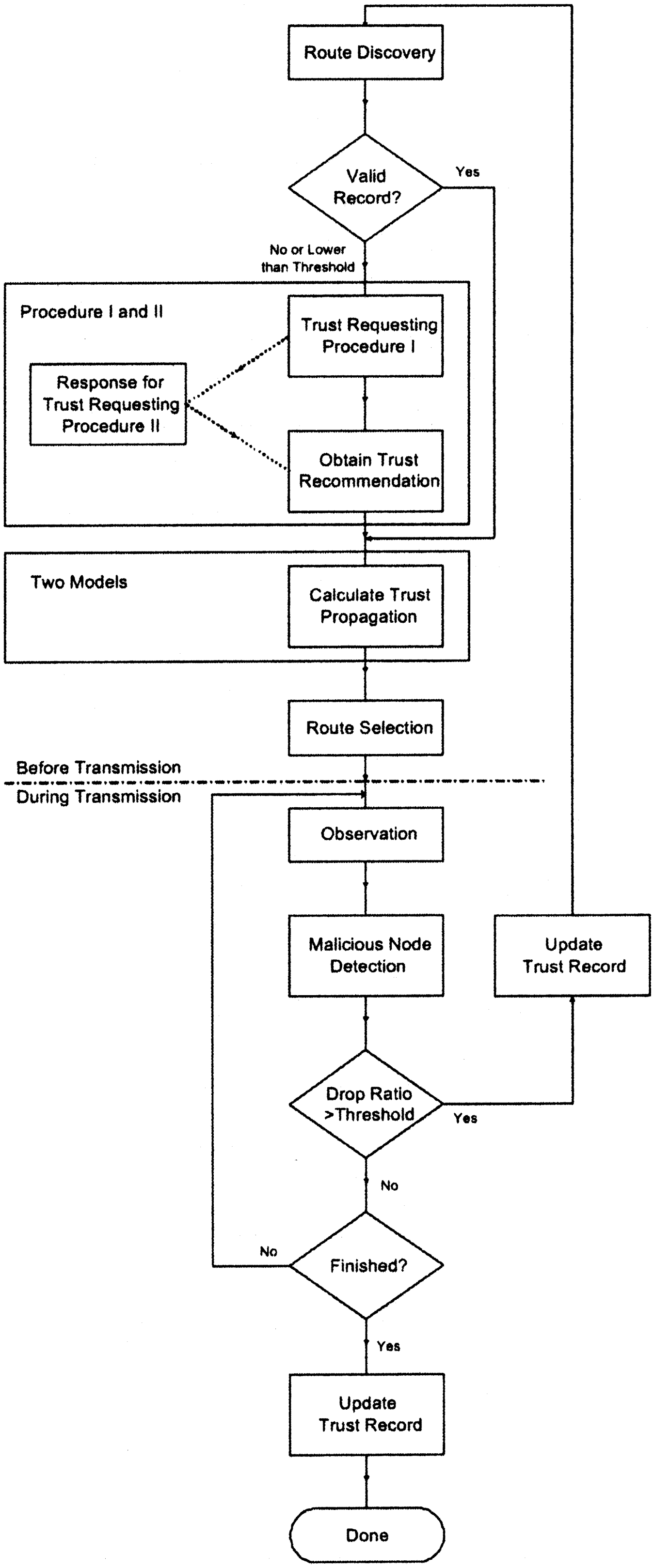

Fig. 4. Flow chart of proposed scheme.

VI. SIMULATIONS

A. Malicious Node Detection

We first investigate the establishment of trust records in a simple system that reveals important insight of trust propagation and the effects of various attack models. The system is set up 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. For good nodes, the packet drop ratio is between 0% and 10%, and they make honest recommendations. Other simulation parameters are $\text{Max\_transit} = 1$, $\mathbf{Z}$ is chosen as all nodes, and the remembering factor is $\beta = 0.999$.

In practice, if X is A's neighbors, A can monitor X's transmission [3], [36] 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, such as the scheme reported in [37]. 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 faulty 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. Eight nodes for types 1, 2, and 3, respectively. In Fig. 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 versus probability value of recommendation trust of all other nodes in this good node's trust record. At the beginning of the simulation, 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 much experience with others. With more observation, 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-upper area, 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 a small potion of packets passing through them. The malicious behaviors in nature will not form very tight clusters.

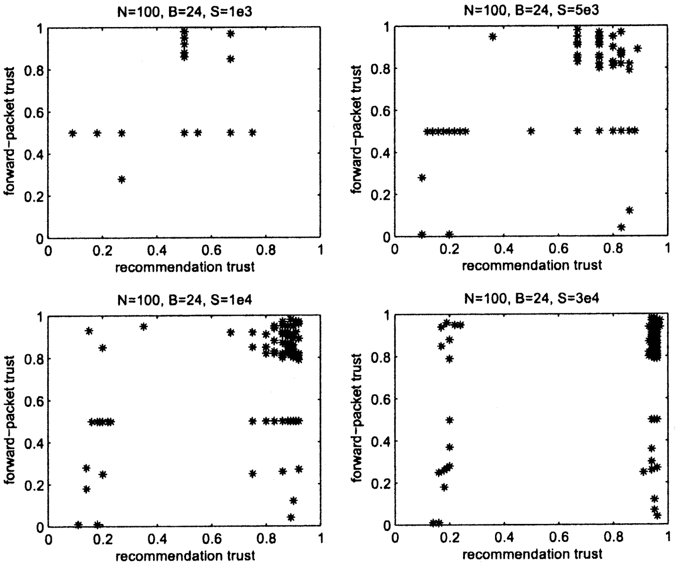

Fig. 5. Trust record of a good node.

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

In the second case shown in Fig. 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 to 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 Fig. 6(b), nodes 1–10 think nodes 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, nodes 1–10 will not give higher weights to the recommendations from nodes 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

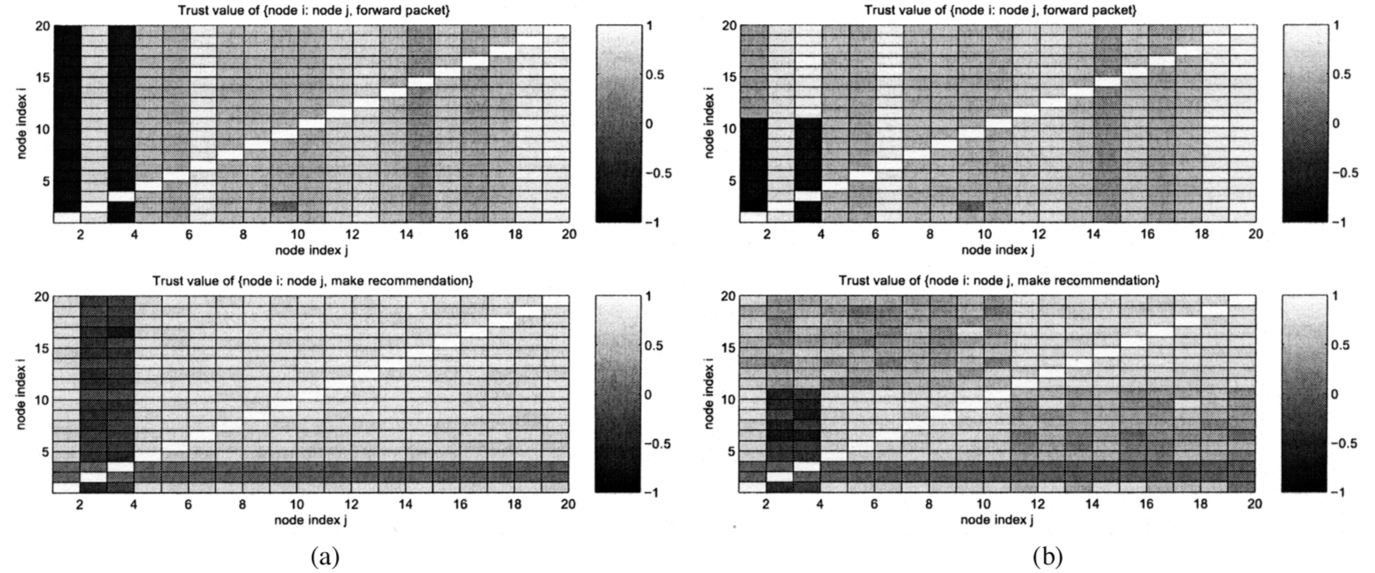

Fig. 6. Trust records of 20 nodes with 3 malicious nodes being (a) malicious to all users (b) malicious to 50% users.

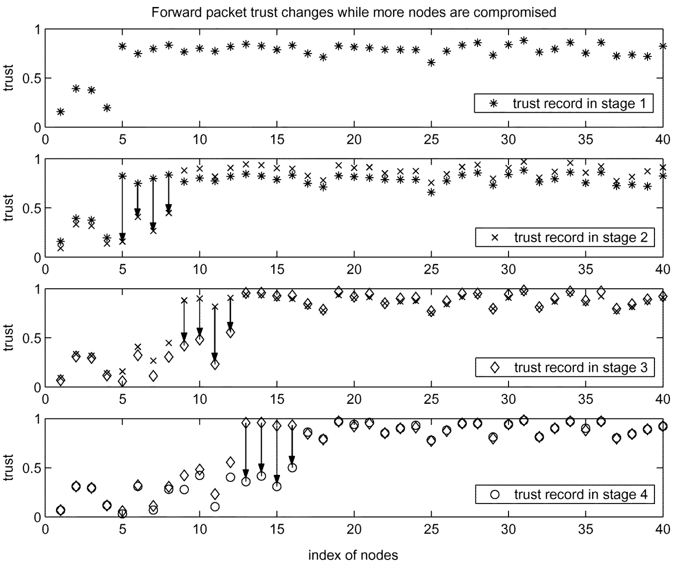

Fig. 7. Dynamic behaviors of malicious node detection.

can easily detect inconsistent behaviors of malicious nodes. In this experiment, since all malicious nodes attack the same subset of users, this is a coordinated attack. On the other hand, in the previous experiments where malicious nodes launch the gray hole attack to everyone, the attack is not coordinated.

In the third simulation, we have a total of 40 nodes. At the beginning, we have four malicious nodes dropping packets. Every time when S increases by 3000, four 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 5 equals to 3000, 6000, 9000, and 12 000, respectively. In Fig. 7, we show the average packet-forward trust among users versus user index. We highlight the changes 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

In this set of simulations, the mobile ad hoc network is simulated as follows. 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 medium access control (MAC) layer protocol simulates the IEEE 802.11 distributed coordination function (DCF) [38]. Dynamic source routing (DSR) [39] is used as the underlying routing protocol. We use a rectangular space of size 1000 m × 1000 m. The total number of nodes is 50, and the maximum transmission range is 250 m. There are 50 traffic pairs randomly generated for each simulation. For each traffic pair, the packet arrival time is modeled as a Poisson process, and the average packet interarrival time is 1 s. 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 five ROUTE REPLYs if it is the destination. The maximum number of hops on a route is restricted to be ten.

In the simulations, each node moves randomly according to the random waypoint model [39] 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 toward the new location with a velocity uniformly chosen between 0 and $v_{\max} = 10$ m/s. When it arrives at the new location, it waits for another random pause time and repeats the process. The average pause time is 300 s.

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 records; 2) the system using entropy-based model for trust recommendations; and 3) the system using probability-based model for trust recommendations. Fig. 8 shows the average packet drop ratios of good

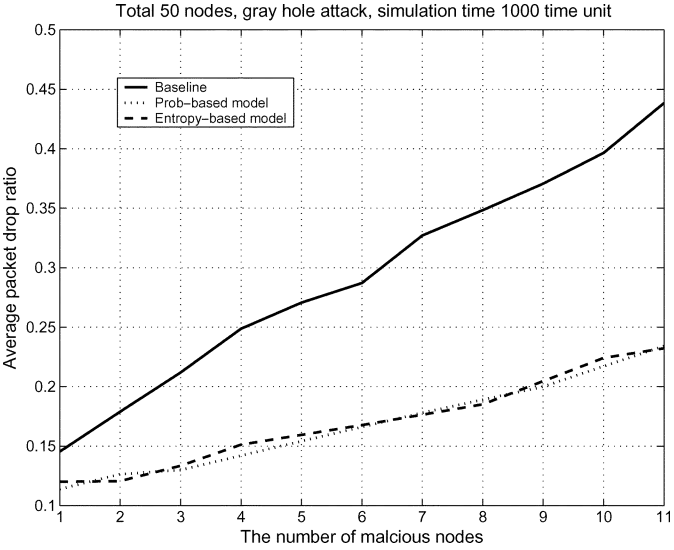

Fig. 8. Average packet drop ratio with different number of malicious nodes.

nodes. The simulation time is 1000 s. We can see that malicious nodes can significantly degrade the performance of the baseline system. Even with four 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 Fig. 8 is for a very short simulation time, where the trust records are built based on very limited observation. Within such a short simulation time, the good nodes and bad nodes are not well separated on the 2-D trust plots [similar as the upper-left plot in Fig. 6(a)], and malicious node detection mechanism is not activated yet. Even under this condition, the proposed scheme still shows performance gain in Fig. 8, which is due to the route selection mechanism based on the proposed trust models.

VII. Discussion

A. Attacks on Trust Evaluation

Since trust evaluation can effectively improve network performance and detect malicious entities, trust evaluation itself is an attractive target for attackers.

A well-known attack is the bad-mouthing attack [28], that is, malicious parties providing dishonest recommendations to frame up good parties and/or boost trust values of malicious peers. The defense against the bad-mouthing attack has been considered in the design of the proposed trust evaluation system. First, the action trust and the recommendation trust records are maintained separately. Only the entities who have provided good recommendations previously can earn high recommendation trust. Second, according to the necessary conditions of trust propagation, only the recommendations from the entities with positive recommendation trust can propagate. Third, the fundamental axioms limit the recommendation power of the entities with low recommendation trust.

Trust evaluation may also be vulnerable to the Sybil attack and the newcomer attack. If a malicious node can create several faked IDs, the trust evaluation system suffers from the Sybil attack [29], [30]. Here, the faked IDs can share or even take the blame, which otherwise should be given to the malicious node. If a malicious node can easily register as a new user, the trust evaluation suffers from the newcomer attack [31]. Here, malicious nodes can easily remove their bad history by registering as a new user. The defense against the Sybil attack and newcomer attack does not rely on the design of trust evaluation system, but the authentication and access control mechanisms, which make registering a new ID or a faked ID difficult.

Besides these known attacks, a malicious node may also reduce the effectiveness of trust evaluation through other methods. For example, as illustrated in Section VI-A, coordinated malicious nodes can reduce good nodes' recommendation trust by attacking only a subset of users and creating conflicting opinions among good nodes. While the focus of this paper is to lay the foundation of trust evaluation with meaningful trust metrics, we do not investigate all possible attacks in this paper.

B. Tradeoffs Among Recommendation Effectiveness, Overhead, and Mobility

Recommendation mechanism is an important component in any trust evaluation systems. The effectiveness of recommendation is closely related with communication overhead and mobility.

In order to establish trust between A and X, through trust propagation, there must exist other nodes who have previous interaction with X and recommendation trust relationship with A. In this section, we call A the requester, X the target, and the other nodes who are in the middle of trust propagation paths the recommenders. We also use P_p to represent the probability that A can establish trust propagation paths to X.

When the TRR messages propagate further away from the requester, it is more likely to establish trust propagation paths between the requester and the target. Of course, this also means higher communication overhead. As discussed in Section V-A, we use the expiration time (TTL) and Max_transit to control how far the TRR messages can propagate from the requester. While the expiration time determines the overhead of broadcasting one TRR message, Max_transit determines the number of TRR messages generated from the initial request. In general, a longer expiration time and a larger Max_transit value lead to a higher P_p value and larger overhead, and *vise versa.*

Mobility has three major impacts on trust evaluation systems in ad hoc networks.

First, at the beginning of trust evaluation when few interactions have taken place in the network, higher mobility requires higher overhead. Due to the usage of the expiration time and Max_transit, the requester tends to establish recommendation trust with nearby nodes. When the requester moves to a new neighborhood, it may not have recommendation trust with its new neighbors. Recommenders may also move further away from the requester. Therefore, with high mobility, if we would like to maintain the P_p value at the beginning, a larger expiration time should be chosen to make the requester reach its trusted recommenders.

Second, after the trust evaluation system has been running for a long time, a mobile node has had opportunities to interact with many other nodes. Compared with a stationary node, a mobile node has a larger probability to interact with recommenders. In this case, the overhead of requesting recommendations for a node with high mobility can be reduced.

Third, high mobility can make the task of malicious node detection harder. The honest nodes can have high packet drop ratio when they move fast. Thus, when the malicious node detection criterion is the packet-forwarding trust, higher mobility can lead to higher false alarm rates when the detection rate is fixed.

VIII. 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 observation and through propagation. 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 trust evaluation methods 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 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 reduction 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 Netw. Mag.*, vol. 13, no. 6, pp. 24–30, Nov./Dec. 1999.

[2] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proc. MobiCom*, Aug. 2000, pp. 275–283.

[3] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. MobiCom*, Aug. 2000, pp. 255–265.

[4] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *Proc. MobiCom*, Sep. 2002, pp. 12–23.

[5] ——, "Rushing attacks and defense in wireless ad hoc network routing protocols," in *Proc. ACM Workshop on Wireless Security*, Sep. 2003, pp. 30–40.

[6] U. Maurer, "Modeling a public-key infrastructure," in *Proc. Eur. Symp. Res. Comput. Security*, vol. 1146, Lecture Notes in Computer Science, 1996, pp. 325–350.

[7] M. K. Reiter and S. G. Stubblebine, "Resilient authentication using path independence," *IEEE Trans. Comput.*, vol. 47, no. 12, pp. 1351–1362, Dec. 1998.

[8] A. Jøsang, "An algebra for assessing trust in certification chains," in *Proc. Netw. Distrib. Syst. Security Symp.*, 1999.

[9] R. Levien and A. Aiken, "Attack-resistant trust metrics for public key certification," in *Proc. 7th USENIX Security Symp.*, Jan. 1998, pp. 229–242.

[10] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proc. 12th Int. World Wide Web Conf.*, May 2003, pp. 640–651.

[11] R. Guha, R. Kumar, P. Raghavan, and A. T. Propagation, "Propagation of trust and distrust," in *Proc. Int. World Wide Web Conf.*, 2004, pp. 403–412.

[12] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proc. IEEE Symp. Security and Privacy*, May 1996, pp. 164–173.

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

[14] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *Proc. 1997 New Security Paradigms Workshop*, 1998, pp. 48–60.

[15] A. Herzberg, Y. Mass, J. Michaeli, D. Naor, and Y. Ravid, "Access control meets public key infrastructure: Or assigning roles to strangers," in *Proc. IEEE Symp. Security and Privacy*, May 2000, pp. 2–14.

[16] D. W. Manchala, "Trust metrics, models and protocols for electronic commerce transactions," in *Proc. 18th IEEE Int. Conf. Distrib. Comput. Syst.*, May 1998, pp. 312–321.

[17] P. Resnick and R. Zeckhauser, "Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system," in *Advances in Applied Microeconomics: The Economics of the Internet and E-Commerce,*, vol. 11, In M. Baye, Ed., Nov. 2000, pp. 127–157. Elsevier.

[18] A. Jsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," Decision Support Systems, 2005.

[19] P. R. Zimmermann, *The Official PGP User's Guide*. Cambridge, MA: MIT Press, 1995.

[20] B. Yu, M. P. Singh, and K. Sycara, "Developing trust in large-scale peer-to-peer systems," in *Proc. 1st IEEE Symp. Multi-Agent Security and Survivability*, Aug. 2004, pp. 1–10.

[21] S. Buchegger and J. L. Boudec, "Performance analysis of the confidant protocol," in *Proc. ACM MobiHoc*, 2002, pp. 226–236.

[22] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," *Commun. Multimedia Security*, pp. 107–121, Sep. 2002.

[23] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *Proc. ACM Workshop Wireless Security*, Oct. 2004, pp. 1–10.

[24] D. Gambetta, "Can we trust trust?," in *Trust: Making and Breaking Cooperative Relations*, D. Gambetta, Ed. Oxford, U.K.: Dept. Sociology, Univ. Oxford, 2000, pp. 213–237.

[25] A. Jøsang and R. Ismail, "The beta reputation system," in *Proc. 15th Bled Electron. Commerce Conf.*, Jun. 2002.

[26] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," in *Proc. ACM Security for Ad-Hoc and Sensor Netw.*, 2004, pp. 66–67.

[27] S. Buchegger and J.-Y. Le Boudec, "The effect of rumor spreading in reputation systems in mobile ad-hoc networks," in *Proc. Wiopt*, 2003.

[28] C. Dellarocas, "Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems," in *Proc. ICIS*, 2000, pp. 520–525.

[29] J. R. Douceur, "The Sybil attack," in *Proc. 1st Int. Workshop on Peer-to-Peer Syst.*, 2002, pp. 251–260.

[30] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: Analysis and defenses," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2004, pp. 259–268.

[31] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara, "Reputation systems," *Commun. ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[32] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[33] D. L. Hall and S. A. H. Mcmullen, *Mathematical Techniques in Multisensor Data Fusion*. Norwood, MA: Artech House, 2004.

[34] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York: Springer-Verlag, 1994.

[35] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," in *Proc. SCS Commun. Netw. Distrib. Syst. Modeling and Simulation Conf.*, Jan. 2002, pp. 27–31.

[36] W. Yu and K. J. R. Liu, "Attack resistant cooperation stimulation in autonomous ad hoc networks," in *Proc. 2nd Ann. IEEE Commun. Soc. Conf. Sensor and Ad Hoc Commun. Netw.*, Dec. 2005, pp. 2260–2271.

[37] W. Yu, Y. Sun, and K. J. R. Liu, "HADOF: Defense against routing disruptions in mobile ad hoc networks," in *IEEE INFOCOM*, Mar. 2005, pp. 1251–1261.

[38] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-1007.

[39] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks, mobile computing," in *Mobile Computing*, T. Imielinski and H. Korth, Eds. Norwell, MA: Kluwer, 1996, pp. 153–181.

Yan Lindsay Sun (S'01–M'05) received the B.S. degree with highest honor from Beijing University, Beijing, China, in 1998, and the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, in 2004.

She is currently an National Science Foundation (NSF) ADVANCE Assistant Professor in the Electrical and Computer Engineering Department, University of Rhode Island, Kingston. Her research interests include network security and wireless communications and networking.

Dr. Sun is a member of the IEEE Signal Processing, Communication, and Computer Societies. She received the Graduate School Fellowship at the University of Maryland from 1998 to 1999, and the Excellent Graduate Award of Beijing University in 1998.

Wei Yu (S'04) received the B.S. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2000, and the M.S. degree in computer science from Washington University, St. Louis, MO, in 2002. Currently, he is working towards the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Maryland, College Park.

From 2000 to 2002, he was a Graduate Research Assistant at the Washington University. From 2002 to 2005, he was a Graduate Research Assistant with the Communications and Signal Processing Laboratory and the Institute for Systems Research, University of Maryland. His research interests include network security, wireless communications and networking, game theory, and wireless multimedia.

Zhu Han (S'01–M'04) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1997 and 2003, respectively.

From 1997 to 2000, he was a Graduate Research Assistant at the University of Maryland. From 2000 to 2002, he was an Engineer in the R&D Group of ACTERNA, Maryland. He is currently a Research Associate at the University of Maryland. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, and wireless multimedia.

Dr. Han is a member of the Technical Programming Committee for the IEEE International Conference on Communications (2004 and 2005), the IEEE Vehicular Technology Conference, Spring 2004, the IEEE Consumer Communications and Networking Conference (2005 and 2006), the IEEE Wireless Communications and Networking Conference (2005 and 2006), and the IEEE Globe Communication Conference 2005, as well as Session Chair of the IEEE Wireless Communications and Networking Conference (2004 and 2005), and the IEEE Globe Communication Conference 2005.

K. J. Ray Liu (F'03) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1983, and the Ph.D. degree in electrical engineering from the University of California at Los Angeles (UCLA), in 1990.

He is a Professor and Director of Communications and Signal Processing Laboratories of the Electrical and Computer Engineering Department and the Institute for Systems Research, University of Maryland, College Park. His research contributions encompass broad aspects of wireless communications and networking, information forensics and security, multimedia communications and signal processing, bioinformatics and biomedical imaging, and signal processing algorithms and architectures.

Dr. Liu is the recipient of numerous honors and awards including Best Paper awards from the IEEE Signal Processing Society, the IEEE Vehicular Technology Society, and EURASIP; IEEE Signal Processing Society Distinguished Lecturer, EURASIP Meritorious Service Award, and the National Science Foundation Young Investigator Award. He also received the Poole and Kent Company Senior Faculty Teaching Award from A. James Clark School of Engineering, and Invention of the Year Award, both from University of Maryland. He is Vice President—Publication and on the Board of Governor of the IEEE Signal Processing Society. He was the Editor-in-Chief of the *IEEE Signal Processing Magazine*, the founding Editor-in-Chief of *EURASIP Journal on Applied Signal Processing*, and the prime proposer and architect of IEEE Transactions on Information Forensics and Security.