TDDC03 Projects, Spring 2004

Trust and reputation in file-sharing Peer-to-Peer systems

Peng Mu, Xianying Cheng

Supervisor: Claudiu Duma

Trust and reputation in file-sharing Peer-to-Peer systems

Peng Mu, Xianying Cheng
ICI2003, Linkoping University
{penmu452, xiach882}@student.liu.se

Abstract

With Peer-to-Peer (P2P) systems becoming more and more popular, trust and reputation models are wider discussed to solve P2P's problems: how to choose a trustable peer, and how to avoid downloading files from malicious peers etc. This report starts from the introduction of P2P system, examines four different trust and reputation models in P2P systems and analyzes their different aspects of setting-up, measuring, storing and updating the values of trust and reputation, and then analyze security problems of different models. Before giving our conclusions, we talk about the practical situations for applying different models, and how to select features of models with given system characteristics.

1. Introduction

Peer-to-peer (P2P) systems have shown an enormous success nowadays. The positive features of scalability, autonomy, robustness, and anonymity significantly contribute to the success of many P2P systems [1]. Good scalability allows flexible network size and a large quantity of simultaneous users. Autonomy refers that P2P systems are self-organized and usually do not require any special administrations [2]. Robustness is gained from the decentralized and distributed nature of P2P system, which gives the P2P system (not the single peer) the potential to be robust to faults or intentional attacks. Anonymity is a favorite feature for P2P users, for there is no requirement to register or authenticate to enroll a P2P network, no access control to search files to the network, or download files from another peer.

However, several negative features are also brought into such systems which may cause serious security problem. In particular, the lack of coordination and control among the peers might open a door to the possible misuses and abuses. The malicious peers can exploit such systems to spread malicious code such as Trojan Horses, viruses, and spam. Access control has been traditionally used to protect against malicious parties. However, access control depends on the existence of identification and authentication services which can not be, in general, provided within a P2P setting. Moreover access control can't prevent peers from providing inconsistent quality of services. So how to help a peer to locate a file with good quality and find the peer who can offer good service automatically become a hard problem in P2P system. In this situation trust and reputation (T&R) models are invented to accomplish two missions: both protect against malicious peers and help to locate good peers and files.

In this paper we first briefly introduce what P2P system is, and how it works in section 2. In recent years many different T&R models have been proposed. We select four models and analyze them in section 3. Our analysis identifies a number of aspects which are common to all the models. We describe these aspects and show how four different models address them. T&R models are susceptible to specific attacks. So in section 4 we also analyze the security attacks and show whether different T&R models work against them. Based on all analysis we give our results on where to apply these models. At the end we present our conclusion.

2. P2P systems

Pure **Peer-to-peer (P2P)** systems are systems where all the nodes have the same role and there are no nodes with a special responsibility to monitor or supervise the network behavior. In such systems each peer acts both as a client and as a server, all peers are both consumers and providers of resources and can access each other directly without centralized coordination. There are hybrid P2P systems with a central server, but in this paper we will not discuss them. In the following, without special note, all P2P systems will refer to pure P2P systems. Compared with a centralized system, a P2P system provides an easy way to aggregate large amounts of files residing on the edge of Internet or in ad-hoc networks with a low cost of system maintenance [3].

According to system function, current P2P systems can be classified to three categories: file sharing, distributed processing (or computing) and instant

messaging [4]. This paper focuses talking about the file-sharing P2P system.

A file-sharing peer-to-peer system is composed of many peers, each peer storing a collection of files. File-sharing are realized in two phases: Search phase and Download phase. 1) Search phase searches for peers storing the requested file. 2) Download phase downloads the file from a peer. There are many search algorithms for pure P2P systems, and the most basic one is flooding algorithm, where the query (search message) is propagated to all neighbors within a certain radius. There are also a lot of papers addressing new algorithms to improve this flooding algorithm [5] [6], but here we just explain the typical flooding search algorithm to show how it works. And the later discussion will base on it.

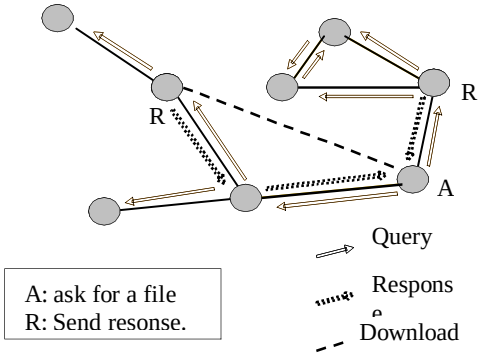

Figure 1 **A file-sharing peer-to-peer system with flooding search algorithm**

When a peer wishes to find a file from the system, it sends a broadcast query the system (see figure 1). The query is propagated through the network up to a certain number of hops (search radius). Any peer that receives the query will check if it has some files satisfying the query. If so, it replies to the query originator, and this peer is called responder. Whether this peer has the queried files or not, it will forward the request along all the links it maintains (flooding), except the one from which the query has arrived. The originator then selects a responder to download the requested file [4] [7].

3. Overview trust and reputation models

3.1 Definitions

Though trust and reputation models have been popular discussed in many papers, there is no universal agreement on the definition of trust and reputation. In this paper, we adopt the following definitions from Yao Wang et al's paper [3].

Trust – 1) A peer's belief in another peer's capabilities, honesty and reliability based on its own direct experiences. A peer has two different direct experiences resulting to two different trusts: downloading files from a peer comes out the trust in a peer to be a file-provider (we call this **direct trust**), and getting recommendations from a peer results the trust in a peer to be a reference (This is called **reference trust**). 2) A peer's belief in a file's quality on its own direct experiences. A file has only direct trust value.

Reputation – 1) A peer's belief in another peer's capabilities, honesty and reliability based on recommendations received from other peers. 2) A peer's belief in a file's quality according to recommendations from other peers.

The difference between trust and reputation is that trust is based on peer's own experiences with another peer. Here is an example to make those concepts clear (see figure 2). Peer A has a downloading experience with peer B, and then A can set up its direct trust value for B (positive or negative). A is quite sure that this value is real since it come from its own experience. Now peer A wants to download a file from peer D, but A has no idea on D's trustworthiness. Then A asks peer B and C for their opinion on D. B and C sends their direct trust values about peer D if they have. From A's view, B and C are called **references** for D, and the values from them are **recommendations** on D. With the recommendations A can compute one value with a specific algorithm, and this value is called D's reputation. A is not sure if this value (reputation) is trustable since A doesn't know if C is honest or if B is absolutely trustworthy, or whether this reputation is correct according to its own standard. Furthermore suppose A decides to download from D according to its standard (depending on models). After the downloading –a direct interaction with D, A setups its opinion on D's capabilities, honesty and reliability and create its direct trust value to D. At the same time A can evaluate recommendations from B and C. For C, A will setup reference trust value to B and C high or low depending C's recommendation correct or not. This example is a general concept, in the following paragraph we will see different models vary a lot in the implementation.

3.2 Common aspects

There are many trust and reputation models with different features and algorithm used in different P2P systems currently. By looking at those different

features, we can get an overview of the trust and reputation models.

1) What to trust: the peer (peer-based trust), the file (file-based trust) or both.

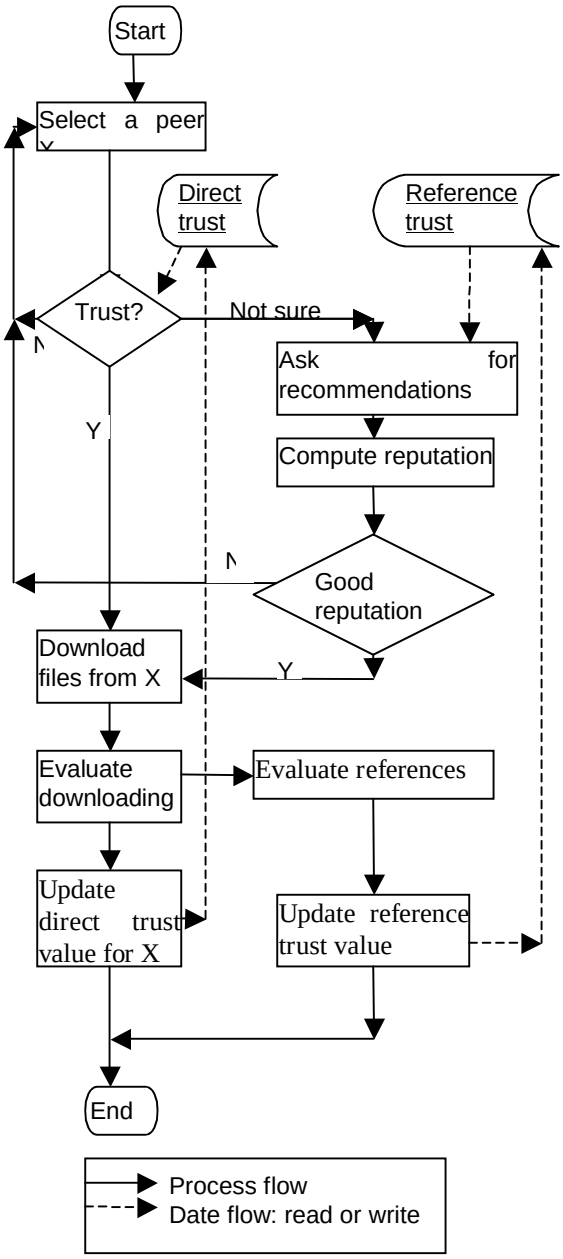

Figure 1 An example of trust and reputation model

Peer-based model is to set up the trust for peers, depending on peer-id which should be well designed in order to be tamper-resistant. This is most common used in all models. Peer-based model is quite simple to implement and use. But peer-based trust and reputation will be nullified if peer's identifier is changed. A malicious peer can make use of this to take a fresh identity to get rid of its bad reputation.

There is one important enhancement in peer-based model: **context-specific**. This is based on the fact that one peer is trustable in one category but might not be trustable in another category. A music fan can guarantee music file he shared is in good quality since he has already played them, but might not guarantee quality of computer games. Context-specific peer-based model can narrow down the trustable peers especially when many types of files are provided by each peer.

File-based model associates trust and reputation to file-id, which should be well designed in case of forging. Normally a digest from the file content can be used as file-id. File-based model can work together with peer-based model to solve the weakness of peer-based model, but more computation for the file-id is required on all shared files at a peer. File-based model makes a good file trustable regardless of who offers it, this means file-based reputation has potential wider scope and longer life cycle than peer-based reputation, but this also requires that a file should be spread wide enough to set up its reputation.

2) What information to use: trust, reputation or both.

Trust: in the following sections, when we use trust only, we normally mean direct trust. 1) Direct trust in a peer: Set up trust values for a peer to be a file-provider according to its previous direct downloading experience. This is used to decide from which peer to download a file. 2) Direct trust in a file: Set up trust values for a file for its quality. This is used to select which file to download when a peer receives some similar file-ids.

Reputation: With a specific peer or file, ask other peers for their direct trust in this peer or file. Normally when reputation is used, another type of trust is used together. **Reference trust** is trust values in a peer to be a reference according to its previous experience. Sometimes it is used to help decide whose recommendation to adopt.

3) How to measure trust or reputation values: binary, limited-multiple levels or unlimited levels.

Binary: use two levels (+/-, 1/0 etc) to record positive or negative opinion.

Limited-multiple levels: A fixed number of levels more than two to record more information than binary.

Unlimited levels: Use natural number, or other continuous number to record values.

4) Where to store trust or reputation values: local or decentralized.

Local: each peer keeps all the trust values of others it has collected. The values might be stored in a vector, matrix or some other data structure. But with time, values of more and more peers will be collected if in a big network, then to store the values locally become a heavy requirement.

Decentralized: Using some data structure and algorithm, each peer store values of several peers to share the workload. And several peers store the same data in order to detect cheating. Those values are global, and everybody can update it with a certain mechanism. One peer doesn't store all others' values. This gives more scalability to a system, but since values are stored at other peers instead locally, and everybody can change these values, a peer has no belief in these values. A peer can't make sure if those values from its own experience, normally not true since it's global. So this model only fit in reputation model instead of trust model. And how to design the storage architecture, how to access the data efficiently, and what if many peers intrigue together to cheat are problems to such models.

5) How to select the destination peer to download: select best, blind select.

Select best is to select the best peer based on the evaluation on the received trust and reputation values,

Blind select is to select randomly from a set of peers when all of a subset peers seem to satisfy the requirement.

6) When to update values of trust and reputation: Interaction triggered, event triggered or time triggered

Interaction triggered: That's the basic trigger to update the values. One peer keeps values of trust and reputation on others. Those values will only be updated according to the peer's own interaction experience.

Event triggered: Values will be updated when peers receive some positive or negative events (like votes or complaints), which may come from their own interaction or others interaction.

Time triggered: When peers are idle, they communicate with each other to exchange the trust and reputation data in order to keep the values up to date. That is: Trust and reputation values will be updated not only according to its own interactions, but also according to other's interactions. For such system normally time trigger works as a complement of interaction trigger.

3.3 Common aspects analysis

In this paragraph, we use XRep [4], P-Grid [8], Bayesian [3] and Mod [9] to represent different models used in different papers. In table 1 we make a summary of common aspects from the previous discussion.

Table 1 common aspects summary

Aspect	Classification
What to trust	peer-based, file-based or both
What to use	trust, reputation or both
How to measure values	Binary, limited-multiple levels, or unlimited levels
Where to store values	local, central or decentralized
How to select destination	select best, blind select
When to update values	Interaction triggered, event triggered or time triggered

Table 2 Aspect analysis of 4 models

Aspect	XRep	P-Grid	Bayesian	Mod
What to trust	Peer and file	Peer or with context	Peer with context	Peer with context
What to use	Reputation with reference trust	Reputation only	Trust first, then reputation with reference trust	Trust first, then reputation with reference trust

How to measure values	File: binary + -; peer: unlimited number Reputation: any	Unlimited	3 levels	4 levels
Where to store values	Local	Decentralized	Local	Local
When to update values	Interaction	Event-complaint	Interaction, Time	Interaction
How to select the destination	Blind select on digest	Best	Blind select based on threshold	Best

4. Security

All the four T&R models can help to locate trustable peers in their designed situation, but to deploy them to practice we have to analyze their security features since some models are susceptible to specific attacks. Because the resiliency of a T&R model to such attacks is paramount for its real deployment within a P2P system. We list some important attacks in T&R based P2P systems to see how it works, and analyze how well different models can work against them.

4.1 Attacks to T&R based P2P systems

There are two main categories of attacks: unintentional attacks and malicious attacks [4]. In malicious attacks there are many different detail attacks according to how they work. Some attacks are inherited from general P2P systems, but they still happen in T&R based P2P systems.

Unintentional attacks are that honest peers redistribute a file which was tampered without their knowledge. That is, some innocent peers are made use of to spread tampered files. This attack can only be avoided by file-based model with file-id integrity check.

Malicious attacks mean that some peers actively try to distribute spam or hostile content including virus, worms, and Trojan horses. According to how the attacks happen, there are four main malicious attacks: self replication attack, man in the middle attack, Pseudo spoofing and shilling.

Self replication: This attack is based on the fact that in P2P systems there is virtually no way of verifying the source or content of a message. A malicious peer answers positively to all queries, and then returns tampered content (maybe with the searched name or condition). Peer-based model can setup bad reputation for this peer and avoid downloading from it.

Man in the middle. This kind of attacks is that a malicious peer lies between two honest peers. Assume that A is a peer searching for a file, B is a peer that has the file A is looking for, and D is a malicious peer. First, A broadcasts a query message and then B responds. Malicious peer D intercepts B's response message and modifies the IP field to contain D's IP address, and then sends back to A. A decides to download the file from D, which provides a fake file, possibly even a hostile version. Without file-id integrity check, this attack can't be prevented either.

Pseudo spoofing: Pseudo spoofing attackers create and control multiple simulated identities which would like to give fake recommendations on it.

Shilling: Shilling attacker creates multiple recommenders with real IP address in order to influence the reputation on a doctored file or on a malicious peer.

Both Pseudo spoofing and shilling attacks are designated to positive reputation-only model. If direct trust is used to make decision or reference trust is combined together with reputation to make decision, this attacker will not succeed due to lack of direct interactions to give good trust values. Negative reputation-only model is also immune to this attack, because in negative reputation-only model every peer should launch complaints instead of positive votes, so simulated recommendations have no chance to play in role.

4.2 Attack analysis of 4 models

Table 3 Attack analysis of 4 models

Attack	XRep	P-Grid	Bayesian	Mod
Unintentional	Y	N	N	N
Self replication	Y	Y	Y	Y
Man in the middle	Y	N	N	N
Pseudo spoofing	Y	Y	Y	Y
Shilling	Y	Y	Y	Y

Y means this model can work against this attack.

N means this model can not work against this attack.

5. Result of models analysis

5.1 Apply models to different P2P settings

XRep model is focusing to provide integrity against all kinds of attacks, so it emphasizes a lot on the encryption and integrity check. This specially fit in the environment where there are many attacks.

P-Grid model mainly talks about how to store the values of trust and reputation distributive and release the storage burden of a single peer, and improve the scalability. So this model is focusing on application in a very big network where it's quite heavy for a single peer to store all the values of trust and reputation.

Bayesian model puts its emphasis on the communication of peers to exchange and update values of trust and reputation. So it's faster to adapt changes and more efficient to apply to frequently changing network.

Mod model targets on Virtual communities grounded in a real-world social trust characteristics. Unambiguous, easy to understand and simple to implement is its expectation. This model is very like Bayesian model except for no exchanging values between recommenders. It's like a simplified version of trust and reputation model. It's designed for virtual communities.

5.2 Select features according to system characteristics

Different P2P system might have different characteristics. Different T&R models have different features. It's important to see that a feature in one model might be learned to another model to make it more fit to a certain real situation. Here we list some practical system characteristics and give suggested features or models.

Frequency of peer-id updating or new peer enrolling: file-based model (like XRep) works better for high frequency updating of peer-id, as in the network expanding phrase with many new peers enroll. More dynamic model (like Bayesian) will work better for the frequently changed network.

Percentage of malicious peer or attacks (cheating rate): the more percentage of malicious attacks, XRep works better than others since it uses encryption to protect the confidentiality, integrity check on both peer-id and file-id, and also it has true vote phrase to beat against attacks. For low percentage of malicious attack, XRep is not preferred since it needs too much computation locally and also on other peers (public key encryption and decryption, digest computing), much more traffic interactions than other models (5 phases). P-Grid is neither good at honest system, because if too few peers fire complaints, it's almost the same as the P2P system without trust and reputation.

Network size: Decentralized storage has more scalability than local storage. P-Grid protocol works better in very large network and also for peers who has limited storage resource.

Trust and reputation system is based on experience history, that is, there must be enough interactions before a peer or a file set up its reputation. So none model works in the very beginning of whole network, but Bayesian will set up the reputation faster than others due to its high update frequency.

Network category: file-based model (like XRep) works better than peer-based model in highly shared network (people share same file with high frequency), like interest community for music, movie etc.

6. Conclusion

T&R models are focusing to solve two big problems in P2P system: protect against malicious peer and locate trustworthy peers and files with good quality. To show how four T&R models gain this and how well they gain this, we analyzed common aspects of T&R models, and compared different implementations of four models. Attacks to general P2P systems are analyzed to show the contribution of T&R models to P2P security. Based on all the analysis we gave suggestion on the use situation where to apply these models. Our analysis result shows that different models have different emphasis, and it's possible to combine advantageous features of different models and make a better one for a certain situation.

As we see trust and reputation models solve some problems in P2P systems, however they also have some limitation, like peer –id changing will nullify reputation, and to protect against more attacks needs more interactions and computation, which add the workload of peers or networks.

7. References

[1] Farag Azzedin, Muthucumaru Maheswaran. "Trust Modeling for Peer-to-Peer bases Computing Systems". In *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, Nice, France, April 2003

[2] Marcelo Werneck Barbosa, Melissa Morgado Costa, Jussara M. Almeida, Virgflio A. F. Almeida Using Locality of Reference to Improve Performance of Peer-to-Peer Applications. ACM SIGSOFT Software Engineering Notes,

Proceedings of the 4th International Workshop on Software and Performance, January 2004

[3] Yao Wang, Julita Vassileva, Trust and reputation model in peer-to-peer networks. In *Proc. of The Third IEEE International Conference on Peer-to-Peer Computing*, Linköping, Sweden. September 1-3, 2003

[4] Ernesto Damiani , De Capitani di Vimercati , Stefano Paraboschi , Pierangela Samarati , Fabio Violante, A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, Washington, DC, USA, November 18-22, 2002, pp 207-216

[5] Vana Kalogeraki, Dimitrios Gunopulos, D. Zeinalipour-Yazti XML schemas: integration and translation: A local search mechanism for peer-to-peer networks *Proceedings of the eleventh international conference on Information and knowledge management*, November 2002

[6] Qin Lv, Pei Cao, Edith Cohen, Kai Li, Scott Shenker Networks: Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th international conference on Supercomputing*, June 2002

[7] Marti, Sergio; Garcia-Molina, Hector. Identity Crisis: Anonymity vs. Reputation in P2P Systems, In *Proceedings of Third IEEE International Conference on Peer-to-Peer Computing*, Linköping, Sweden, September1-3 2003.

[8] K. Aberer and Z. Despotovic. "Managing trust in a peer-2-peer information system". In *Proc. of the Tenth International Conference on Information and Knowledge Management (CIKM 2001)*, Atlanta, Georgia, November 2001.

[9] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proc. of the Hawaii International Conference on System Sciences*, Maui, Hawaii, January 2000.