Simulating Trust Overlay in P2P Networks

Yan Zhang[1], Wei Wang[2], and Shunying Lü[1]

[1] Faculty Mathematics and Computer Science, Hubei University, 430062, Wuhan, China
[2] Department of computer Science and Technology, Tongji University, 201804, Shanghai, China
willtongji@gmail.com

Abstract. Though research on the overlay network has progressed at a steady pace, its promise has yet to be realized. One major difficulty is that, by its very nature, the overlay networks is a large, uncensored system to which anyone may contribute. This raises the question of how much dependability to give each information source. Traditional overlay network simulators provide accurate low-level models of the network hardware and protocols, but none of them deal with the issue of trust and reliability in the large scale overlay networks. We tackle this problem by employing a trust overlay simulator, which offer a viable solution to simulate trustworthy behavior in overlay networks. With this simulator, we can examine varies kinds of trust and reputation mechanisms in a large scale overlay environment.

Keywords: overlay networks, trust, simulator, security, peer-to-peer.

1 Introduction

Over the past two decades, researchers have proposed lots of solutions to extend the Internet's functionality, and to improve its resilience and security [1]. After sustained efforts to add new functions such as mobility and multicast to IP, researchers have recently turned their attention to developing new network architectures [2, 3] and using overlays to address the Internet's limitations.

Overlay networks tend to be large, heterogeneous systems with complex interactions between the physical machines, underlying network, application and user. Hence, developing and testing an overlay-oriented algorithm or protocol in a realistic environment is often not feasible. So, it is possible to use a simulation of an overlay network to evaluate the proposed applications and protocols in controlled environment.

However, though research on these works has progressed at a steady pace, its promise has yet to be realized. One major difficulty is that, by its very nature, the overlay networks is a large, uncensored system to which anyone may contribute, for example using a P2P infrastructure for information sharing. This raises the question of how much dependability to give each information source. Traditional network simulators only provide low-level models of the network hardware and protocols and do not consider trustworthy behavior of the overlays. So, simulating trustworthy behavior in overlay networks is a challenge task in recent simulation research. The aim of this paper is to motivate the need for overlay network simulators for developing and testing trust and reputation-based schemas.

Y. Shi et al. (Eds.): ICCS 2007, Part I, LNCS 4487, pp. 632–639, 2007.

In this paper, we propose a novel *trust overlay simulator* (TOSim). TOSim is developed based on PeerSim which is wildly used in simulating overlay networks [8, 9]. We extend this simulator by introducing varies kinds of components into it, which is important when simulating trust and reputation mechanisms. The advantage of this proposed simulator is that it can easily simulate varies kinds of dynamic behavior in overlay networks which can be used to evaluate trust and reputation mechanism proposed by other researches. With this simulator, research can develop and evaluate their proposed trust-related protocols easily in an overlay enrolment.

The rest of the paper is organized as follows. We review some related work in Section 2. Section 3 addresses the detail of the trust overlay simulator design. Section 4 describes the threat model in the proposed simulator. Section 5 makes simulation experiments. Then the paper concludes in Section 6.

2 Related Work

Traditional network simulators provide accurate low-level models of the network hardware and protocols but are too detailed to be effective in analysis of large scale overlay networks. For example, the *NS2 simulator* [5] is perhaps the most widely used networking simulator. In addition, there is a number of existing P2P simulators. But none of them can simulate the need of trust and reputation mechanisms, or trustworthy behavior, in overlay networks.

Packet-level P2P [6] models on an otherwise packet simulator. It creates wrappers that translate P2P level events into commands to the underlying packet simulator. *SimP2* [7] is a graph-based simulator for analysis of ad-hoc P2P networks. The analysis is based on a non-uniform random graph model, and is limited to studying basic properties such as reach ability and nodal degree. *Peersim* [8] is a Java based search framework that allows modeling of P2P overlay search algorithms, which is under the GPL open source license. *DHTSim* [17] is a discrete event simulator for structured overlays, specifically DHTs. It is intended as a basis for teaching the implementation of DHT protocols, and as such it does not include much functionality for extracting statistics. *P2PSim* [18] is a discrete event packet level simulator that can simulate structured overlays only. It contains implementations of six candidate protocols: Chord, Accordion, Koorde, Kelips, Tapestry and Kademlia. *PlanetSim* [19] is a discrete-event overlay network simulator, written in Java. It supports both structured and unstructured overlays, and is packaged with Chord- SIGCOMM and Symphony implementations.

On the other hand, varies kinds of trust and reputation models are proposed to deal with the problem of lacking trust in overlay networks. EigenTrust algorithm [13] is proposed to decrease the number if downloads of insecurity files in P2P networks that assigns each peer a unique global trust value, based on the peer's history of uploads. The framework for trust reasoning in distributed systems (FTRDS) in [14] is based on sociological studies. A reputation information exchange amongst members of the community assists on trust decisions. Elements of the work have been incorporated into Sun's JXTA framework and Ericsson Research's prototype model. The mathematical framework for modeling trust and reputation (MFMTR) in [15] is rooted in finding from the social science. The framework makes explicit the importance of social information (i.e. indirect channels of inference) in aiding members of a social

network choose whom they want to partner with or to avoid. We also proposed our own Bayesian trust model in networked computing environment [4, 10].

In this paper, we propose a novel trust overlay simulator (TOSim) which can simulate the need of trust and reputation mechanisms in overlay networks.

3 Trust Overlay Simulator Design

The architecture of overlay networks is illustrated in Figure 1. According to it, TOSim has been designed to be highly modular and configurable, without incurring in excessive overhead both in terms of memory and time. The simulated network is composed of a collection of nodes, and each of them may host one or more protocols.

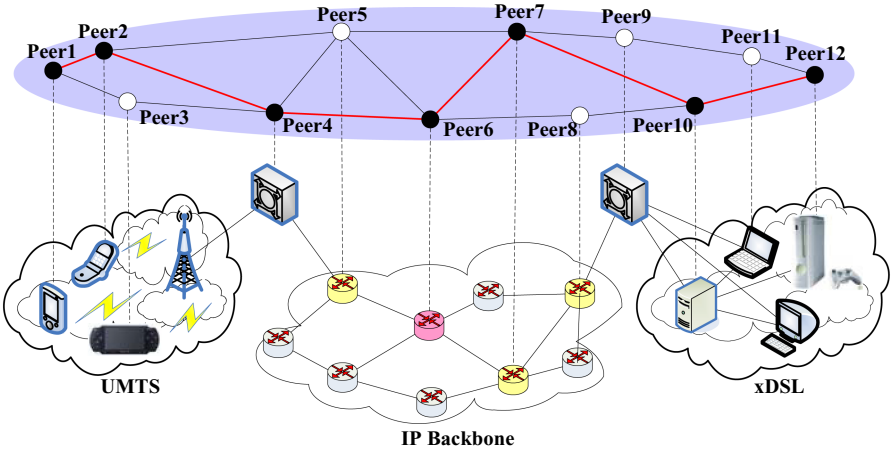

Fig. 1. Architecture of the overlay networks

A trust overlay simulator for P2P systems may have very different objectives from general-purpose networks simulators:

- **Extreme scalability.** Simulated networks may be composed of millions of nodes. This may be obtained only if a careful design of the memory layout of the simulator is performed. Being able to store data for a large number of nodes, however, is not the only requirement for large-scale simulations; the simulation engine must be optimized as well, trying to reduce, whenever possible, any form of overhead.
- **Support for dynamicity.** The simulator must be capable to deal with nodes that join and leave the network, either definitively or temporarily. This has some implications on memory management in the simulator, requiring mechanisms for removing nodes that are not useful any more.
- **Simulation all kinds of threads model.** As for developing and testing trust and reputation-based applications, the proposed simulator should simulate all kinds of behavior in the overlay network. We perform this by developing some kinds of thread models which will be described in section 4.

In addition to these requirements, the modular approach we are pursuing in this paper must be reflected in the architecture of the simulation environment as well. The idea is to provide a composition mechanism that enables the construction of simulations as collections of components. Every component of the simulation (for example, protocols or the environment) must be easily replaceable through simple configuration files. The flexibility offered by this mechanism should enable developers to re-implement, when needed, every component of the system, with the freedom of re-using existing components for fast prototyping.

Network model: We consider a typical P2P network (other overlay scheme will be developed in the future) : Interconnected, file-sharing peers are able to issue queries for files, peers can respond to queries, and files can be transferred between two peers to conclude a search process. When a query is issued by a peer, it is propagated by broadcast with hop-count horizon throughout the network; peers which receive the query forward it and check if they are able to respond to it.

Node model: Our simulator consists of good nodes (normal nodes, participating in the network to download and upload files) and malicious nodes (adversarial nodes, participating in the network to undermine its performance). We consider different threat models, where a threat model describes the behavior of a malicious peer in the network. Threat models will be described in more detail later on. Note also that some good nodes in the network are appointed as highly trusted nodes.

Simulation engine: The simulation engine is the module that will actually perform the simulation; its task is to orchestrate the execution of the different components loaded in the system. The engine adopts a time-stepped simulation model instead of more complex and expensive event-based architecture. At each time step, all nodes in the system are selected in a random order, and a callback method is invoked on each of the protocols included in that node.

Content distribution model: Interactions between peers – i.e., which queries are issued and which queries are answered by given peers – are computed based on a probabilistic content distribution model. Briefly, peers are assumed to be interested in a subset of the total available content in the network, i.e., each peer initially picks a number of content categories and shares files only in these categories. Reference [11] has shown that files shared in a P2P network are often clustered by content categories. Also, we assume that within one content category files with different popularities exist, governed by a Zipf distribution. The number of files shared by peers and other distributions used in the model are taken from measurements in real-world P2P networks [12].

Simulation execution: The simulation of a network proceeds in simulation cycles: Each simulation cycle is subdivided into a number of query cycles. In each query cycle, a peer in the network may be actively issuing a query, inactive, or even down and not responding to queries passing by. Upon issuing a query, a peer waits for incoming responses, selects a source among those nodes that responded and starts downloading the file.

Some of these goals may appear contradictory. A careful design is needed trying to obtain the best equilibrium.

4 Threat Model

In order to simulate behaviors related to trust, we consider several strategies of malicious peers to cause insecurity upload [13]. In short, malicious peers operating under *threat model A* simply try to upload insecurity files and assign high trust values to any other malicious peer they get to interact with while participating in the network. In *threat model B*, malicious peers know each other upfront and deterministically give high local trust values to each other. In *threat model C*, malicious peers try to get some high local trust values from good peers by providing authentic files in some cases when selected as download sources. Under *threat model D*, one group of malicious peers in the network provides only authentic files and uses the reputation they gain to boost the trust values of another group of malicious peers that only provides insecurity files.

Threat Model A: *Individual Malicious Peers.* Malicious peers always provide an insecurity file when selected as download source. For example, you can set Malicious peers local trust values to be *insecurity* file downloads instead of authentic file downloads.

Threat Model B: *Malicious Collectives.* Malicious peers always provide an insecurity file when selected as download source. Malicious peers form a malicious collective by assigning a single trust value of 1 to another malicious peer in the network. In terms of the probabilistic interpretation of our scheme, malicious peers form a collective out of which a random surfer or agent, once it has entered the collective, will not be able to escape, thus boosting the trust values of all peers in the collective.

Threat Model C: *Malicious Collectives with Camouflage.* Malicious peers provide an insecurity file in all cases when selected as download source. Malicious peers form a malicious collective as described above.

Threat Model D: *Malicious Spies.* Malicious peers answer a small fraction of the most popular queries and provide a good file when selected as download source. Malicious peers of type D assign trust constant values to all malicious nodes of type B in the network.

Threat Model E. *Sybil Attack.* An adversary initiates thousands of peers on the network. Each time one of the peers is selected for download, it sends an insecurity file, after which it disconnected and replaced with a new peer identity.

Threat Model F: *Virus-Disseminators.* (Variant of threat model C) A malicious peer sends one virus-laden (insecurity) copy of a particular file every 100th request. At all other times, the authentic file is sent.

All of these threat models can be easily simulated in TOSim by setting the appropriate parameters. And researchers can carry their experiments on the simulator under different kinds of threat situations.

5 Simulation Experiments and Analysis

The presented TOSim in the previous section was conceived in such a way that existing trust models could be easily simulated in the simulator. In the following, we

present the mapping of the local trust models to the introduced model and suggest some algorithmic adaptations. The investigated algorithms discussed here are proposed by Abdul-Rahman et al. [14], Lik Mui et al. [15] and Sepandar D. Kamvar et al. [13] separately, which have been introduced in related work section.

To assess the quality of the trust algorithms presented above, a series of test scenarios was developed. Each scenario simulates a different behavior pattern of trustor and trustee as a list of ratings. This pattern is then reflected by each single trust algorithm as trust dynamics. A test scenario can bring forward a specific feature or a malfunction of a trust update algorithm.

We want to stress the fact that due to the subjectiveness of trust in general, also the quality estimation of the behavior reflected in the trust dynamics is subjective. Therefore, we do not offer a ranking of trust algorithms, but instead point out the distinctive features of the algorithms, so that each user can simulator the algorithm that most closely reflects his expected behavior.

Based on the previous research [16], we choose the following two scenarios for our simulator:

MinMaxRatings: First, a series of minimal ratings is given, which is followed by a series of maximal ratings. We would expect the trust dynamics to decrease and eventually approach the minimal trust value. After switching to maximal ratings, trust should rise again.

MaxMinRatings: First, a series of maximal ratings is given, followed by a series of minimal ratings. We expect trust to rise at first. When the series of minimal ratings starts, trust should decrease again.

In MinMaxRatings (Figure 2) when the maximal ratings start, trust starts rising again in all analyzed algorithms but FTRDS. In this latter case, trust remains at the lowest level until as much maximal ratings as minimal ratings have been received.

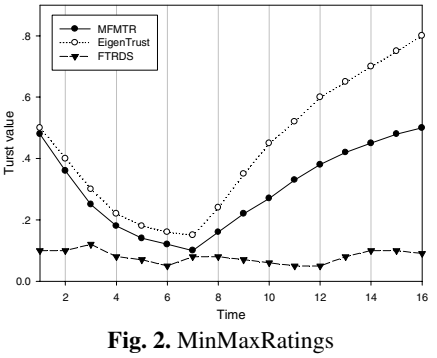

Fig. 2. MinMaxRatings

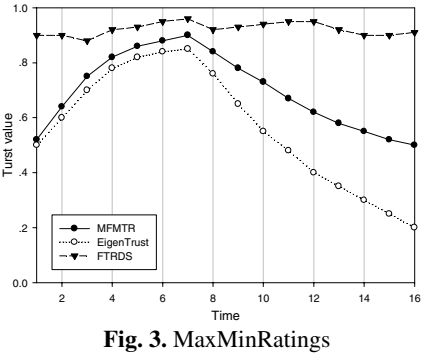

Fig. 3. MaxMinRatings

The MaxMinRatings test scenario (Figure 3) shows similar results as the previous scenario. When the minimal ratings start, trust drops with all but FTRDS's algorithm. Here, trust suddenly drops from maximum to the minimal value at the end of the scenario which is the point when more minimal than maximal ratings are recorded in the history. In both scenarios we notice that EigenTrust shows a quick reaction to the pattern change in the ratings.

6 Conclusions

In this paper, aiming at the characteristic of trust overlay networks, we presented a trust mechanism oriented overlay simulator. The presented simulator can simulate varies kinds of peers' behavior with different kinds of thread models. Besides, we simulate some experiments with three different kinds of trust mechanisms, and analyze the results. The ever increasing demand of new applications has led researchers to propose new network architectures that address limitations of the current Internet. Given the rigidity of the Internet today, overlay networks are used to implement such architectures, in the hope of gaining a large user base.

References

1. Joseph, D., Kannan, J., et al.: OCALA: An Architecture for Supporting Legacy Applications over Overlays, (2005)
2. Andersen, D., Balakrishnan, H., Kaashoek, F., et al.: Resilient Overlay Networks. In Proceedings of SOSP'01, (2001)
3. Stoica, I., Adkins, D., Zhuang, S., et al.: Internet Indirection Infrastructure. In Proceedings of SIGCOMM'02, (2002)
4. Wang, W., Zeng, G. S., Yuan, L. L.: A Semantic Reputation Mechanism in P2P Semantic Web, In: Proceedings of the 1st Asian Semantic Web Conference (ASWC), LNCS 4185 (2006) 682-688
5. Berkeley/LNBL/ISI. The NS-2 Network Simulator. http://www.isi.edu/nsnam/ns/
6. He, Q., Ammar, M., Riley, G., et al.: Mapping Peer Behavior to Packet-level Details: A Framework for Packet-level Simulation of Peer-to-Peer Systems. In Proceedings of MASCOTS 2003, Orlando, FL, (2003)
7. Kant, K. Iyer, R.: Modeling and Simulation of Adhoc/P2P Resource Sharing Networks. In Proceedings of TOOLS'03, (2003)
8. Jelasity, M., Montresor, Jesi, A. G. P.: Peersim Peer-to- Peer Simulator, (2004). http://peersim.sourceforge.net/
9. Mark J., Alberto M., Ozalp B. Gossip-based aggregation in large dynamic networks. ACM Transactions on Computer Systems, 23(3) (2005) 219–252
10. Wang, W., Zeng, G. S., Liu, T.: An Autonomous Trust Construction System Based on Bayesian Method, In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'06), Hong Kong, China: IEEE Computer Society Press, December 18-22 (2006) 357–362
11. Arumugam, M., Sheth, A., Arpinar, I. B.: Towards Peer-to-Peer Semantic Web: A Distributed Environment for Sharing Semantic Knowledge on the Web. Technical report, Large Scale Distributed Information Systems Lab, University of Georgia, (2001)
12. Saroiu, S., Gummadi, P. K., Gribble, S. D.: A Measurement Study of Peer-to-Peer File Sharing Systems. In Proceedings of Multimedia Computing and Networking 2002 (MMCN '02), San Jose, CA, USA, January (2002)
13. Kamvar, S. D., Schlosser, M. T., Garcia-Molina, H.: The Eigentrust algorithm for reputation management in p2p networks, In Proceedings of the 12th International Conference on World Wide Web, ACM Press, (2003) 640–651
14. Abdul-Rahman, A.: A Framework for Decentralized Trust Reasoning. PhD thesis, Department of Computer Science, University College London, (2004)

15. Mui, L.: Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks. PhD thesis, Massachusetts Institute of Technology, (2003)
16. Kinateder, M., Baschny, E., Rothermel, K.: Towards a Generic Trust Model – Comparison of Various Trust Update Algorithms. iTrust 2005, LNCS 3477 (2005) 177–192
17. 17."DHTSim" accessed 01-May-2006. [Online]. Available: http://www.informatics.sussex.ac.uk/users/ianw/teach/dist-sys/
18. "P2Psim: A Simulator for Peer-to-Peer (P2P) Protocols," 2006. [Online]. Available: http://pdos.csail.mit.edu/p2psim/
19. Virgili, U. R.: "PlanetSim: An Overlay Network Simulation Framework," (2006). [Online]. Available: http://planet.urv.es/planetsim/