Ant-based Adaptive Trust Evidence Distribution in MANET*

Tao Jiang
Institute for Systems Research
University of Maryland
College Park, 20742
tjiang@umd.edu

John S. Baras
Institute for Systems Research
University of Maryland
College Park, 20742
baras@isr.umd.edu

Abstract

Due to the lack of infrastructure and vulnerability of wireless links, security in ad hoc networks is considered to be much more difficult than in traditional hierarchical networks. Building the trust relationship between entities is a fundamental problem in ad hoc networks, since the availability of servers, which distribute trust certificates, is not guaranteed. Furthermore, the existence of any trusted server might not be assumed either. Therefore, the commonly used key distribution center (KDC) and certification authority (CA) are not applicable in such highly autonomous environments. In this paper, we propose a scheme for the distribution of trust certificates, which is completely distributed and adaptive to mobility. Our scheme is based on the swarm intelligence paradigm, which has been used for routing both in wired and wireless networks. Our simulations in ns-2 show that it performs very well in ad hoc environments.

1. Introduction

A mobile ad hoc network (MANET) is a collection of autonomous, self-organized, mobile wireless nodes. Due to the absence of infrastructure, vulnerability of wireless links and changes in topology, securing such networks is much more difficult than in traditional hierarchical networks.

Recently there has been a lot of research focusing on the security issues of ad hoc networks and many solutions have been proposed for security services for wireless and decentralized networks. However, almost all the schemes assume the existence of certain trust relationships between communicating nodes. The trust relationship could be, for instance, the authentication of public keys or authorization of certain actions. However, in distributed and autonomous systems, obtaining evidence for these trust relationships is not an easy job.

In traditional networks, trust evidence is generated via a trusted third party (TTP), such as certification authority (CA) for public-key certificates or key distribution center (KDC) for symmetric keys. In these centralized schemes, there exists a vulnerable point of the network, which is responsible for the security of the entire network. Whenever the TTP is out of reach, the whole system breaks down. This is not rare in MANET because of the unique characteristics of ad hoc networks, such as highly dynamic network topology, frequent node joining and leaving, vulnerable wireless links and poor physical protection of devices. Therefore, the assumption of reliable centralized TTP is not applicable and a distributed trust model is needed, which is completely decentralized and hence must be scalable and self-organized.

1.1 Trust Model

The trust model has two components. The first part is the *trust computation* model that evaluates trust level of each entity based on behavioral data or trust evidence. Some trust evaluation models have been proposed in [1], [2], [3], and [4]. The second part is the *trust evidence distribution* system that distributes and obtains the trust evidence. Trust evidence distribution is the foundation of the computation part. It provides the input for the evaluation model. Till now, the evidence management and retrieval problems that exist in distributed ad hoc environments have not been well addressed.

Although choosing the right model to evaluate trust and obtaining the evidence to compute trust go hand in hand, trust evidence distribution is fairly inde-

*Research supported in part by the U.S. Army Research Laboratory under contract No. DAAD190120011

pendent of the specific computation model of trust. At the first stage of our research, in this paper we are emphasizing the distribution and retrieval of trust evidence.

1.2 P2P File-sharing System

A framework that addresses both the computation and distribution problems in decentralized systems is provided in [5]. Its trust management scheme P-Grid [6] is based on scalable replication of tree structures, which is derived from the peer-to-peer file-sharing systems. As distributed and self-organized networks, P2P systems have many common characteristics with ad hoc networks. In [7], the authors use hash-based routing in one of popular P2P networks – Freenet [8] for distribution of trust evidence. Request routing in Freenet avoids flooding and improves with time. Files, or trust evidence documents in this context, are replicated by caching at every node, which causes information to converge to where it is most needed. However, in P2P systems, mobility is not taken into account. In this work, we present a new scheme with the advantages of P2P file-sharing systems and also suitable for mobile environments.

1.3 Swarm Intelligence Paradigm

We propose a new approach: ant-based evidence distribution (ABED). Our ant-based scheme uses the swarm intelligence paradigm [9], which is widely used in dynamic optimization problems, such as traveling salesman problem (TSP) and routing in communication networks. The swarm intelligence paradigm is inspired from artificial ant colonies techniques to solve combinatorial optimization problems [10]. The main principle behind the interaction in a swarm is called *stigmergy* – indirect communication through the environment. An example of *stigmergy* is pheromone laying on the trails followed by ants. Ants are attracted to pheromones and thereby they tend to follow the trails that have high pheromone concentrations.

In ABED by the interaction with each other using information, i.e. "pheromone", deposited in nodes they pass, mobile artificial agents, called "ants", are able to find the optimal path toward their food, i.e. trust evidence in this context. The pheromone regulation process, explained later, enables the exploration of new paths, which makes it particularly suitable for dynamically changing environments, such as MANETs. The interactive cooperation of nodes in ad hoc networks is analogous to the operation of swarms through emergent behavior of ants following a set of simple rules. Thus the swarm intelligence paradigm is well-suited for MANETs. Furthermore, as we argue later, the decision made to find the optimal path in ABED could be easily adjusted by the influence of security metrics, such as the availability of network sources and trustworthiness of entities.

The rest of the paper is organized as follows. In section 2, ABED is explained in detail. Section 3 presents the simulation results. We conclude and suggest future work in section 4.

2. Ant-Based Evidence Distribution (ABED)

2.1 Assumption

Evidence is presented in the form of trust certificates, which are signed by their issuers' private keys. The content of certificates depends on the specific trust computation model. Normally, it could be the public key or any action authorized by the signer. For example, a certificate states that B is able to read database D and is signed by A, which means A authorizes B to read database D. Certificates are uniquely identified by the signer ID, the owner ID and the sequence number.

We assume that the public key of the signer is well known and authenticated, and the corresponding private key cannot be compromised. Due to the limited number of signers in a network, the public key authentication could be done off-line, i.e. before the setup of the network or before entities join in. When a certificate is requested, the signer need not necessarily be online.

2.2 Algorithm Specification

ABED is a reactive evidence distribution scheme. Ants are sent out only when certain certificate is required. Along the path to obtain the certificates, ants modify the information stored in the certificate table (CT) of each node. In the following, we discuss the certificate table and different functions of ants in detail.

Table 1 Certificate Table (CT)

Neighbors

	N_1	N_2	……	N_m
$Cert_1$	P_{11}	P_{12}	……	P_{1m}
$Cert_2$	P_{21}	P_{22}	……	P_{2m}
…	…	…	…	…
$Cert_n$	P_{n1}	P_{n2}	……	P_{nm}

1) Certificate Table (CT) of each node, shown in Table 1: similar with the distance-vector

routing table. CTs differ from distance-vector routing tables in two ways: (i) each entry in CT corresponds to one certificate; (ii) the metric is the probability of choosing each neighbor as the next hop instead of the hop count to destinations. For a node k, certificate table CT_k defines the searching policy currently employed at node k: for each certificate $cert_n$ and for each neighbor node i, the probability value P_{ni} expresses the chance of choosing i as next node when searching cert n, under current network topology. P_{ni} satisfies:

$$\sum_{i\in N_k} P_{ni} = 1, N_k \text{ is the neighbor set of node k}$$

In the rest of the paper, we use P_i instead of P_{ni}, since the probabilities we refer to are of the same certificate, except when explicitly indicated.

2) There are two kinds of forward ants sent out for a particular certificate:

- Unicast ants are sent out to the neighbor with the highest probability in the certificate table.
- Broadcast ants are sent out when there is no preference to the neighbors, i.e. there is no entry in CT for this certificate. This happens either when no path to the certificate has been explored or the information the node has is outdated. The criteria for deciding whether the information is outdated or not is related to the density of pheromone, which we will define in subsection 2.4.

3) Once forward ants find the required certificate, a backward ant is generated. The backward ant retraces the path of the forward ant back to the source. It takes the certificate in its packet. The backward ant then induces certificate table modifications at each intermediate node according to some learning rule – reinforcement. The reinforcement update rule is the core of ABED, which leads to the optimum solution for finding the trust evidence. We will discuss this update rule in the next subsection.

As in Freenet, the certificate is cached in every node on the path of backward ants. As a result, trust certificates are distributed in places where they are most needed. Therefore after a period of adaptation, the request overhead will be drastically reduced, since probability of obtaining certificates from neighbors is high. The replicated evidence assures the availability of certificates, even when the signer is out of reach.

2.3 Reinforcement Rule

Backward ants update the certificate table entries when they visit each node. We begin with a simple non-linear reinforcement-learning rule proposed by D. Subramanian et al in [11].

$$P_i(n) = \frac{P_i(n-1)+\Delta p}{1+\Delta p}$$
$$P_j(n) = \frac{P_j(n-1)}{1+\Delta p} \qquad j \in N_k,\ j \neq i \tag{1}$$

where i is the neighbor the backward ant came from, $\Delta p = k/f(c)$, $k > 0$ is a constant and $f(c)$ is a non-decreasing function of cost c.

Cost c could be any parameter that reveals the information of evidence or the scenario of current network. For instance, c could be the metric of hop counts from current node to the node the certificate resides, the delay in obtaining the certificate, available bandwidth of the link, traffic density experienced during the transmission or the energy of each node along the way. An interesting point is that we could also put the security metrics into this model. An example is to make use of the cumulative trust value of the path as the cost. The higher the trust value, the lower the cost. Suppose the path backward ant traversed is $\{N_1, N_2, \ldots, N_L\}$, L > 0 is the total hop count, and the trust value with each node is $\{Q_1, Q_2, \ldots, Q_L\}$. Then we could define

$$f(c) = \frac{a \cdot L}{\sum_{i=1}^{L} Q_i}, \text{ a is a constant.} \tag{2}$$

In order to explore all the information carried by the ants, we investigated another more complicated reinforcement rule.

$$P_i(t) = \frac{[\tau_i(t)]^\alpha [\eta_i]^\beta}{\sum_{j\in N} [\tau_j(t)]^\alpha [\eta_j]^\beta} \tag{3}$$

where η_i is the goodness value of the link between current node k and its neighbor node j, such as the inverse of bandwidth usage of link k→i. τ_i is the pheromone deposit, which is defined as follows, if at time $t+\Delta t$, current node k receives a backward ant from node i, and the last processing time is t,

$$\tau_i(t+\Delta t) = f(\tau_i(t), \Delta t) + \Delta p$$
$$\tau_j(t+\Delta t) = f(\tau_j(t), \Delta t) \qquad j \in N,\ j \neq i \tag{4}$$

Δp is the same as in equation 1. $f(\tau_i(t), \Delta t)$ is the pheromone evaporation function, defined as

$$f(\tau_i(t), \Delta t) = \frac{\tau_i(t)}{e^{\Delta t/k}} \qquad (5)$$

α and β are constants varied in different network environments and determined usually by simulation.

Another function of pheromone is to decide when to send out broadcast forward ants. When node k receives a request at time t, it first searches if there is an entry for the desired certificate. If no such entry exists, it simply sends out broadcast ants. Otherwise, it finds the one with the highest probability. Suppose the time last pheromone update corresponding to this next hope is t_0. Then the pheromone deposit at time t would be

$$\tau(t) = \frac{\tau(t_0)}{e^{(t-t_0)/k}} \; (6)$$

Then if $\tau > \tau_0$, node k sends a unicast ant to this neighbor, otherwise it finds the next highest probability and repeats until send out a unicast ant or it sends out broadcast ants when there is no remaining neighbor with probability $P > 1/|N|$. τ_0 is a threshold for the freshness of the pheromone, $|N|$ is the number of neighbors.

When an update takes place, it not only reinforces the corresponding entry, but also possibly affects other certificate entries, which are related with the current one. The correlation between different certificates is dependent on the content of trust evidence. For instance, if two certificates are signed by the same private key, it is reasonable to assume that with high probability these two certificates are stored in the same area or at least not far away. Then instead of only reinforcing one certificate table entry; the other one gets updated too. We use the same reinforcement rule, with $\Delta p' = \gamma \cdot \Delta p$, for the other entry. $0 \le \gamma \le 1$ represents the correlation of two certificates.

2.4 Mobility

Mobility is the essential characteristic of ad hoc networks. ABED takes into account mobility changes with respect to link breaks and the influence on the pheromone trail, which are discussed below.

1) Link break and negative reinforcement

Link breaks between two nodes occur when one node is unable to relay a packet to the other. Link can break when two nodes move far apart, the devices are broken or there is an obstacle between them. It is crucial for an algorithm to handle link breaks in ad hoc networks. As for ABED, in equation 2, η_i represents the goodness of link between two nodes, once link break is detected, η_i is set to 0 or a small value near 0, such as 0.1, as a means of assigning negative reinforcement to the certificate.

2) Pheromone

The concentration of pheromone deposit is to direct the ants to find an optimal solution while the evaporation allows the system to forget the old information, search new paths, and also avoid convergence to sub-optimal solutions.

As we stated before, the pheromone evaporation is a function of elapsed time. Intuitively, pheromone evaporation is also a function of mobility, i.e. higher mobility means faster evaporation. Then how can the nodes gain the knowledge of mobility in a distributed manner? Here we make use of the parametric statistical model in AntNet [10] to address the mobility problem.

For each certificates, estimated mean μ and variance σ^2 represent the expected time to obtain the certificate and its stability. We compute the statistics in a time window — only those backward ants received in the time window are valid. The time it takes from sending the request to receiving the backward ant can be derived from information in backward ant packets. This elapsed time is used to compute the mean and variance. Then the evaporation function is modified to

$$f(\tau_i(t), \Delta t) = \frac{\tau_i(t)}{e^{\Delta t \cdot \sigma/k}} \qquad (7)$$

Large variance means high mobility, so the information in the past is less useful.

3. Simulation

We simulated ABED using the discrete time network simulator ns-2. We compared ABED with the Freenet based scheme proposed in [7].

The simulation parameters we used are as follows. A total of 300 nodes are randomly placed in a field of a 3000×3000 meters. Each node transmits packets using a 2Mbps wireless channel with a transmission

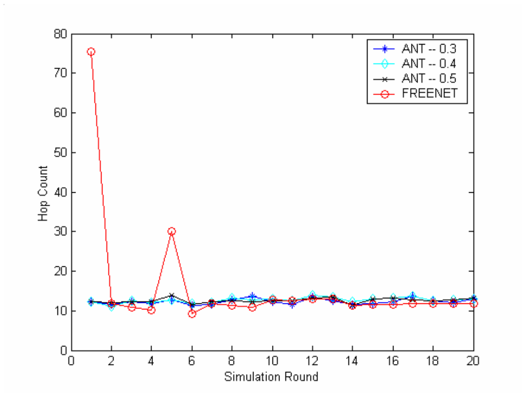

Figure 1 Comparison of hop count between Freenet and ABED

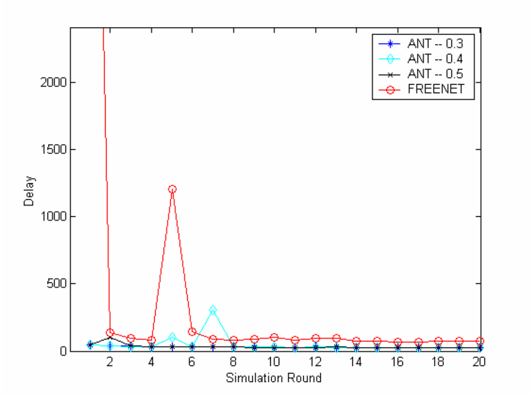

Figure 2 Comparison of delay

range of 250m. Therefore the diameter of this network, which is defined as the maximum number hops between two nodes, is approximately 12 hops.

The simulation proceeds in rounds. Each round, the updated certificates are inserted in the network, and 20 randomly chosen nodes request one of the 4 existing certificates. We ran each simulation 16 times, with the combination of four different node placements and four different request setups. The numbers presented are averaged over all the 16 runs.

The metrics that used in the comparison are:

Hop count: the number of hops that forward and backward ants traversed in order to carry the certificate back to the requester.

Delay: the time elapsed from sending out the forward ant to receipt of the first backward ant.

Success rate: the percentage of requests for which the requester successfully obtains the certificate. In simulation, it is the number of certificates obtained over the total number of requests each round.

Figure 1, 2 and 3 are the comparisons of hop counts, delay and success rate between the Freenet-based scheme and ABED. The reinforcement rule used is equation 1, with f(c) = the number of hops to reach the node caching the certificate. Parameter k is set to be 0.3, 0.4, and 0.5 separately. As shown in Figure 1 and Figure 2, both schemes converge to the same value. Freenet has a “slow start” period, while ABED finds the best solution very fast. Fast convergence is highly desired in mobile scenarios. Normally, by fully searching the network situation, our scheme gains a better view of the whole network. For instance, as we notice in Figures 1 and 2, Freenet performs very badly at certain round, because it has no pre-knowledge of newly requested certificates.

In terms of success rate, ABED outperforms the Freenet-based scheme as shown in Figure 3. It also shows that in ABED, the optimal value for parameter k is 0.5. The higher k is, the greater the effect that reinforcement has on the probability. But k should not be too high; otherwise the probability cannot converge.

We also observe the network load during each round, as shown in Figure 4. Except at the beginning, when ABED has higher overhead than the Freenet-based scheme, in the remaining rounds they are almost

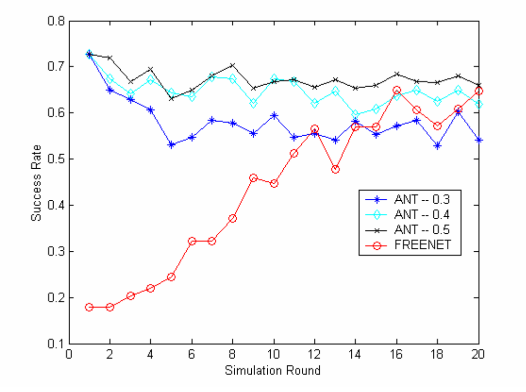

Figure 3 Comparison of success rate

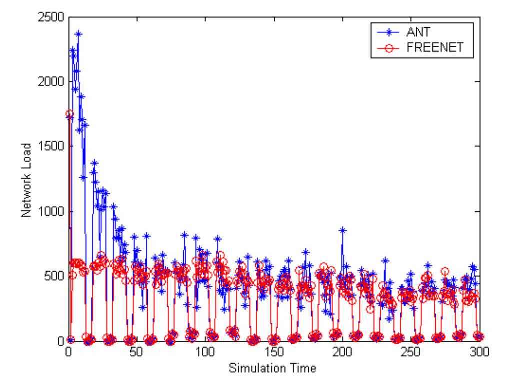

Figure 4 Comparison of Network Load

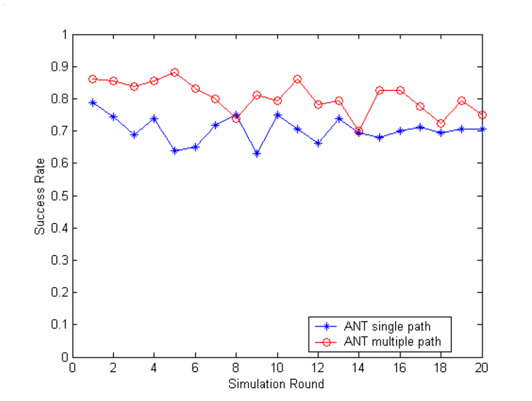

Figure 5 Comparison of Success Rate between single path and multiple path

the same. The high overhead at the beginning is the trade-off for the fast convergence achieved by using broadcast.

Notice that in Figure 3 the success rate of ABED decreases by a small amount with time. The reason is the collisions in wireless channel, which introduce packet loss during transmission. In the first two rounds, the broadcast explores multiple paths from the requestor to the certificates, so transmission failure on one path will not affect the final result. As time progresses, the number of unicast ants increases. A single collision leads to the failure of a request. Multiple paths are inherent in the swarm intelligence paradigm, so we send unicast ants along multiple paths. The results are depicted in Figure 5. The success rate improves as we expected, but we found that the delay is much larger than before. One reason is that the traffic load increases because more ants are sent out. Another reason is the back-off scheme in MAC layer, since more than one unicast ants are sent out simultaneously by the same node, all ants have to wait for a random time period in the buffer and are sent out later. For applications with loose requirement on delay, the scheme with multiple paths is desirable in terms of both performance and security concern.

4. Conclusions and Future Work

In this paper, we present an approach to distribute trust evidence in ad hoc networks based on the swarm intelligence paradigm and ideas of P2P file-sharing systems. Recently, *Gnutant* in the Anthill project [14] also provided a swarm intelligence based p2p file-sharing system, but it is mainly dependent on hash-key routing. As we argued before, it does not address the special problems in MANET, such as mobility and the vulnerability of both links and nodes. The advantages of ABED are its adaptability to network changes and tolerance for the faults in networks. Moreover, because of flexibility of metrics in the reinforcement rule, it is easy to embed security and trust content in the certificate distribution process in ABED. The objective of ABED is to properly and efficiently distribute evidence in the wireless ad hoc network, which facilitates the evidence requests and reduces the communication cost. Performance results for ABED in autonomous, decentralized and mobile systems is promising.

As future work, we plan to analyze the parameters under different network settings and further explore the influence of mobility and malicious nodes. The convergence property of swarm intelligence paradigm needs to be considered in our scheme. Most importantly, we hope to construct a good trust computation model, and combine computation and distribution models to build a self-organized, adaptive, fault-tolerant and scalable trust model in MANET.

References

[1] M. K. Reiter and S. G. Stubblebine. Toward acceptable metrics of authentication. In *Proc. IEEE Conference on Security and Privacy*, Oakland, CA, 1997.

[2] R. Yahalom, B. Klein, and T. Beth. Trust relationships in secure systems—A distributed authentication perspective. In *Proc. 1993 IEEE Symposium on Research in Security and Privacy*, pages 150-164, May 1993.

[3] T. Beth, M. Borcherding, and B. Klein. Valuation of trust in open networks. In *Proc. ESORICS 94*. Brighton, UK.

[4] R. Kohlas and U. Maurer. Confidence Valuation in a Public-key Infrastructure Based on Uncertain Evidence. In *Proc. Public Key Cryptography* 2000.

[5] K. Aberer and Z. Despotovic. Managing Trust in a Peer-2-Peer Information System. In *Proc. CIKM* 2001.

[6] K. Aberer. P-Grid: A self-organizing access structure for P2P information systems. In *Proc. 6^{th} CoopIS* 2001.

[7] L. Eschenauer. *On Trust Establishment in Mobile Ad-Hoc Networks*, Master thesis, University of Maryland, College Park, 2002

[8] I. Clarke, O. Sandberg, B. Wiley, and T.W. Hong. Freenet: A distributed Anonymous Information Storage and Retrieval System. *In Proc. ICSI Workshop on Design Issues in Anonymity and Unobservability*, Berkeley, CA, 2000.

[9] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligece – From Natural to Artificial Systems.* Oxford University Press, New York, 1999.

[10] G. D. Caro and M. Dorigo. AntNet: Distributed Stigmergetic Control for Communications Networks, *Journal of Artificial Intelligence Research*, volume 9, 1998

[11] D. Subramanian, P. Druschel, J. Chen. Ants and Reinforcement Learning: A case Study in Routing in Dynamic Networks. In *Proc. MILCOM*, Atlantic City 1997

[14] O. Babaoglu, H. Meling and A. Montresor. Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems. In *Proc. 22^{nd} ICDCS*, Vienna, Austria, 2002.