Journal of Computations & Modelling, vol. x, no. x, 2022, x-x
ISSN: 1792-7625 (print), 1792-8850 (online)

Trust Management in Growing Decentralized Networks

Noriko Matsumoto[1] Masaya Hiraide[2], and Norihiko Yoshida[3]

Abstract

In a volunteer-based self-organizing network, such as a peer-to-peer network and a wireless ad-hoc network, trustworthiness of each participant node is crucial to assure information security and resilience of the whole network. We call this kind of network a "decentralized" network.

In a decentralized network, mutual evaluation of trust among nodes are considered useful to manage trustworthiness in many studies. As far as we know, all the preceding related studies discussed trust management upon stable mature networks. However, most decentralized networks in reality begin with initial states comprised of a small number of nodes, and grow gradually with new-comer nodes joining in.

This paper proposes a new trust management method for a growing decentralized network. The network grows from the initial stage where we assume all the initial nodes are trustworthy enough. This paper exhibits that our method extending GossipTrust brings better performance and better quality of trust evaluation than the original GossipTrust.

Mathematics Subject Classification : xxxxx
Keywords: Trust, Reputation, P2P networks, Ad-hoc networks, Network growth

[1] Rissho University, Japan
[2] Hitachi Systems, Ltd.
[3] Rissho University, Japan

Article Info: *Received* : January 10, 2011. *Published online* : July 17, 2011

1 Introduction

In a volunteer-based self-organizing network, such as a peer-to-peer network and a wireless ad-hoc network, trustworthiness of each participant node is crucial to assure information security and resilience of the whole network. We call this kind of network a "decentralized" network in this paper as we cannot assume any centralized global management nor control.

In a decentralized network, nodes are not always trustworthy. Some node may be untrustworthy on purpose, by accident, or by breakdown, doing nothing good or doing something bad. Such node may not provide any contents but only consumes other contents, or may spread malicious contents. It may not transfer packets properly but only discards them, or may transfer packets wrongly, or even worse, may cause flooding of waste contents and waste packets. Such node should be forced to become trustworthy, or otherwise, should be eliminated from the network.

Mutual evaluation of "reputation" among nodes are considered useful to manage trustworthiness in many studies. Examples are: Wang, et al. [1], Zuo, et al. [2], Liu, et al. [3], and Ahmed et al. [4] for trust modeling; Kamvar, et al. [5], Xiong, et al. [6], Zhou, et al. [7], and Selçuk, et al. [8] for peer-to-peer network implementation; Martinovic, et al. [9] and Velloso, et al. [10] for wireless ad-hoc network implementation. Among them, EigenTrust [5] is one of the earliest and the most famous, and it has many extensions, by Nishikawa, et al. [11], Lu, et al. [12], Alhussain, et al. [13], and Afanador, et al. [14] for instance.

As far as we know, all the preceding related studies discussed trust management upon stable mature networks. However, most decentralized networks in reality begin with initial states comprised of a small number of nodes, and grow gradually with new-comer nodes joining in. We may assume that the initial nodes are considered to be trustworthy. Kamvar, et al. on EigenTrust [5] also stated "early users of a P2P network are likely to have less motivation to destroy the network they built." However, none of related studies discussed upon growing networks, nor utilized this assumption.

This short paper proposes a method to utilize growth of decentralized networks in trust management, and shows some improvement over a preceding study. The rest of the paper is organized as follows: Section 2 describes our method as an extension to GossipTrust. Section 3 presents simulation-based experiments to evaluate our method, and their results together with some consideration. Section 4 contains concluding remarks and future work.

2 Proposed Method

2.1 Overview

The essence of our proposed method is as follows:

- A network begins its life with a small number of nodes. All the initial nodes are assumed to be trustworthy.

- The network grows in size with new nodes coming in. A new comer is evaluated by existing trustworthy nodes already in the network.

- Only a node evaluated to be trustworthy enough can join. Otherwise, a new comer is rejected in the manner that all the existing nodes refuse to communicate with it.

In this paper, we use GossipTrust [7] as a design platform. However, our method can be applied to other platforms as well. We choose GossipTrust because it is one of the most basic implementations for trust management on unstructured peer-to-peer networks. EigenTrust [5] and its extensions are on structured peer-to-peer networks using distributed hash tables, and are much more difficult to construct in reality.

2.2 GossipTrust

The GossipTrust procedure is summarized as follows. Its details are found in the reference [7]. Actually, the aggregation cycle scheme is common to EigenTrust and GossipTrust, whereas the gossip step scheme is specific to GossipTrust.

Each node evaluates and collects trust scores of its neighbors, which are called local trust scores. Then all the local scores are aggregated among all the nodes to evaluate the network-wide global trust scores.

(1) Aggregation cycle

Consider a trust matrix $R = (r_{ij})$ $(1 \leq i, j \leq n)$ where n is the number of nodes and r_{ij} $(0 \leq r_{ij} \leq 1)$ is the local trust score issued by node i for node j. For global trust aggregation, each node must normalize all local scores issued by itself as $s_{ij} = r_{ij}/\Sigma_j r_{ij}$. Then we have a normalized trust matrix $S = (s_{ij})$.

Let $v_j(t)$ be the global trust score of node j at aggregation cycle t, where t is the cycle number. The global scores of all nodes form a normalized trust vector $V(t) = \{v_j(t)\}^T$ where $\Sigma_j v_j(t) = 1$.

For all iterative cycles t, we generate successive trust vectors performing matrix-vector computation $V(t) = S^T \times V(t-1)$. This iterative computation continues until the average relative error between $V(t-1)$ and $V(t)$ is lower than δ for a given aggregation error threshold δ. The global trust vector converges to the eigenvector of trust matrix S.

(2) Gossip step

To compute successive trust vectors, GossipTrust uses a gossip-based protocol to perform the matrix-vector computation $V(t) = S^T \times V(t-1)$.

Each aggregation cycle consists of several gossip steps. In a gossip step, each node receives trust vectors from others, selectively integrates the vectors with its current trust vector, and then sends the updated one to a random node in the network. This gossiping process continues until the gossiped scores converge as determined by a gossiping error threshold ϵ.

Each node i is associated with a set of gossip pairs $\{x_{ij}(k), w_{ij}(k)\}$ for node j at each gossip step k. At time t, we have the initial weighted score $x_{ij}(0) = s_{ij} \times v_i(t)$ as the local score s_{ij} weighted by the global score $v_i(t)$ of node i. The $w_{ij}(k)$ is called a consensus factor of node i at step k, and its initial value is 1 if $i = j$, and 0 otherwise.

During each gossip step, every node i executes two computing threads. One thread sends the halved gossip pair $\{x_{ij}(k)/2, w_{ij}(k)/2\}$ to itself and to a randomly selected node in the network. The other thread receives the halved pairs from other nodes and computes the updated x_{ij} and w_{ij} as $x_{ij}(k) = \Sigma_r x_{rj}(k-1)/2$ and $w_i(k) = \Sigma_r w_{rj}(k-1)/2$ respectively, where r refers the index of a remote node which has sent the halved gossip pair.

This procedure continues until the gossip value $u_{ij}(k) = x_{ij}(k)/w_{ij}(k)$ agree on all nodes i. The global score v_j is thus generated as $v_j(t) = u_{ij}(k)$ on all n nodes at the final step k.

2.3 Our extension

In the case of a growing network, the GossipTrust procedure is applied every time a new node comes in. At the initial stage of the network, all the nodes, two at least, are assumed to be fully trustworthy.

Suppose that a network $N(m-1)$, where $m-1$ is the number of nodes, evaluates a new node, and is going to be $N(m)$ if the new node is accepted. In $N(m-1)$, all the global trust scores $v_j(m-1)$ of node j for $j = 1, 2, ..., m-1$ are already computed. The GossipTrust procedure for $N(m)$ inherits these $v_j(m-1)$ of node j $(1 \leq j \leq m-1)$ from $N(m-1)$, and apply them to a corresponding part of the trust matrix $R(m) = (r_{ij})(m)$ $(1 \leq i, j \leq m)$, instead of computing all the $(r_{ij})(m)$ by itself. This reflects our consideration that the trust score of a node evaluates trustworthiness of the node, and the trustworthiness does not change between $N(m-1)$ and $N(m)$.

The inherited v_j has been normalized so that $\Sigma_j v_j = 1$, whereas r_{im} for the new node m is between 0 and 1. Therefore, v_j is denormalized to v'_j so that the maximum of v'_j is 1. These r_{im} and $v'_j(m-1)$ are assigned to $r_{ij}(m)$ $(1 \leq i, j \leq m)$ as follows:

$$r_{ij}(m) \leftarrow \begin{cases} 0 & \text{if } i = j, \\ r_{im} & \text{if } j = m, \\ v'_j(m-1) & \text{for all } i \text{ if } j \leq m-1. \end{cases}$$

Below is an example of the trust matrix $R(m) = (r_{ij})(m)$ in the case of $m = 4$, i. e. the 4th node is a new comer.

$$R(4) = \begin{pmatrix} 0 & v'_2(3) & v'_3(3) & r_{14} \\ v'_1(3) & 0 & v'_3(3) & r_{24} \\ v'_1(3) & v'_2(3) & 0 & r_{34} \\ v'_1(3) & v'_2(3) & v'_3(3) & 0 \end{pmatrix}$$

Table 1: Experiment setup.

Parameter	Value
The initial number of nodes	2
The (final) number of nodes	10, 20, 50, 100
Goodness value of a node	20%: 0.1–0.3, 80%: 0.7–0.9
Trust threshold	0.0, 0.5
Aggregation cycle convergence threshold δ	2×10^{-3}
Gossip step convergence threshold ϵ	2×10^{-4}

3 Experiments

3.1 Setup

We conducted some simulation-based experiments to evaluate our proposed method. We implemented our method together with the original GossipTrust for comparison. Table 1 shows experiment parameters.

- The (final) number of nodes: For the original GossipTrust, where the number of nodes does not increase, 10, 20, 50, or 100 nodes are included in a network from the first.

- Goodness: This is the true value of good or bad of each node. A node having the goodness 0.8 shows good responses to 8 out of 10 incoming packets, and bad responses to 2 out of 10 packets. 80% of the nodes are assigned 0.7–0.9 randomly, and 20% are assigned 0.1–0.3. The closer an evaluated trust score is to the goodness value, the better the trust evaluation is.

- Trust threshold: If this is set to 0.5, only a new node having a global trust score (normalized so that its maximum is 1.0) above 0.5 can join the network. If this is set to 0.0, a new node can always join regardless of its trust score. The latter is for reference.

- Convergence threshold δ and ϵ: The GossipTrust procedure as well as our method is composed of two-fold convergence loops, i. e. the outer aggregation cycle and the inner gossip step. Each has a threshold to

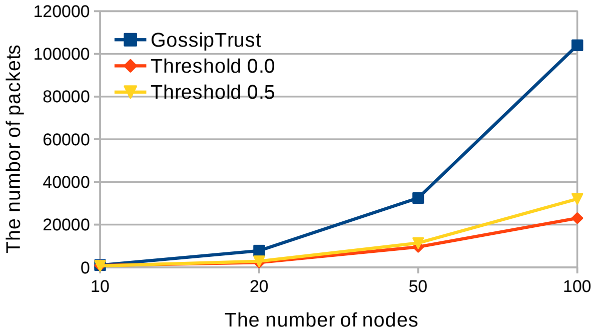

Figure 1: The number of packets

determine whether the convergence error is small enough and the loop is to be terminated or not yet.

3.2 Results and Consideration

We compare four measures among the original GossipTrust, our method with the trust threshold 0.0, and our method with the trust threshold 0.5. Each result presented in the chart is an average of five trials.

(1) The number of packets

Figure 1 shows that our method reduces the number of total packets for evaluating global trust scores of all the nodes. GossipTrust evaluates the scores once. Our method evaluates the scores every time a new node comes, however the global scores are inherited from the previous network. This inheritance reduces the number of convergence cycles, resulting in acceleration of the evaluation process and packet traffic reduction. When the trust threshold is 0.0, the number of nodes in the network reaches to the final one faster, therefore the number of packets is even smaller.

The experiment is done with model-based simulation, therefore we cannot measure or estimate the acceleration in actual time.

(2) Trust evaluation errors (average and standard deviation)

Figure 2 and Figure 3 show that our method reduces both the average and the standard deviation of trust evaluation errors at each node compared to

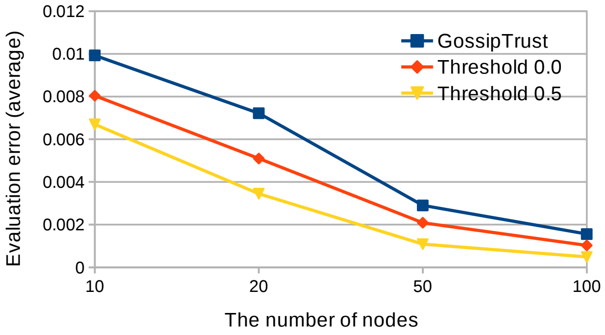

Figure 2: Trust evaluation errors (averages)

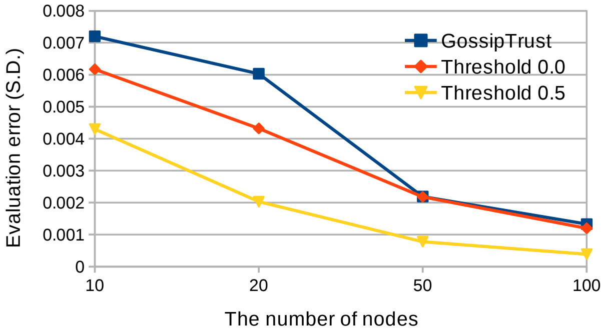

Figure 3: Trust evaluation errors (standard deviations)

the goodness of the node, respectively. These imply our method brings more accurate trust evaluation. This must be because our method evaluates the global trust scores of all the nodes in an incremental manner even if the trust threshold is 0.0. If the trust threshold is 0.5, the network contains trustworthy nodes only, therefore the evaluation is even better.

(3) Node goodness (average)

Figure 4 shows that our method with the trust threshold > 0.5 increases average goodness of all the nodes in the network as a whole. If the trust threshold is 0.0, all new node can join, and the resulting network is identical to the network for GossipTrust, therefore the average goodness is the same as of GossipTrust.

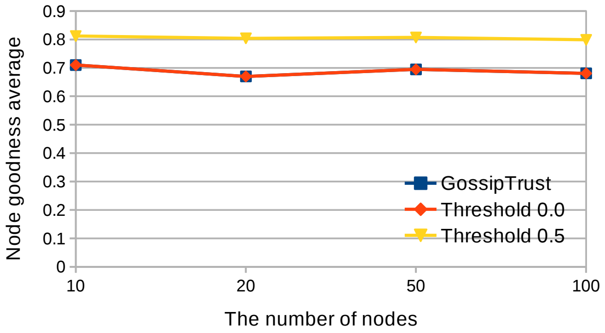

Figure 4: Node goodness averages

4 Concluding Remarks

This paper proposes a new trust management method for a growing decentralized network. The network grows from the initial stage with only a few nodes involved, where we assume all the initial nodes are trustworthy enough. We suppose this assumption and the incremental nature of our method added to GossipTrust bring better performance and better quality of trust evaluation than the original GossipTrust.

We have obtained some preliminary, yet promising results, however we are still at the starting point of this research, and there are still many issues which must be dealt with. One of the most important issue is how to handle a node which changes its goodness after joining, whether permanent or intermittent. A good node may change to a bad one after joining a network. As of now, our idea is to apply history logging or profiling of node communication patterns to detect such changes. Security for the trust evaluation procedure itself is closely related to this issue. There have been already some studies on security and defense issues [15], and their results must be very helpful.

Other further work includes implementation of our method on another platform. EigenTrust and GossipTrust are classic ones, and recently there have been some new interesting trends emerging for trust management. For instance, so-called “bio-inspired” approaches for optimization of complex systems have been derived from social behaviors of insects and animals among which ants and bees are typical examples, and are applied to many issues successfully. Some study tried to apply this approach to trust management, and obtained promising results [16]. Another trend is integration of trust management with distributed ledger technologies such as Blockchain [17, 18]. This trend is collecting attention, and we may expect some new noticeable breakthrough. It is worth and interesting applying our incremental method to these.

References

[1] Y. Wang, Y. Bai, J. Hou, and Y. Tan, "A Malicious Users Detecting Model Based On Feedback Correlations," International Journal of Computer Networks and Communications, Vol.5, No.1, 2013, pp.53–68.

[2] C. Zuo, J. Zhou, and H. Feng, "A Novel Multi-Level Trust Model to Improve the Security of P2P Networks," 3rd IEEE International Conference on Computer Science and Information Technology, Vol.5, 2010, pp.100–104.

[3] K. Liu, G. Jin, D. Rao, J. He, and X. Jiang, "A Participation-Based Trust Model for Mobile P2P Network," Journal of Networks, Vol.9, No.7, 2014, pp.1738–1746.

[4] A. Ahmed, K. A. Bakar, M. I. Channa, K. Haseeb, and A. W. Khan, "A Survey on Trust Based Detection and Isolation of Malicious Nodes in Ad-Hoc and Sensor networks," Frontiers of Computer Science, Vol.9, No.2, 2015, pp.280–296.

[5] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," Proc. 12th International Conference on World Wide Web, 2003, pp.640–651.

[6] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities," IEEE Transactions on Knowledge and Data Engineering, Vol.16, No.7, 2004, pp.843–857.

[7] R. Zhou, K. Hwang, and M. Cai, "GossipTrust for Fast Reputation Aggregation in Peer-to-Peer Networks," IEEE Transactions on Knowledge and Data Engineering, Vol.20, No.9, 2008, pp.1282–1295.

[8] A. A. Selçuk, E. Uzun, and M. R. Pariente, “A Reputation-Based Trust Management System for P2P Networks,” International Journal of Network Security, Vol.6, No.3, 2008, pp.227–237.

[9] I. Martinovic, C. Leng, F. A. Zdarsky, A. Mauthe, R. Steinmetz, and J. B. Schmitt, “Self-Protection in P2P Networks: Choosing the Right Neighbourhood,” Proc. International Workshop on Self-Organizing Systems, 2006, pp.23–33.

[10] P. B. Velloso, R. P. Laufer, D. de O. Cunha, O. C. M. B. Duarte, and G. Pujolle, “Trust Management in Mobile Ad Hoc Networks Using a Scalable Maturity-Based Model,” IEEE Transactions on Network and Service Management, Vol.7, No.3, 2010, pp.172–185.

[11] T. Nishikawa and S. Fujita, “A Reputation Management Scheme for Peer-to-Peer Networks based on the EigenTrust Trust Management Algorithm,” Journal of Information Processing, Vol.20, No.3, 2012, pp.578–584.

[12] K. Lu, J. Wang, L. Xie, Q. Zhen, and M. Li, “An EigenTrust-based Hybrid Trust Model in P2P File Sharing Networks,” Procedia Computer Science, No.94, 2016, pp.366–371.

[13] A. Alhussain and H. Kurdi, “EERP: An Enhanced EigenTrust Algorithm for Reputation Management in Peer-to-Peer Networks,” Procedia Computer Science, Vol.141, 2018, pp.490–495

[14] J. Afanador, N. Oren, M. Baptista, and M. Araujo, “From Eigentrust to a Trust-Measuring Algorithm in the Max-Plus Algebra,” Frontiers in Artificial Intelligence and Applications, No.325, 2020, pp.3–10.

[15] Y. Sun, Z. Han, and K. J. R. Liu, “Defense of Trust Management Vulnerabilities in Distributed Networks,” IEEE Communications Magazine, Vol.46, No.2, 2008, pp.112–119.

[16] N. Al-Otaiby, A. Alhindi, and H. Kurdi, “AntTrust: An Ant-Inspired Trust Management System for Peer-to-Peer Networks,” Sensors, Vol.22, No.533, 2022, 15 pages.

[17] M. Masmoudi, C. A. Zayani, I. Amous, and F. Sedes, "A New Blockchain-Based Trust Management Model," Procedia Computer Science, No.192, 2021, pp.1081–1091.

[18] W. Li, J. Wu, J. Cao, N. Chen, Q. Zhang, and R. Buyya, "Blockchain-Based Trust Management in Cloud Computing Systems: a Taxonomy, Review and Future Directions," Journal of Cloud Computing: Advances, Systems and Applications, Vol.10, No.35, 2021, 34 pages