Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 94 (2016) 360 – 365

Procedia Computer Science

International Workshop on Big Data Security and Trust Computing (BDSTC-2016)

A Novel Reputation Management Mechanism with Forgiveness in P2P File Sharing Networks

Mingchu Li[a], Junlong Wang[a], Kun Lu[a,*], Cheng Guo[a,**], Xing Tan[b]

[a] *School of Software Technology, Dalian University of Technology, Dalian, 116621, China*
[b] *School of Information Technology, York University, Toronto, M3J 1P3, Canada*

Abstract

In peer-to-peer (P2P) file sharing networks, it is common practice to manage each peer using reputation systems. A reputation system systematically tracks the reputation of each peer and punishes peers for malicious behaviors (like uploading bad file, or virus, etc). However, current reputation systems could hurt the normal peers, since they might occasionally make mistakes. Therefore, in this paper, we introduce forgiveness mechanism into the EigenTrust reputation system to reduce such malicious treatments and give them opportunities to gain reputation back. Particularly, we take four motivations (the severity of current offence, the frequency of offences, the compensation and the reciprocity of the offender) into consideration to measure forgiveness. The simulation work shows that the forgiveness model can repair the direct trust breakdown caused by unintentional mistakes and lead to less invalid downloads, which improves the performance of P2P file sharing systems.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Trust Management; Forgiveness; File sharing system; Reputation system.

1. Introduction

Trust issues occur along with the quick development of the open, anonymous, and distributed peer-to-peer (P2P) applications. Traditionally, reputation systems[2,4,5,6,7] for P2P sharing network focused on the accuracy to locate a reliable parter to obtain good services and ensure robustness against malicious behaviors like malicious ratings or inauthentic uploads. However, current trust and reputation mechanisms emphasize the need to 'punish' an offender, while they neglect of accounting for alternative ways to repair the offence[3]. Generally, the good peers who downloaded unreliable contents from other peers would give negative evaluations for others, no matter what the real identity of the file provider is. In such case, a mistake behavior of a good peer is also treated as malicious and given a negative evaluation, which is strict and unfair for good peers. In fact, Liang et al.[1] found that, in KaZaA, users often acciden-

* Corresponding author. Tel.: +86-130-5053-1026 ; fax: +86-0411-6227-4467.
** Corresponding author. Tel.: +86-158-4063-1531 ; fax: +86-0411-6227-4455.
E-mail address: lukun@dlut.edu.cn, guocheng@dlut.edu.cn

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.08.055

tally create damaged files and inject them into P2P file sharing networks. We argue that unintentional mistakes like this should not be considered as malicious behavior. In contrast, they should be forgiven by the reputation mechanism.

In this paper, we explore this problem and propose a novel model to incorporate forgiveness[8] into EigenTrust reputation system to address the issue above. In our model, forgiveness is regarded as an additional feature to build an enhanced reputation system, thus fixing the trust break-down made by unintentional mistakes. Here, the peer who uploads a malicious file is considered as the *offender*, and the peer who downloads the malicious file is known as the *victim*. The process of transmitting a malicious file is called an *offence*. Based on the provisions of Vasalou[9], we introduce four motivations (the severity of current offence, the frequency of offences, the compensation, and the reciprocity of the offender) for the victims to perform forgiveness to ensure the future cooperative interactions. It is worth noting that an abuse of forgiveness could result in an vulnerable situation since attack behaviors by malicious peers may also be forgiven. In other words, the malicious attackers must not be forgiven. We also address this issue by giving a comprehensive method to evaluate forgiveness in our model. The simulation works under Quantitative Trust Management (QTM) project developed by University of Pennsylvania to show the effectiveness of our proposed model.

The rest of this paper is organized as follows: Section 2 illustrates the proposed model in detail. Section 3 represents the simulation experiments and analysis. Finally, we conclude our work in Section 4.

2. Proposed model

2.1. System Overview

In P2P file-sharing systems (e.g. Gnutella), files are stored on the computers of peers. And they are exchanged through a direct connection between the file requester and provider. Every peer can issue a file request by sending a query packet to her/his neighbors. The neighbors who received the query packet may transfer it to their neighbors, and going on in this way before the TTL (time to live) reduces to 0. After choosing a file owner from the response peers, the requester establishes an HTTP connection with the file owner and begins to download the file. After the above transaction process completes, the requester will give a feedback to the provider if a trust/reputation mechanism is utilized in the P2P file application. In our model, EigenTrust algorithm is utilized to build trust among peers, while forgiveness is introduced as an additional module to repair the direct trust break-down caused by some unintentional mistakes. The victim will forgive the offender with a probability value derived from our model. In this way, a forgiveness process can be conducted between the victim and offender.

With the above knowledge, we give the detail design of our model in the following sections. The notations used in this paper are listed in Table 1.

2.2. EigenTrust Reputation System

The origin EigenTrust uses a simple method to compute local trust: If the download is satisfactory, then $sat(i, j)$ adds by 1, otherwise, $unsat(i, j)$ adds by 1. Then the local trust is as Eq. (1) shows:

$$s_{ij} = sat(i, j) - unsat(i, j) \tag{1}$$

The above measurement of direct trust is easy but exposes drawbacks. For example, it cannot distinguish the case like $sat(i, j) = 2$ and $unsat(i, j) = 0$ with the case of $sat(i, j) = 10$ and $unsat(i, j) = 8$. The later case may indicate that peer j is a malicious peer. Therefore, we propose a direct trust calculation method in Eq. (2):

$$dt_{ij} = \begin{cases} (1 - \alpha^{s_{ij}}) \cdot sat(i, j)/tr_{ij}\,, & if \quad s_{ij} \geq 0; \\ 0 \qquad\qquad\qquad\quad\;\; , & otherwise. \end{cases} \tag{2}$$

where α is a system parameter.

Eq. (2) indicates the trust value of i to j is increased from 0 to α right off when a client requests from a server for the first time and he or she is satisfied with that transaction. However, as the times of satisfactory transactions increase, the growth rate of direct trust value slows down. This principle is in accord with the human experience: when two