Distributed Trust Management in Wireless Autonomic Networks

John S. Baras, Tao Jiang and George Theodorakopoulos
Institute for Systems Research,
Electrical and Computer Engineering Department,
and Computer Science Department
University of Maryland College Park

***Abstract*— As an important concept in network security, trust is interpreted as a set of relations among agents participating in the network activities. Trust relations are based on previous behaviors of agents as well as on trust documents. We present our results on distributed trust management in MANET. The trust information or evidence used to evaluate trustworthiness is provided by peers, i.e. the agents that form the network. We describe first a new trust document distribution scheme based on swarm intelligence. Then we describe various methods for distributed trust evaluation and the associated trust (and mistrust) spreading dynamics. Under such dynamics the whole network evolves as the local interactions iterate from isolated trust islands to a connected trust graph. Our interest is to discover rules and policies that establish trust-connected networks using only local interactions, to find the conditions under which trust spreads to a maximum set, as well as the parameters (e.g. topology type) that speed up or slow down this transition. We analyze the dynamics induced by local interaction rules using algebraic graph theory and methods inspired from the statistical physics of spin-glass materials. We describe and explain the phase transition phenomena that we have found in these evolutions. We model the interactions among agents as cooperative games and show that trust can encourage agents to collaborate. We also describe a model for trust evaluation that uses pairwise iterated graph games between the agents to create a trust reputation with evolution coupled to the game dynamics. Finally we present a new modeling framework for trust metric evaluation as linear iterations over ordered semirings, by treating such evaluations as path problems in MANET. This allows us to formulate problems of resilience of trust metrics and trust evaluation to attacks.**

I. Introduction

Trust is interpreted as a set of relations among entities participating in network activities [1]. In traditional networks, such as the Internet, sources of trust evidence are centralized control servers, such as trusted third parties (TTPs) and authentication servers (ASs). Those servers are trusted and available all the time. In contrast, wireless autonomic networks have neither fixed infrastructures, nor centralized control servers. In these networks, the sources of trust evidence are peers, i.e. the entities that form the network. We summarize the essential and unique properties of distributed trust management in autonomic networks as opposed to traditional centralized approaches:

- **uncertainty and incompleteness**: Trust evidence is provided by peers, and so it can be incomplete and even incorrect.
- **locality**: Trust information is exchanged locally through individual interactions.
- **distributed computation**: Trust evaluation is performed in a distributed manner.

Trust management involves collecting, analyzing and presenting trust-related evidence, and making assessments and decisions regarding trust relationships between entities in a network [2]. In this work, we investigate various components of trust management: direct and indirect trust evidence, trust evidence distribution and trust evaluation policies. Furthermore, self-organized networks rely on cooperation among participating agents. We study how trust affects their cooperation and enables the network to function normally.

In a distributed trust management system, pieces of trust evidence are stored in a distributed manner. In Sec. II, we propose a scheme which efficiently distributes trust evidence to places where they are mostly needed. Our scheme is inspired by the swarm intelligence paradigm, where agents mimic ants looking for their food – the trust evidence in this context. Our scheme is compared with another scheme based on Freenet, a distributed P2P file sharing system. The simulation results show that our scheme is more reliable and suitable for mobile environments.

By defining a trust evaluation rule based on local voting, we study how trust spreads throughout the whole network in Sec. III. We first study a simple deterministic voting rule. Our analytic results on its convergence show that trust can only be established if the number of headers in the network satisfies certain conditions. Furthermore, the uncertainty of trust is introduced in the local voting rule by modelling it as an iterated stochastic process. The convergence of this process and the stationary distribution at the steady state are provided. We further investigate the resulting trust values at the steady state. Our investigation gives several important conclusions, such as the choice of threshold and phase transition phenomena. Those results provide a way to properly design a feasible evaluation rule.

Autonomic networks rely on cooperation of participating nodes for almost all their functions. However, because such data forwarding consumes valuable (and scarce) battery power and channel bandwidth without receiving any direct gain, intermediate nodes would prefer not to cooperate. In order to form the necessary infrastructure that makes multi-hop com-

munication achievable, cooperation enforcement mechanisms are developed to cope with such selfish behavior of nodes in ad hoc networks in Sec. IV. We provide conditions for achieving collaboration among selfish users. Furthermore, the trust mechanism is introduced to promote cooperation and circumvent misbehaving nodes.

In Sec. V, we assume that trust is built up through repeated interactions of the users within the protocol that is in effect in the network. The next step is to use this trust in a transitive way, i.e. somebody can benefit from the interactions of that others have had in order to build "indirect" trust without direct interactions. We present a game theoretic model for building up trust values from interactions. Users are interacting strategically, i.e. they are trying to maximize their own personal gain. Through these interactions, their interacting partners can estimate their trustworthiness. The result is a game thoretic equilibrium from which no user can diverge without suffering a loss. In this sense, the final estimates are robust. Moreover, we present a linear system approach to aggregate these "direct" estimates. We use a non-conventional algebraic framework, the theory of semirings, to replace the regular multiplication and addition. We argue that this approach naturally fits the real problem we are trying to attack. Furthermore, it can be used as a general model for trust computation if different semiring operators are used.

II. Trust Evidence Distribution

In a distributed trust management system, pieces of trust evidence cannot be provided by a centralized server. Such a distributed manner of operation raises several problems. For instance, where should a user look for trust evidence he/she requests, and how could he/she efficiently obtain this evidence?

The problem of evidence distribution shares many characteristics of distributed peer-to-peer file sharing systems ([3], [4]). Inspired from P2P networks, Eschenauer [1] proposed a trust establishment scheme based on Freenet [4], where pieces of trust evidence are routed and searched using distributed hash table (DHT). However, trust evidence distribution is not a simple static "request routing" problem. First, the scheme must be adaptive to the mobility of nodes. Second is the security issue. Since malicious nodes can simply drop any evidence passing through them, the distribution scheme must carefully select nodes and paths which are reliable. The most important issue is to efficiently distribute trust evidence such that the evidence searching cost can be significantly reduced.

Based on the above concerns, we introduce an Ant-Based Evidence Distribution scheme (ABED) which arises from the swarm intelligence paradigm. The main principle behind the interaction in a swarm is called *stigmergy*, such as pheromone laying on the trails followed by ants. In our scheme, "ants" are sent out searching for their food – trust evidence. *Forward ants* are originated by requesters and choose next-hop nodes according to the "pheromone deposit" at each intermediate node. Upon discovery of the evidence, a backward ant is created. It takes the requested evidence and traces the reverse route of the forward ant. On its way it updates the pheromone deposit of intermediate nodes and caches a copy of the evidence. In the following, we discuss the important components of ABED.

- Hash function: similar as in [1], a global hash function is used to map evidence identities into values, which are called keys, and each key is linked to a particular node in the network.
- Pheromone deposit: denoted by Q_{ijd} for outgoing link (i, j) and target d. A higher Q-value represents higher quality. Q-values are updated when backward ants are received. Assuming q_{ijd} is the ratio of the number of backward ants received to the number of forward ants sent out through link (i, j) for the current request to d, the update rule is

$$Q_{ijd}(n+1) = (1-\gamma)Q_{ijd}(n) + \gamma q_{ijd} \qquad (1)$$

  where γ is a pre-defined parameter between $[0, 1]$. The first term captures the evaporation procedure of pheromone deposit.
- Routing table: each entry in the evidence routing table corresponds to the hash value of an evidence piece, and each value in the table, denoted as p_{ijd}, represents the probability of an ant (searching for target d) choosing j as the next hop at node i. p_{ijd} is a function of Q-values. The one we use is

$$p_{ijd}(n) = \frac{(\tau_{ij}(n))^{\alpha} (Q_{ijd}(n))^{\beta}}{Z_i(n)}, \qquad (2)$$

  where τ_{ij} is the quality of link (i, j) and Z_i is a normalization constant.

We simulated ABED using the discrete time network simulator ns-2. We compared ABED with the Freenet based scheme proposed in [1]. Our results show that ABED outperforms the Freenet-based scheme. In particular, ABED finds the best solution much faster, which is highly desired in mobile scenarios. Furthermore, the inherent property of multiple paths in the ant-based scheme makes it more resilient and reliable. More details of the ABED scheme and the simulation results can be found in [5].

III. Dynamics of Trust Evaluation

After obtaining the trust evidence, network users follow certain policies to evaluate the trustworthiness of their targets. This procedure is called trust evaluation or trust computation.

A. Network Model

We model a network as a directed graph $G(V, E)$[1]. A directed link from node i to node j, denoted as (i, j), corresponds to the *direct* trust relation that i has on j[2] and the weight on the link represents the degree of confidence i has on j, denoted as $c_{ij} \in [-1, 1]$. We define the neighbor set of

[1] G is called the *trust graph*, as opposed to the physical graph due to the communication constraints.

[2] A trust relation from i to j does not necessarily mean that i trusts j. Trust relations include distrust (i.e. negative opinions) as well.

node i, $\mathcal{N}_i$, as the set of nodes that are directly connected to i.

Nodes in the network are assumed to be either GOOD or BAD, denoted by $t_i = 1$ or -1 for node i. The vector $T = [t_1, \ldots, t_N]$ is called the *true* trust vector. Trust evaluation is to estimate the trustworthiness of nodes. Let vector $S = [s_1, \ldots, s_N]$ be the estimated trust vector. If $s_i = 1$, we call node i trusted, which is a subjective concept. The confidence value c_{ij} is the degree of confidence node i has on node j, where $c_{ij} \in [-1, 1]$.

Suppose node i is the target of trust evaluation. The natural approach is to aggregate all its neighbors' opinions. We call this approach a *local voting rule*, in which votes are neighbors' c-values on the target. Furthermore, opinions from nodes with high (estimated) trust values are more credible, so they should carry larger weights. Therefore we define the local voting rule as the following iterated rule

$$s_i(k+1) = f\left(c_{ji}s_j(k) | j \in \mathcal{N}_i\right). \tag{3}$$

Thus the trust evaluation can be considered as a *dynamic process* which evolves with time. Our interest is to study the evolution of the estimated trust vector S, its values at the equilibrium and whether S can correctly estimate the trust vector T at the steady state.

1) Deterministic voting rule: Guided by the above reasoning, we first design a simple *deterministic* rule based on weighted voting. The matrix form of this rule is written as:

$$S(k+1) = D^{-1}CS(k). \tag{4}$$

where D is a diagonal matrix representing the in-degree of each node. We studied its convergence property and investigate the spreading of trust as the system reaches the steady state. To determine the trustworthiness of nodes, we apply a threshold rule when the above voting rule converges. Let $s_i = \lim_{k\to\infty} s_i(k)$. Node i is trusted if $s_i \geq \eta$ and not trusted if $s_i < \eta$.

Our analytic result indicates that $s_i \ll 1$, i.e., trust can not be established, which shows the difficulties of designing algorithms in self-organized distributed networks. In order to overcome the problem, we introduce the notion of *headers*, which are entities that are always trusted by some nodes with trust value 1. Define the average votes provided by headers for node i as b_i. We have the following result: given that the threshold of trustworthiness is η, the number of headers for each node must satisfy

$$B\mathbf{1} \geq \frac{\eta}{1-\eta}(D - V)\mathbf{1}.$$

This result provides a method to design a trust-connected network. For a detailed discussion of the deterministic local voting rule, please refer to [6].

B. Stochastic Voting Rule

As we have discussed, uncertainty of opinions by peers is inevitable for autonomic networks. Thus we introduce randomness into our rule. Define the weighted sum $m_i(k) = \sum_{j\in\mathcal{N}_i} c_{ji}s_j(k)$. At each iteration, assume that the voting result is binary and s_i is decided by the threshold rule, i.e., $s_i(k+1) = 1$ if $m_i(k) \geq \eta$ and $s_i(k+1) = -1$ if $m_i(k) < \eta$. So our stochastic threshold rule is defined as:

$$\Pr\left[s_i(k+1)|m_i(k)\right] = \frac{e^{bs_i(k+1)(m_i(k)-\eta)}}{Z_i(k)}. \tag{5}$$

where $Z_i(k)$ is the normalization factor and $b > 0$ is a constant representing the *degree of certainty*. A small b represents a highly uncertain scenario. Then we have the following results: for the stochastic voting rule defined above, if $b \in (0, \infty)$, we have that the voting rule converges to the steady state with a unique stationary distribution $\pi_S = \frac{e^{bU(S)}}{Z}$.

The stationary distribution π_S can be easily linked to the Ising model and spin glass model in statistical physics. The Ising model describes interaction of magnetic moments or "spins" of particles. In the Ising model, s_i is the orientation of the spin at particle i. $s_i = 1$ or -1 indicates the spin at i is "up" or "down" respectively. The rich literature in statistical physics help us to understand our voting model at the steady state.

We studied the probability of correct estimation ($s_i = t_i$), denoted as $P_{correct}$ at the steady state. One interesting observation is the *phase transition phenomenon* observed when the threshold $\eta = 0$. Phase transitions have been extensively studied by physicists. In [7], the authors theoretically studied phase transitions in spin glass models, and introduced the replica symmetry method to solve them analytically. Based on this method, very good approximations of critical values can be derived. The discovery of phase transition in our voting rule is quite surprising given that the rule itself is very simple. More importantly, the fact that a small change in the parameter might result in a diametrically opposite performance of our voting rule proves the necessity of doing more analyses before applying any distributed algorithms.

In our paper [8], we also studied the impacts of different adversary models and network topologies on our evaluation rule. Please refer to this paper for more details.

IV. Collaboration and Trust

Autonomic networks rely on cooperation of participating nodes for almost all their functions, for instance, to route data between source and destination pairs that are outside each other's communication range. However, because such data forwarding consumes valuable (and scarce) battery power and channel bandwidth without receiving any direct gain, intermediate nodes would prefer not to cooperate. In order to form the necessary infrastructure that makes multi-hop communication achievable, cooperation enforcement mechanisms are developed to cope with such selfish behavior of nodes in autonomic networks.

The conflict between cooperation and cost naturally leads to game-theoretic studies. In our work, the interactions among nodes are modeled as *cooperative games*. In cooperative games, players form *coalitions* to obtain the optimum payoffs. We investigated conditions under which a grand coalition

that includes all players together is formed. Furthermore, we introduce trust as a mechanism to help form a grand coalition in the context of cooperative games.

Since nodes only communicate with their physical neighbors, the interactions among neighbors can be modeled as games on graphs. This area of research has a lot in common with statistical mechanics of complex systems with game theoretic interactions. The Ising model and the more complex spin glass model can be also interpreted as cooperative games. In the Ising model, each particle selects its own spin to maximize its own payoff, defined as the following

$$R_i = \left(\sum_{j \in \mathcal{N}_i} J_{ij} s_i s_j \right) / T.$$

A system with high T means that particles are conservative and not willing to change, while the one with low T has aggressive particles. Therefore, a collection of local decisions reduces the total energy of the interacting particles. This inspires an approach where trust is used as an incentive for cooperation. The value of s_i represents whether node i is willing to cooperate or not ($s_i = 1$ or -1). J_{ij} can be interpreted as the worth of player j to player i, which can be a function of the trust relations between i and j. Then each node decides to cooperate or not based on benefit from cooperation and trust values of its neighbors

In cooperative games, players form coalitions to obtain the optimum payoffs. A coalition S is a subset of N in which all nodes cooperate. The characteristic function $v(S)$ is interpreted as the maximum payoff S can get without the cooperation of the rest of the players $N \setminus S$. Then in the Ising model, the characteristic function for every coalition of players $S \subset N$ is set as

$$v(S) = \sum_{i \in S} R_i = \sum_{i,j \in S} J_{ij} - \sum_{k \notin S, i \in S} J_{ij}.$$

Our object is to find what form or policy for J_{ij} can induce all (or most) nodes to cooperate, i.e., to maximize the coalition.

There are different concepts of stable solutions in cooperative games, such as the core, stable sets, and the nucleolus. In particular, we studied the *core*, in which all players cooperate with their neighbors. However, the core does not always exist. So we studied the conditions under which the core exists in the cooperative game. Furthermore, we studied how negotiation can help to form the grand coalition that includes all players.

The trust management system can be used as an incentive for collaboration. Nodes who refrain from cooperation get lower trust values, and will be eventually penalized because other nodes tend to cooperate only with highly trusted ones. After adding trust into the aforementioned network formation game, the conditions of the existence of core are relaxed. We showed that by introducing a trust mechanism, all nodes are induced to collaborate without any negotiation. For more information on trust and games, the reader is referred to our work [9] and [10].

V. Trust Building and Trust Inference

A. Non-Cooperative Game Theory for Trust Building

Let the graph $G(N, E)$ model a network of interacting users. The set of users is N, and edges exist between the pairs of those who directly interact forming the set E. Each user i has a *type* $t_i \in \{G, B\}$, either Good or Bad, and knows his own type, but does not know the other users' types. The neighborhood of $i \in N$ is denoted $\Gamma(i) = \{j \in N | ij \in E\}$. Time is discrete and the network operates synchronously according to some abstract protocol which the users have the freedom to follow or break. The intuition is that following the protocol means making oneself available for communication (forwarding other users' packets, etc.), whereas breaking the protocol means shutting off all communications. At each time instant each user $i \in N$ decides whether he is going to follow the protocol (C for Cooperate) or not (D for Defect). This decision is denoted $\alpha_i \in \{C, D\}$ for user i, it is observed by all neighbors, and may depend on past observed actions.

After all users choose their actions $\vec{\alpha} \in \{C, D\}^N$, user $i \in N$ receives a payoff R_i that depends on his own and his neighbors' actions, as well as his type: $R_i = R_i(\vec{\alpha}_{\Gamma(i) \cup i}, t_i)$. Each user is trying to maximize his payoff by choosing his action appropriately. This may involve randomizing between the two available actions. Let $\sigma_i(\alpha_i | t_i)$ be the probability that user i chooses action α_i when he is of type t_i. For now, we assume that the users do not collaborate, so these probabilities are independent. The function $\sigma_i(\cdot | t_i)$ is called a *strategy* for i and it can take four values $\sigma_i(C|G), \sigma_i(D|G), \sigma_i(C|B), \sigma_i(D|B)$.

The expected payoff for User 1 when he uses strategy σ_1 and his neighbors $\Gamma(1) = \{2, \ldots, k\}$ use $\sigma_2, \ldots, \sigma_k$ is:

$$R_1(\vec{\sigma}_{\Gamma(1) \cup 1}, t_1) = \sum_{\vec{\alpha}_{\Gamma(1) \cup 1} \in \{C,D\}^k} \sigma_1(\alpha_1 | t_1) \vec{\sigma}_{\Gamma(1)}(\vec{\alpha}_{\Gamma(1)} | \vec{t}_{\Gamma(1)}) R_1(\vec{\alpha}_{\Gamma(1) \cup 1}, t_1)$$

We want to find a Nash equilibrium. A Nash equilibrium is a set of strategies $(\sigma_1^*, \sigma_2^*, \ldots, \sigma_N^*)$ such that no user can unilaterally increase his own payoff by changing his strategy when everybody else's strategy remains the same.

Each user type is private, i.e. known only to himself. There is only a prior probability for the types of his neighbors. The corresponding notion of equilibrium is a Bayes-Nash equilibrium, where each user maximizes his payoff in expectation over not just the strategies of his neighbors, but also over their types.

We now take into account the repetitive nature of the game. Each user remembers all past actions of himself *and his neighbors*. We define as the n-round *history* the collection: $\mathcal{H}^{1 \ldots n} = \{\vec{\alpha}^n, \ldots, \vec{\alpha}^1\}$. So, in general, the strategy function at time n depends on the observed history up to time $n-1$ as well as the type of the player:

$$\sigma_i = \sigma_i(\cdot | \mathcal{H}^{1 \ldots n-1}, t_i).$$

The history is used to update the probability $\Pr(t_j = G | a_j^{1 \ldots n})$ that User j is Good given the actions he has played so far. This

probability in turn influences the actions that are chosen by j's neighbors at the next round, since from the point of view of the neighbors the probability that j will play action a_j^n at round n is $\Pr(t_j = G)\sigma(a_j^n|t_j = G) + \Pr(t_j = B)\sigma(a_j^n|t_j = B)$.

Note that storing the whole history would require unbounded memory, so we may limit the users' ability to recall observations to some fixed number of rounds. One approach would be for the users to remember what happened only in the previous $(n-1)^{\text{st}}$ round. Another approach would be to summarize the history with the help of a *finite length* statistics vector (for example posterior probabilities); then consider strategies σ_i that are functions only of these statistics instead of the entire history $\mathcal{H}^{1\cdots n-1}$.

Apart from finding the Bayes-Nash equilibrium, we also examine the possibility of computing it in a distributed way. It should be noted that there might be more than one equilibria, in which case a distributed algorithm may converge to any one. Not all equilibria give the same payoffs to all players. We would like to reach an equilibrium that gives the highest possible payoff to the Good users.

B. Trust Inference Using Semirings

We view the trust inference problem as a generalized shortest path problem on a weighted directed graph $G(V, E)$ (*trust graph*). The vertices of the graph are the users/entities in the network. A weighted edge from vertex i to vertex j corresponds to the *opinion* that entity i, also referred to as the *issuer*, has about entity j, also referred to as the *target*. The weight function is $w(i, j) : V \times V \longrightarrow S$, where S is the opinion space.

Each opinion consists of two numbers: the *trust* value, and the *confidence* value. The former corresponds to the issuer's estimate of the target's trustworthiness. For example, a high trust value may mean that the target is an ally (in a military setting). The confidence value corresponds to the accuracy of the trust value assignment. A high confidence value means that the issuer has interacted with the target for a long time, and no evidence for malicious behavior has appeared. Opinions with a high confidence value are more useful in making trust decisions.

The core of our approach lies in the two operators that are used to combine opinions: One operator (denoted $\otimes$) combines opinions along a path, i.e. A's opinion for B is combined with B's opinion for C into one indirect opinion that A should have for C, based on B's recommendation. The other operator (denoted $\oplus$) combines opinions across paths, i.e. A's indirect opinion for X through path p_1 is combined with A's indirect opinion for X through path p_2 into one aggregate opinion that reconciles both. Then, these operators can be used in a general framework for solving path problems in graphs, provided they satisfy certain mathematical properties, i.e. form an algebraic structure called a semiring.

The aim is to compute the aggregate opinion from a source i to a destination j along all $i \to j$ paths p.

$$d_{ij} = \bigoplus_p w(p).$$

A *semiring* is an algebraic structure $(S, \oplus, \otimes)$, where S is a set, $\oplus$ is commutative and associative, $\otimes$ is associative and distributes over $\oplus$ $(a, b, c \in S)$:

$$a \oplus b = b \oplus a \qquad (a \oplus b) \oplus c = a \oplus (b \oplus c)$$
$$(a \otimes b) \otimes c = a \otimes (b \otimes c) \quad (a \oplus b) \otimes c = (a \otimes c) \oplus (a \otimes c).$$

A semiring $(S, \oplus, \otimes)$ with a partial order relation $\preceq$ that is monotone with respect to both operators is called an *ordered semiring* $(S, \oplus, \otimes, \preceq)$:

$$a \preceq b \text{ and } a' \preceq b' \Longrightarrow a \oplus a' \preceq b \oplus b' \text{ and } a \otimes a' \preceq b \otimes b'.$$

A semiring is called idempotent when $\forall a \in S : a \oplus a = a$.

Based on intuitive concepts about trust establishment, we can expect the binary operators to have certain properties in addition to those required by the semiring structure. Since an opinion should deteriorate along a path, we require the following for the $\otimes$ operator $(a, b \in S)$:

$$a \otimes b \preceq a, b.$$

Regarding aggregation across paths with the $\oplus$ operator, we generally expect that opinion quality will improve, since we have multiple opinions. If the opinions disagree, the more confident one will weigh heavier. In a fashion similar to the $\otimes$ operator, we require that the $\oplus$ operator satisfies $(a, b \in S)$:

$$a \oplus b \succeq a, b.$$

In our proposed semiring, the opinion space is $S = [0, 1] \times [0, 1]$ Our choice for the $\otimes$ and $\oplus$ operators is:

$$(t_{ik}, c_{ik}) \otimes (t_{kj}, c_{kj}) = (t_{ik}t_{kj}, c_{ik}c_{kj}) \tag{6}$$

$$(t_{ij}^{p_1}, c_{ij}^{p_1}) \oplus (t_{ij}^{p_2}, c_{ij}^{p_2}) = \begin{cases} (t_{ij}^{p_1}, c_{ij}^{p_1}) & \text{if } c_{ij}^{p_1} > c_{ij}^{p_2} \\ (t_{ij}^{p_2}, c_{ij}^{p_2}) & \text{if } c_{ij}^{p_1} < c_{ij}^{p_2} \\ (t_{ij}^{*}, c_{ij}^{p_1}) & \text{if } c_{ij}^{p_1} = c_{ij}^{p_2} \end{cases} \tag{7}$$

where $(t_{ij}^{p_1}, c_{ij}^{p_1})$ is the opinion that i has formed about j along the path p_1, and $t_{ij}^{*} = \max(t_{ij}^{p_1}, t_{ij}^{p_2})$.

This semiring computes the trust distance along the most confident trust path to the destination. This distance is computed along a single path, since the $\oplus$ operator picks exactly one path. Other paths are ignored, so not all available information is being taken into account. One of the advantages is that if the trust value turns out to be high, then a trusted path to the destination has also been discovered. Also, fewer messages are exchanged for information gathering.

Our aim is to evaluate the performance of the proposed semiring (or other semirings) with respect to their resistance to attackers.

The first issue is modeling the attacker's capabilities. We can have node attacks, edge attacks, or both. In a k-node attack, the attacker can choose any k nodes and modify the weights (opinions) on any outgoing edges, including adding new edges. In a k-edge attack, the attacker can change the weights on any k edges. However, no new edges can be added. Obviously, a k-node attack is at least as powerful as a k-edge attack. In general, when the attacker can simultaneously do an x-node attack and a y-edge attack, we call this an x, y attack.

The second issue is to quantify the damage that the attacker causes. An honest user computes trust values for everybody else. and the attacker wants to change the computed trust value for a destination (or as many destinations as possible) as much as possible. So, we calculate the difference between the trust values computed before and after the attack. The damage of the attack can be the sum of the differences, or the maximum of the differences.

Given the trust topology and the weights, what is the maximum damage that an x, y attack can cause? This relates to the robustness of a given trust graph. Now, assume that the Designer of the network can choose the weights on the edges. What is the best choice, i.e. the one that minimizes the damage that the Attacker can cause? If the Designer and Attacker choose simultaneously weights and edges to attack, what will the outcome be?

Crucial in the above discussion is the connection between the weighted topology and the outcome of the trust computation. To analyze this connection and make it more explicit, we treat the outcome of the computation as the steady state of a linear system where the state is the vector of opinions of a node s for every other node, and A is the weighted adjacency matrix of the graph.

$$x = Ax \oplus b$$

where the matrix multiplications and additions are in the semiring arithmetic.

The vector b is used to set the opinions about certain nodes to a fixed value. For example, s's opinion about himself should be the "highest" possible, and should never change. Similar things can hold for the opinion about other nodes. The existence of a set of pre-trusted nodes may help to compute better trust paths (higher confidence).

The idea motivating the use of this approach is that we can study the properties of the trust computation algorithm by studying specific properties of the matrix A. Baccelli, Cohen, Olsder, and Quadrat [11] have made considerable steps towards linking the analysis of the linear system with the properties of A in the case where the operators involved are $\max$ for $\oplus$, and $+$ (regular addition) for $\otimes$. We want to transfer these results to other operators that make more sense for trust.

The algorithm properties that we want to analyze are:

- conditions of convergence (vs. oscillation)
- solution of the system (steady state)
- speed of convergence
- effect of pre-trusted nodes (vector b)

For more information, the reader is referred to our work [12], [13], [14].

VI. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] L. Eschenauer, "On trust establishment in mobile ad-hoc networks," Master's thesis, University of Maryland, College Park, 2002.

[2] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis, "The role of trust management in distributed systems security," *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, pp. 185–210, 1999.

[3] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, California, August 2001.

[4] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," *Lecture Notes in Computer Science*, vol. 2009, p. 46, 2001.

[5] T. Jiang and J. S. Baras, "Ant-based adaptive trust evidence distribution in manet," in *Proceedings of 2nd International Workshop on Mobile Distributed Computing (MDC04)*, Tokyo, Jopan, March 2004, pp. 588–593.

[6] ——, "Autonomous trust establishment," in *Proceedings of 2nd International Network Optimization Conference (INOC)*, Lisbon, Portugal, February 2005.

[7] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. Oxford University Press, 2001.

[8] T. Jiang and J. S. Baras, "Trust evaluation in anarchy: A case study on autonomous networks," in *Proceedings of 2006 INFOCOM*, Barcelona, Spain, April 2006.

[9] J. S. Baras and T. Jiang, *Proceedings of Symposium on Systems, Control and Networks, honoring Professor P. Varaiya*. Birkhauser, June 2005, ch. Cooperation, Trust and Games in Wireless Networks, pp. 183–202.

[10] ——, "Cooperative games, phase transitions on graphs and distributed trust in manet," in *Proceedings of 2004 IEEE Conference on Decision and Control*, Bahamas, December 2004, pp. 93–98.

[11] F. L. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat, *Synchronization and Linearity: An Algebra for Discrete Event Systems*. John Wiley & Sons, 1992.

[12] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *Proc. of the 2004 ACM workshop on Wireless security*. ACM Press, 2004, pp. 1–10.

[13] ——, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 318–328, February 2006.

[14] ——, "Linear iterations on ordered semirings for trust metric computation and attack resiliency evaluation," in *Proc. of 17th International Symposium on Mathematical Theory of Networks and Systems, MTNS 2006*, July 2006.