A Malicious Peers Detection Framework for Peer-to-Peer Systems

Xianglin Wei[1], Jianhua Fan[1], Tongxiang Wang[1], Qiping Wang[1]
[1]PLA University of Science and Technology
Nanjing, 210007, China
wei_xianglin@163.com, ricy1213@126.com

JDIM
Journal of Digital Information Management

ABSTRACT: *A number of reputation mechanisms are introduced in recent years to alleviate the blindness during peer selection in distributed P2P environment where malicious peers coexist with honest ones. They indeed provide incentives for peers to contribute more resources to the system, and thus, promote the whole system performance. However, little attention has been paid on how to identify the malicious peers in this situation. In this paper, a general framework is presented for detecting malicious peers in Reputation-based P2P systems. Firstly, the malicious peers are divided into various categories and the problem is formulated. Secondly, the general framework is put forward which mainly contains four steps, i.e. data collection, data processing, malicious peers detection and malicious peers clustering. Thirdly, an algorithm implementation of this general framework named IDEA is put forward based on Principal Direction Divisive Partitioning and K-means clustering. Finally, a series of simulation experiments are conducted to evaluate the effectiveness of IDEA. Simulation results have shown that IDEA can precisely distinguish and cluster the malicious peers.*

Subject Categories and Descriptors
C.2.1 [Network Architecture and Design]: Distributed networks; **I.5.3 Clustering]: Algorithms**

General Terms: Distributed networks, P2P, Clustering

Keywords: Peer-to-Peer, Malicious Peers, Framework, Reputation

Received: 21 February 2015, Revised 1 April 2015, Accepted 9 April 2015

1. Introduction

Peer-to-Peer (P2P)technology has become a useful building block for a variety of distributed applications, attributing to its outstanding characteristics such as anonymous and openness. Various P2P commercial products and free software packages are available for people's daily life, such as file sharing, cooperative computing, video on demand, live streaming and so on. The decentralized, cooperative and self-organizing nature of P2P systems help to overcome or at least mitigate many challenges faced by the traditional Client/Server systems. Measurement results have shown that unencrypted P2P traffic accounts for about 20% of the current inter domain traffic in the Internet and its traffic accounts for even more than 50% in the last years[1]. Nowadays, P2P traffic is still an important component of the Internet [2]. However, the self-organizing and decentralized nature also brings the inherent vulnerabilities of management and security to the P2P systems that place restrictions on the efficiency of the content delivery.

In order to stimulate peers to contribute resources and assist peers to select the most trustworthy collaborators, several reputation management schemes have been proposed [3][4]. These schemes try to evaluate the transactions performed by peers and assign reputation values to them to reflect their past behavior features. These reputation values will be the basis for identifying trustworthy peers to reduce the blindness of peer selection and to stimulate the collaboration among all the members. Although these schemes have been proved to be theoretically attractive, they still have a long way to go before practical deployment since they cannot gracefully

cope with the security and management issues including self-promoting, whitewashing, slandering, collusion [5][6], Botnets [7] and Sybil attack [8]. A report from Wired magazine stated that 45% percent of the executable files downloaded through KazaA contain malicious codes like viruses and Trojan horses [9]. Adar et al. claimed that nearly 70% of Gnutella users share no files [10].

In contrast to the designation of novel reputation mechanism or incentive schemes, how to detect malicious peers has attracted little attention in the last years. As a burgeoning area, researchers have put forward a few detecting methods under various environments for particular reputation schemes. However, a general framework for detecting malicious peers in P2P systems is still missing for now and is the focus of this paper.

In the framework presented in this paper, we focus on the general architecture of the detecting system rather than some particular detecting algorithm. Many existing algorithms presented before can be treated as the particular implementation of our framework. Based on the framework, several detecting algorithms can be realized in various environments. Moreover, a detecting and clustering algorithm based on Principal Direction Divisive Partitioning (PDDP) and K-means with in this framework is shown and evaluated.

The rest of this paper is organized as follows. Section 2 summarizes related work. Section 3 presents the detection environment and formulates the problem. The general framework is detailed in Section 4. A detecting algorithm based on PDDP within this framework is shown in Section 5.Section 6 evaluates the detecting algorithm presented in Section 5. Section 7 is a brief conclusion of this paper.

2. Related Work

The reputation mechanisms developed for P2P systems help participants decide who to trust, encourage trustworthy behavior, and deter dishonest participation by providing a means through which reputation and eventually trust can be quantified and disseminated. In recent years, a lot of reputation mechanisms have been proposed by researchers, such as EigenTrust [4], iRep [11] and so on. The malicious peers are those peers who utilize the systems' resources to fulfill their particular targets, such as free-riding, collusion etc., through violating the systems' principle. In order to find out these malicious peers, researchers have presented a few methods.

Mekouar et al. proposed a Malicious Detector Algorithm in [12] to detect 'liar peers' that send wrong feedback to subvert reputation system. That is, after each transaction between a pair of peers, both peers are required to generate feedback to describe the transaction. If there is an obvious gap between the two pieces of feedback, both are regarded being suspicious. Ji et al. raised a group based metric for protecting P2P network against Sybil attack and collusion by dividing the whole network into some trust groups based on global structure information which is hard to obtain [13]. In [5], Lian et al. recommended various collusion detection approaches including pair-wise detector and traffic concentration detector with data of Maze file sharing application based on trace analysis. In order to guarantee the correctness of the reputation calculation, Despotovic et al [14] compared the probabilistic estimation and social network methods. Besides, they also identified four classes of collusive behavior. Tehale et al used the false message concept for identifying and verifying the Sybil nodes in the network [15]. Selvaraj et al presented a comprehensive survey of security issues in Reputation Management Systems for P2P networks in [16]. Jin et al proposed a peer based monitoring method in Peer-to-Peer Streaming environment [17]. Koutrouli et al provided a thorough view of the various credibility threats against a decentralized reputation system and the respective defense mechanisms [18].

Liu et al have presented an upload entropy scheme to prevent collusions and further enhance robustness of private trackers' sites [3]. But the threshold of this scheme needs to be settled manually. Moreover, Lee et al. put forward a simplified clique detection method to detect the colluders [19], but their method is restricted to colluders who form a clique. Ciccarelli et al [20] surveyed the literature on P2P systems security with specific attention to collusion, to find out how they resist to such attacks and what solutions can be used. On the one hand, they summarized five collusive categories, and then investigated the influence of collusion on various applications. On the other hand, they discussed the feasible solutions that can be utilized to resist collusions, such as game theory and so on. Liu et al [21] brought forward a new strategy based on trust value and considers both the quality and the number of shared resources to avoid the phenomenon of free riding. Moreover, they also sketched collusion, slander and other misbehavior during strategy design. A Multiscale Principal Component Analysis (MSPCA) and Quality of Reconstruction based method PeerMate was proposed in our former work [22], it can efficiently detect malicious peers for P2P systems. However, PeerMate cannot find out malicious peers which initiate Sybil attack to the system. Moreover, PeerMate needs a reconstruction threshold, which can remarkably impact its efficiency. SMART [23] was put forward based on MSPCA and Shewhart control chart [24].

In this paper, we aim to build a general detecting framework which can be implemented through a variety of methods and can be applied in many current proposed P2P systems.

3. Detecting Environment and Problem Statement

3.1 Reputation-based Peer to Peer System

There are two typical architectures of the reputation-based P2P systems, i.e. the centralized and the decentralized architecture.

In the centralized architecture, there are a few central servers or facilities in the system. They are in charge of collecting, updating, disseminating and deleting the reputation values of each peer according to their behaviors during the content delivery, such as uploading or downloading chunks. The whole system and each peer will provide service to a peer based on its reputation value.

In contrast, no central server is deployed to collecting the reputation values in the decentralized RP2P system. All the peers need to calculate the reputation value of its neighbors based on their historical transaction records. Therefore, each peer may have distinct view of the same peer's reputation value due to their different transaction records.

Based on the peers' reputation value, each peer can judge its neighbors' trustworthiness more precisely than the situation without reputation information. However, Reputation system also brings new security threats to the system, such as Sybil attack, collude attack and so on. Therefore, a framework is needed to distinguish these malicious peers from the honest ones.

3.2 Malicious Peers

According to their different behavior features, the malicious peers can be divided into various categories as shown in [22][23], and hence it is hard to summarize all the categories comprehensively due to the complexity of the behaviors. Here, we mainly focus on the following categories and evaluate our framework based on these categories. Note that MP stands for Malicious Peer in the following analysis.

· **MP1:** Peers that utilize P2P's resources without providing appropriate amount of resources (i.e., free-riders), such as BitTyrant and BitThief clients. This is because many peers are only in pursuit of maximizing their own profit while lack enthusiasm for contributing services to the entire system;

· **MP2:** Peers that upload inauthentic objects to persecute the community, such as the peers controlled by the music industry which inject fake files to KaZaA. This is mainly due to the fact that many contents shared in the P2P community are copyrighted materials which violates the copyright owners' profit;

· **MP3:** Peers that collude with each other. They can be organized to a collusive group or chain through collaborating with each other to promote their reputation values or to decrease other peers' reputation values, such as the colluders in Maze or eBay system;

· **MP4:** Peers that create Sybil peers to promote their own reputation values, and hence they can consume more resources in the system, such as the peers in eBay system with fake feedbacks from their Sybil peers;

· **MP5:** Peers that exploit P2P's resources for their malicious purposes like worm dispatching, Denial of Service etc.

We recognize that this partition is incomplete and there also exist some other malicious peers with more complex behaviors. However, as shown in [23], other malicious peers' behavior can be derived from these basic categories. Besides the reputation matrix, the topology and log files of the system can also be utilized to find out the malicious peers. However, these files are usually hard to obtain and is incomplete due to data missing and error.

3.3 Detecting Environment

The detecting environment used here is the same as those used in PeerMate [22] and SMART [23], which is derived from current RP2P systems, such as TVTorrents (www.tvtorrents.com) [25], EigenTrust [4] and Maze (http://maze.tianwang.com) [26]. In this context, the content exchange process obeys the typical P2P workload models and is divided into several time slots (rounds). During each round, each peer initiates requests and the request process follows some typical P2P workload model, such as the workload model in KaZaA and the BitTorrent. For the sake of simplicity, we adopt the typical model of the literature [27]. Moreover, each peer is assigned an initial reputation value, which will increase by X_u when it uploads a piece of valid content and decrease by X_d when downloading a valid piece, and $X_u \geq X_d$.

Reputation Matrix. Let N be the total number of peers and X_p^T be the reputation value of peer *p* at the end of the T^{th} round, $1 \leq p \leq N$. Consequently, the reputation value of all the peers can form a reputation vector $XV^T = (X_1^T, X_2^T, \ldots, X_N^T)$ at the end of the T^{th} round. Besides, from the perspective of one single peer p, X_p^t, $1 \leq t \leq T$, can form a reputation time series $XS_p = (X_p^1, X_p^2, \ldots, X_p^T)$. Then we can obtain a reputation matrix $X^{T \times N}$ as in (1) at the end of the T^{th} round. The i^{th} column of $X^{T \times N}$ is the reputation time series of peer i.

And the t^{th} row of $X^{T \times N}$ is the reputation vector at the end of round t.

$$X^{T \times N} = \begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_N^1 \\ X_1^2 & X_2^2 & \cdots & X_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ X_1^T & X_2^T & \cdots & X_N^T \end{bmatrix} \quad (1)$$

Reputation Matrix Retrieval. In centralized RP2P systems, $X^{T \times N}$ is usually collected and stored by a centralized facility, such as the tracker servers or the central server in Maze. In contrast, $X^{T \times N}$ can be collected and calculated by each peer respectively in RP2P systems with decentralized reputation management scheme. For instance, each peer can obtain a reputation of all its neighbors in iRep and EigenTrust system through iterative computing or analyzing historical transaction records [4][11]. Hence, at least in some way, we can always get

$X^{T\times N}$.For the sake of simplicity, we will use x to represent $X^{T\times N}$ in the remainder of our paper.

Topology Information. Topology indicates the connection relationship among the peers in the P2P system. Each peer establishes its connection with the help from the central facility or its neighbor peers. For instance, the tracker deployed by BitTorrent sends peer list to the peers to accelerate content delivery in the system. Then, the peer can choose their neighbors from the list. Moreover, each peer will exchange its neighbor list with its neighbors. Through the neighbor list interaction, a peer can obtain its neighbors' collaborators and thus to infer the structure of the network.

Log Files. In those P2P systems with centralized facilities, the central component of the system will record the state of the system in the log files, which may contain the neighbor relationship among peers, the transactions happened among the peers as well as the events in the system, such as node joining and departure. In those RP2P system like Maze, each peers' reputation will be stored at the central server, which is very import log records to construct the reputation matrix. However, many of the log files can only be analyzed manually and are hard to be automatically handled by the computer.

3.4 Problem Statement

After defining these malicious peers and listed the available data, the problem is now to find out those malicious peers based on the collected data, through some particular time-series analysis and graph theory methods.

4. The General Detection Framework

As illustrated in the above section, all the malicious peers are with various objectives when joining the system. Despite of this, they possess an identical feature, which also differentiates them from honest ones, i.e. they behave differently from honest peers. Since the reputation value of a peer reflects its behavior features, different behaviors will lead to different reputation values, which will afterward lead to their different reputation time-series in the reputation matrix. Therefore, we can distinguish malicious peers from honest ones if we can extract their different behavior features, which are embedded in the different deterministic features of their reputation time-series in the reputation matrix. Moreover, malicious peers will construct some community on top of the topology of the P2P systems. This is helpful for malicious peer detection. However, the precise topology of a dynamic system is hard to draw.

Based on this observation, the general framework of malicious peer detection is show in Figure 1. As shown in Figure 1, the general framework contains 4 steps to identify the malicious peer, i.e. Data Collection, Data Processing, Malicious Peers Detection and Malicious Peers Clustering.

The Data Collection step is in charge of collecting various information which can help to find out malicious peers, including reputation matrix, topology information, log file and transaction record etc. The reputation matrix can be recorded by a central facility like the central server used in Maze or be collected by the tracker in BitTorrent-like systems through distributed message scheme. Note that, the general framework also accept the topology information, log file and the transaction record to assist malicious peers identification although they are hard to obtain and are not necessary for a few methods proposed in [22][23].

The data collected in the first step will be processed by the Data Processing step. As shown in Figure 1, there are a number of methods which can be utilized by the second step, including Time Series Analysis, Principal Direction Divisive Partitioning [28] [29], Multiscale Principal Component Analysis [22], Factor Analysis [30], Non-negative Matrix Factorization [31] and Graph Theory Analysis [19]. Note that the listed methods are incomplete, other methods can also be used to achieve this goal. Moreover, various processing methods are applied to different types of data collected in the first step. For example, the Graph Theory Analysis method identifies the malicious community based on the topology information while other methods use the collected data to reconstruct the reputation matrix and to obtain the residual matrix. The process results of this step can be used in the third step to find out the malicious peers.

After data processing, the malicious peers can be distinguished from the honest ones. This can be achieved by many methods. For instance, the Shewhart control chart[24] can be applied on the residual matrix obtained in the second step to find out the malicious peers based on the assumption that the reconstruction error of malicious peers in the residual matrix are larger than those of the honest ones. For the factor analysis method, those peers which share the same factor can be treated as the same category.

As defined in Section III, the malicious peers belong to various categories. Therefore, the last step of our framework in Figure 1 is to cluster the malicious peers. The data for clustering can be drawn from the transaction record or the log file collected in the first step. For instance, the malicious peers can be clustered based on their upload entropy defined in reference [3].

According to this general framework, we can draw a number of detecting and clustering methods through various mathematical time-series analysis methods and clustering algorithms such as K-means, DBScan [32] and so on. From this point of view, the algorithms presented in [3] [22] [23] can be treated as the concrete implementation of our general framework.

5. Framework Application

5.1 Malicious Peers Detection

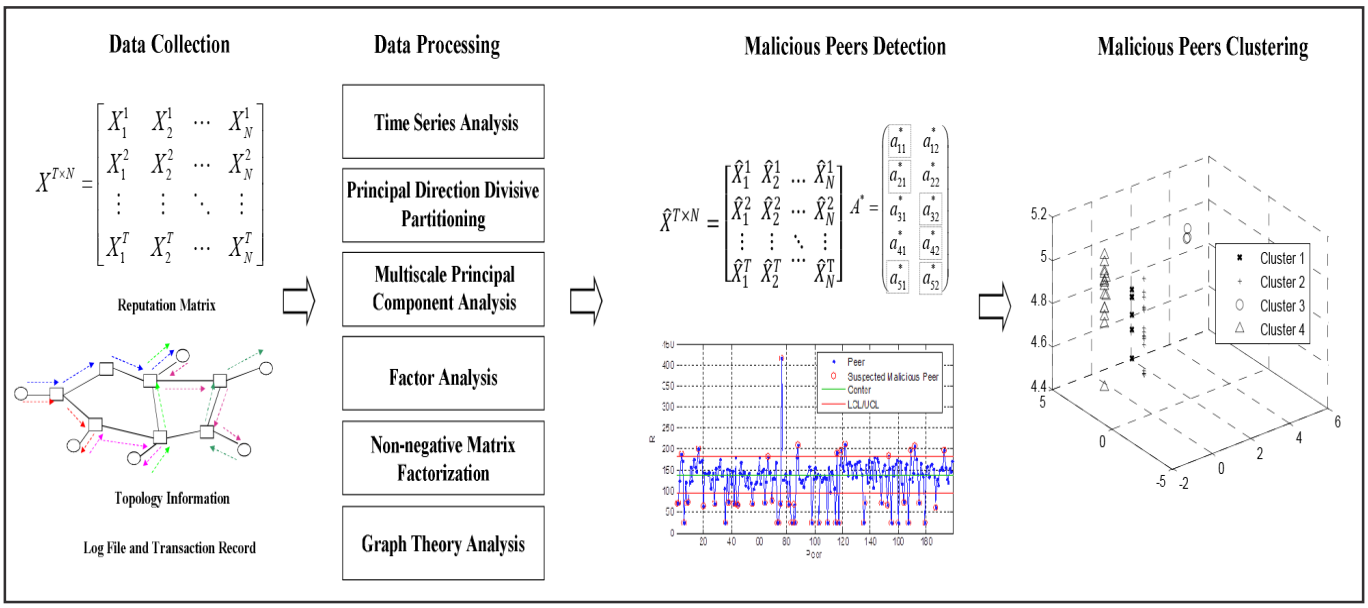

Figure 1. The General Framework of Malicious Peer Detection

Let μ and $C = (X - \mu \times e^T)(X - \mu \times e^T)^T$ be the mean and covariance matrix of $X = [X_1, X_2, ... X_N]$ respectively, where X_i is the i^{th} column of X and e is a unit vector. The Karhunen-Loeve (KL) transformation consists of projecting the columns of X onto the space spanned by the leading k eigenvectors (those corresponding to the k largest eigen values). The result is a representation of the original time-series (i.e. the columns of X) in a new k-dimensional space instead of the original T-dimensional space. Besides reducing the dimensionality, the transformation often has another beneficial effect of removing much noise presented in the data. In our case, we are interested in projecting each column onto the single leading eigenvector, which is denoted by μ, since the leading eigenvector is the direction of maximum variance. Hence it should also be the direction in which the peers tend to be the most spread out.

The projection of the R_i is $\sigma \times v_i = \mu^T \times (X_i - \mu)$, where σ is a positive constant arising from applying Singular Value Decomposition (SVD) to matrix $A = (X - \mu \times e^T)$, and the sign of the values $v_1, v_1, ..., v_N$, are used to determine the splitting for the two clusters in X, To efficiently compute $\sigma \times v_i$, literature has proposed to obtain $\sigma \times v_i$ through utilizing the SVD result of the matrix A [12].

After obtaining $\sigma \times v_i$, we can split the columns strictly according to the sign of the corresponding v_i's. All the peers R_i for which $v_i \leq 0$ are partitioned into one category, and all the peers for which $v_i > 0$ are put into the other category. The intuitive justification of the choice of splitting at the mean is in the following. If there are two well-separated clusters, then the mean would likely be between them [12]. Accordingly, all the peers are divided into two clusters since each column is a time-series of a peer. Consequently, the peers are also divided into two categories. One category is composed of honest peers and the other consists of malicious peers. Here, the category with $v_i > 0$ is considered as suspected honest peers set (SHP) and the other category with $v_i \leq 0$ is considered as suspected malicious peers set (SMP), since the first eigenvector tends to be close to the time pattern of the time-series of honest peers which account for the majority of the system.

In summary, we can partition all the peers to suspected honest peers and suspected malicious ones through principal direction divisive partitioning. The next step is to cluster malicious peers through mapping malicious peers into a three dimensional Interaction Character Entropy (ICE) space.

5.2 Interaction Character Entropy

An Interaction Character refers to a particular profile of a peer's exchange and request activities for content. The exchange activities of peer *p* refer to its content exchange activities, such as "peers that upload content to *p* (UP)" and "peers that download content from *p* (DP)". The request activities of peer p refer to its content request activities, such as "peers that request *p* for content (RDP)" and "peers that *p* requests for content (RUP)".

The Interaction Character Entropy (ICE) of a particular interaction character *ic* is computed as:

$$H(ic) = \sum_{i=1}^{G} \left(\frac{G_i}{S}\right) log_2 \left(\frac{G_i}{S}\right) \qquad (2)$$

where G is the total amount of different elements of this ic, i refers to some particular Interaction Character, and G_i is the number of element i, with $1 \leq i \leq G$, while S is the total amount of elements given by $S \Sigma G_i$. By definition, the more dispersed the elements are, the higher the $H(ic)$ is. $H(ic) = 0$ if $G = 1$, while $H(ic) = \log_2 G$ when $G_1 = G_2 = \ldots = G_N > 0$. Specially, $H(ic) = -1$ if $G = 0$. Accordingly, we can define the ICE of each interaction character (ic)

mentioned before: $H(RUP)$ for RUP, $H(UP)$ for UP and $H(DP)$ for DP.

For example, in Figure 2, the peers that upload content to peer p are $\{i, j, m\}$, and the number of pieces of content they upload to p is denoted by $\{1, 1, 2\}$. So, $G = 3$, $S = 4$, and we can compute $H(UP)$ to be 1.0397 using (2).

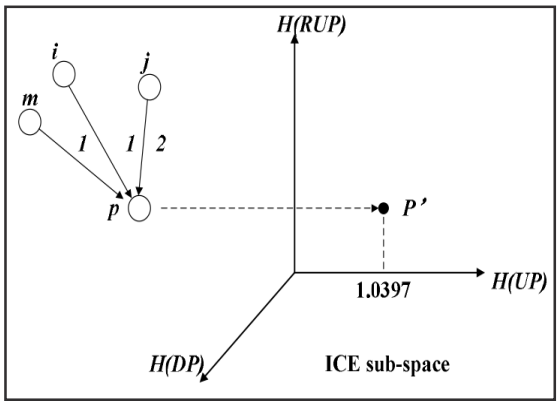

Figure 2. Content exchange of p and the projection to ICE space

5.3 ICE based Peers Clustering

Projecting Malicious Peers onto ICE Space. We choose $H(UP)$, $H(DP)$, $H(RUP)$ to construct the three-dimensional ICE space. We first collect the interaction information and compute ICE of all the peers in SMP. Then each malicious peer p can be mapped into a point in the three-dimensional ICE space with coordinates $(H(UP)_p, H(DP)_p, H(RUP)_p)$, where $H(UP)_p$ is $H(UP)$ of p, $H(DP)_p$ is $H(DP)$ of p, and $H(RUP)_p$ is $H(RUP)$ of p. As in Figure 2, p is projected to the point p' in the ICE space with $H(RUP) = 1.0397$.

Clustering in the ICE space. Different behaviors will lead to different interaction characteristics, which will then in turn bring about different ICEs. Hence peers that belong to the same category will be close to each other, while peers that belong to different categories will be far away from each other. For simplicity and flexibility, we use the k-means clustering algorithm [33] to cluster the points in the ICE space since we know the number of categories of the malicious peers.

Peer type	$H(DP)$	$H(UP)$	$H(RUP)$
Honest	positive number	positive number	positive number
MP1	-1	lowerpositive number	positive number
MP2	-1	-1	positive number
MP3	0	0	positive number
MP4	-1	0	positive number
MP5	positive number	-1	positive number

Table 1. The ICEs of the peers belong to different categories

Table 1 illustrates the different ICE of each interaction character. As Table 1 demonstrates, MP5 cannot be distinguished by principal direction divisive partitioning since it has similar behavior with the honest peers, we will cluster malicious peers by four different categories: MP1; MP2; MP3; MP4.

5.4 IDEA

IDEA. The ICE and principal direction divisive partition based malicious peers Detecting and clustering Algorithm (IDEA) is presented in Algorithm2. IDEA begins (Lines 1-11) by isolating malicious peers from honest ones using principal direction divisive partition as described in Section 5.1, calculating ICEs of malicious peers, and then store them in **H**.

The next step is to apply K-means to **H** to categorize the malicious peers into four clusters and judge their categories according to Table 1 (Lines 12-14).

Algorithm 2: IDEA

Input: $R^{T\times N}$

Output: $SMP1, SMP3, SMP4, SMP2$

1: $n = 0$

2: $\boldsymbol{A} = \boldsymbol{R}^{T\times N} - \mu \boldsymbol{e}^{\boldsymbol{T}}$

3: $[\boldsymbol{U}\,\Sigma\,\boldsymbol{V}^T] = SVD(\boldsymbol{A}), \Sigma = diag\{\sigma_1, \sigma_1, \ldots, \sigma_{min}\{m, n\}\}$

4: $\boldsymbol{u}_1{}^T\boldsymbol{A} = \sigma_1 \boldsymbol{v}_1{}^T$, $\boldsymbol{u}_1$ is the first column of $\boldsymbol{U}$, $\boldsymbol{v}_1$ T is the first column of $\boldsymbol{V}$

5: **for** $i = 1$ **to** N

6: **if** $\boldsymbol{v}_1(i) \le 0$

7: add peer i to SMP

8: $n = n + 1$

9: Compute coordinate of i in the ICE space as $(H(DP)_i, H(UP)_i, H(RUP)_i)$ and add it to $\boldsymbol{H}$

10: ***endif***

11: ***endfor***

12: $\boldsymbol{H} = \{(H(DP)_1, H(UP)_1, H(RUP)_1), (H(DP)_2, H(UP)_2, H(RUP)_2), \ldots, (H(DP)_n, H(UP)_n, H(RUP)_n)\}$

13: K-means($\boldsymbol{H}$, 4) to obtain $SMP = \{SMP\text{-}1, SMP\text{-}2, SMP\text{-}3, SMP\text{-}4\}$

14: judge clustering results according to Table 1

Figure 3. The Algorithm description of IDEA

Time Complexity of IDEA. The time complexity of IDEA is mainly introduced by the computation of the principal direction and the k-means clustering algorithm. The time complexity of computing of the principal direction is $O(TN^2)$ [34]. The time complexity of the k-means algorithm is $O(Nkl)$, where N is the number of peers, k is the number of clusters, and l is the number of iterations. Consequently, the time complexity of IDEA is $O(TN^2)$.

6. Algorithm Evaluation

It is difficult to explore all aspects of IDEA and compare it with existing algorithms using traces of real systems. We adopt instead a simulation-based approach for understanding and evaluating IDEA and comparing with existing algorithms. Such an approach provides the flexibility of carefully controlling the configuration parameters of the various detecting algorithms. This would be difficult or even impossible to achieve using live Internet measurement techniques. Thus, while certain interactions specific to a real deployment are missed, we believe the abstraction is rich enough to expose most details that are relevant to our simulation. We believe that our simulation-based approach is sufficient to demonstrate the effectiveness and utility of IDEA. In this section we first describe the simulation context, then present the simulation results, and finally analyze the results including a discussion of the utility and applicability of IDEA.

6.1 Simulation Setting

Here, we adopt the workload model in [4] as the underlying workload model of our simulations. Concretely, the workload is as follows. The contents arrive at constant rate $\lambda_O > 0$ and the popularity of them follows *Zipf* distribution. When a piece of content arrives, its popularity rank is determined by selecting randomly from the *Zipf*(1) distribution. On average, a client requests a constant number of pieces of content per round, choosing which piece of content to fetch from a *Zipf* probability distribution with parameter 1.0. To simplify our model, we assume that all of the content in the system is of equal size. Table 2 describes the parameters setup in the simulation. And the malicious peers are selected randomly from all the peers.

Symbol	Meaning	Base value
N	# of peers	200
O	# of contents	4000
λ_R	per-user request rate	2 contents / round
λ_O	content arrival rate	varies
P_M	the ratio of # of malicious peers to # of peers	varies
P_h	the honest possibility that strategic malicious peer act as honest ones	varies

Table 2. Simulation Parameters

Comparison Benchmarks. We choose two existing schemes as comparison benchmarks: EigenTrust[4], Upload Entropy (UEntropy) schemes and InteractionEntropy (IEntropy) [3]. In EigenTrust, iterative calculation is implemented to obtain each peer's global reputation value and peers with the lowest reputation values are treated as the least trustworthy peers, which therefore will be treated as malicious peers distinguished by EigenTrust scheme in our simulation. The second scheme aims at stimulating peers to share content in Private BT society. And those peers with lowest upload entropy will be considered as the least trustworthy collaborators, in other words, they are the suspicious malicious peers. Therefore, in order to guarantee the fairness of comparison, in UEntropy and IEntropy schemes, those peers that have the lowest entropy will be treated as suspected malicious peers as found by these schemes. Moreover, in order to investigate the importance of upload information, we also consider a scheme called Interaction EigenTrust (IEigenTrust), under which only information of request activities can be utilized to compute global reputation rather than the upload information as in EigenTrust. Thus we have the following four schemes to be used as comparison benchmarks: EigenTrust, IEigenTrust, UEntropy and IEntropy.

Let **MPS** (Malicious Peers Set) be the malicious peers set, **HPS** (Honest Peers Set) be the set of honest peers, and **SMP** (Suspected Malicious Peers set) be the malicious peers set found by particular scheme. Then we define two metrics TPR (True Positive Ratio) and FNR (False Negative Ratio) as follows:

TPR = |***SMP***)"***MPS***| / |***MPS***|;
FNR = |***SMP***)"***HPS***| / | ***HPS***|.

where | | represents the rank of a set, and ∩ stands for the intersection of two sets. Consequently, both TPR and FNR range from 0 to 1.

Let the number of malicious peers be N_{mali}, and let $N_{correct}$ be the number of malicious peers that have been cluster to their own categories. To evaluate the preciseness of malicious peers clustering, we define the clustering correctness ratio as $N_{correct}/N_{mali}$.

6.2 Evaluation Results

Malicious Peer Detecting. Here we first compare malicious peers detecting results of proposed IDEA with benchmarks EigenTrust, IEigenTrust, UEntropy and IEntropy. Here, we choose $\lambda_O = 2$, $P_h = 0$ and $P_M = 0.2$. Other parameter values are listed in Table 2.

The simulation results of the TPRs and FNRs of all the schemes at the end of the 100th round are shown in Figure4 and Figure 5 respectively. As Figure4 demonstrates, the TPR of IDEA is 95% and it is the highest among the five schemes. The TPR of EigenTrust is 92.5% which is the second highest, TPR of UEntropy is about 57.5% at the end of the 25th round, which ranks third of the five, TPRs of IEntropy and IEigenTrust are both unacceptably lower than 5% after the 60th round. In contrast, from Figure 5 we can see that the FNR of IEntropy reaches 100% at the 20th round and is the worst among the five schemes. The FNR of IEigenTrust increases from 80% to 97.5% as

simulation rounds increase and is the second worst, FNR of UEntropy decreases from 57.5% to 42.5% as simulation rounds increase, while FNR of EigenTrust remains stable at 7.5% after the 5th round. The FNR of IDEA decreases from 37.5% to 0.63% as simulation rounds increase, and this is the best among the five schemes.

Malicious Peer Detecting with Missing Data. IDEA may suffer from Missing Data due to network congestion and churn in the P2P system. Here we investigate how IDEA can adapt to a Missing Data context with different ratios of missing data between 0 and 50%, while the missing data are selected randomly from X. Since IDEA needs the full set of X, we fix X as follows: if X_i^t is missing, then we set $X_i^t = (X_i^{t-1} + X_i^{t+1})/2$, if $1<t<T$; $X_i^t = X_i^{t+1}$, if $t = 1$; $X_i^t = X_i^{t-1}$, if $t = T$. After the 100th round, we have found that the TPR of IDEA is higher than 90% with FNR staying at 0.63% when up to 50% of the elements are missing. This means the performance of IDEA is acceptable when less than 50% of the elements are missing. The main reason lies on the fact that missing data cannot change the essential differences between malicious and honest ones.

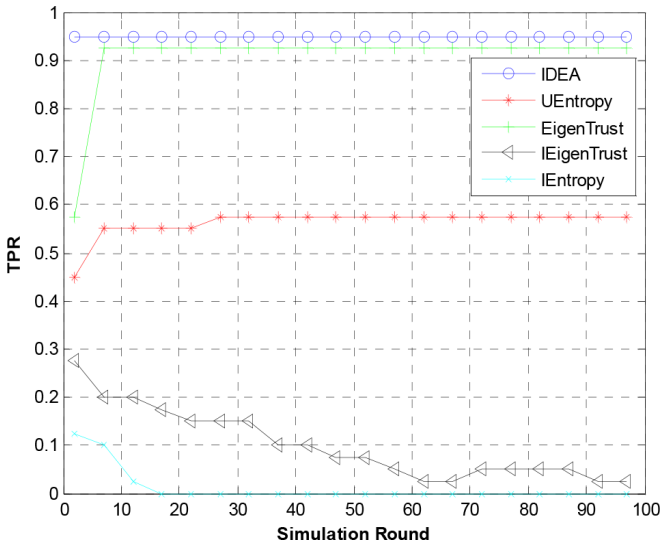

Figure 4. The comparison results of TPRs

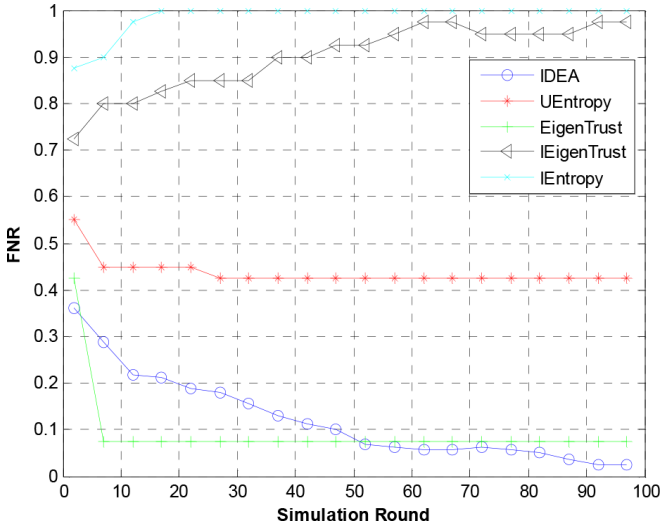

Figure 5. The comparison results of FNRs

Malicious Peers Clustering. IDEA clusters malicious peers in the ICE space using the *K*-means clustering algorithm. The clustering result of malicious peers is shown in Figure 6. In Figure 6, the malicious peers are clustered into four classes: MP3, MP4, MP1, MP2. Each class has different ICEs in the ICE space. The clustering correctness ratio of IDEA is 95%.

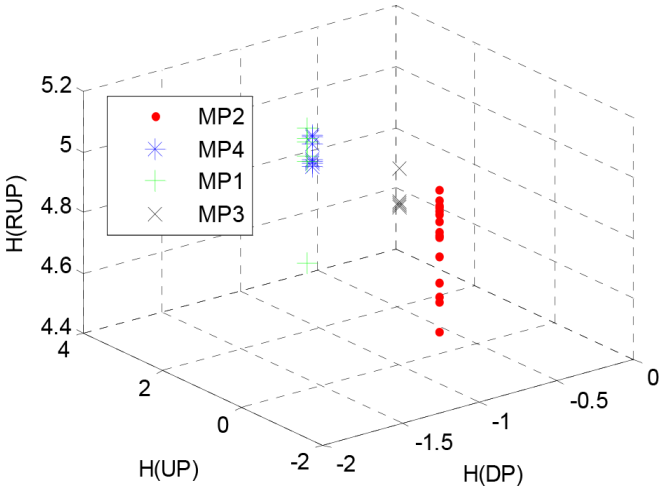

Figure 6. Clustering result of IDEA

7. Conclusion and Future Work

Finding out and clustering the malicious peers in Reputation-based P2P (RP2P) systems is vital to ensure the effectiveness of the reputation system.

This paper presents a general framework for detecting malicious peers in RP2P systems, which mainly contains four steps, i.e. Data Collection, Data Processing, Malicious Peers Detection and Malicious Peers Clustering. Moreover, the algorithm of this framework is shown. To our best knowledge, this work is the first effort towards a general framework for malicious peer detection for Reputation-based P2P systems. Moreover, a detecting algorithm based on PDDP is proposed within this framework. Simulation results have validated the efficiency of the detecting algorithm as well as the proposed framework.

In the future, we will generalize this framework to make it can be used to identify malicious peers in cloud computing[35] environment.

Acknowledgment

This research was supported in part by the Major State Basic Research Development Program of China (973 Program) No. 2012CB315806, National Natural Science Foundation of China under Grant No. 61402521 No. 61070173, National Natural Science Foundation of China under Grant No. 61201216, Jiangsu Province Natural Science Foundation of China under Grant No. BK20140068, Foundation of National Key Laboratory of Science and Technology on Communications.

References

[1] Labovitz,C., Johnson,S.I., McPherson,D., Oberheide, J., Jahanian, F. (2010). Internet inter-domain traffic. *In*: Proceedings of the ACM SIGCOMM 2010 conference (SIGCOMM '10). ACM, New York, NY, USA, 75-86.

[2] Li, J., Aurelius, A., Nordell, V., DuM., Arvidsson, A., Kihl, M. (2012). A five year perspective of traffic pattern evolution in a residential broadband access network, Future Network & Mobile Summit (FutureNetw), p.1,9, 4-6.

[3] Liu, Z., Dhungel, P., Wu, D., Zhang, C., Ross, K. W. (2010). Understanding and Improving Incentives in Private P2P Communities. In ICDCS 2010, Italy.

[4] Kamvar, S. D., Schlosser, M. T., Garcia-Molina, H. (2003). The EigenTrust Algorithm for Reputation Management in P2P Networks. In Proceedings of the Twelfth International World Wide Web Conference, Budapest, May.

[5] Lian, Q., Zhang, Z., Yang, M., Zhao, B. Y., Dai, Y., Li, X. (2007). An empirical study of collusion behavior in the maze P2P file-sharing system. *In: IEEE ICDCS*, June 2007.

[6] Liu, Y., Yang, Y., Sun, Y. L. (2008). Detection of collusion behaviors in online reputation systems, 42nd Asilomar Conference on Signals, *Systems and Computers*, 1368,1372, 26-29 Oct. 2008.

[7] Elhalabi, M. J., Manickam, S., Melhim, L. B., Anbar,M. Alhalabi, H. (2014). A Review of Peer-to-Peer botnet Detection Techniques. *Journal of Computer Science* 10 (1) 169-177.

[8] Douceur, J. R. (2002). The Sybil attack. In First International Workshop on Peer-to-Peer Systems (IPTPS '02), March.

[9] Despotovic, Z. (2010). P2P Reputation Management Through Social Networking, *Handbook of Peer-to-Peer Networking* (2010**)**, Part 6, 733-760.

[10] Adar, E., Huberman, B. (2000). Free riding on gnutella, First Monday 5, October.

[11] Wei, X., Chen, M., Tang, C., Bai, H., Zhang, G., Wang, Z. (2013). iRep: Indirect Reciprocity Reputation based Efficient Content Delivery in BT-like Systems. Telecommunication Systems, 54 (1) 47-60.

[12] Mekouar,L., Iraqi,Y., Boutaba,R. (2006). Peer-to-Peer's Most Wanted: Malicious Peers. Comp. Net., 50 (4) March. 545–62.

[13] Ji, W., Yang, S., Chen, B. (2008). A Group-Based Trust Metric for P2P Networks: Protection against Sybil Attack and Collusion. International Conference on Computer Science and Software Engineering, 2008 3, 90 - 93.

[14] Despotovic, Z., Aberer, K. (2006). P2P reputation management: Probabilistic estimation vs. social networks, Computer Networks 50 (2006) 485–500.

[15] Tehale, A., Sadafule, A., Shirsat, S., Jadhav, R., Umbarje, S., Shingade, S. (2012). Parental Control algorithm for Sybil detection in distributed P2P networks. *International Journal of Scientific and Research Publications*, 2 (5), May.

[16] Selvaraj, C., Anand, S. (2012). A survey on Security Issues of Reputation Management Systems for Peer-to-Peer Networks. *Computer Science Review*, 2012, 6(4):145-160.

[17] Jin, X., Chan, S.-H. G. (2010). Detecting malicious nodes in peer-to-peer streaming by peer-based monitoring. ACM Trans. *Multimedia Comput. Commun. Appl.*, 6 (2), 9:1–9:18.

[18] Koutrouli, E., Tsalgatidou, A. (2012). Taxonomy of attacks and defense mechanisms in P2P reputation systems—Lessons for reputation system designers. *Computer Science Review*, 2012, 6 (2) 47-70.

[19] Lee, H., Kim, J., Shin, K. (2010). Simplified clique detection for collusion-resistant reputation management scheme in P2P networks. In 2010 *International Symposium on Communications and Information Technologies (ISCIT)*, 273 - 278.

[20] Ciccarelli, G., Cigno, R. L. (2011). Collusion in peer-to-peer systems, *Computer Networks*, Comp. Net., 55, (15), October, 3517–3532.

[21] Liu,Y., Li, Y., Xiong, N., Park, J. H., Lee,Y. S. (2010). The incentive secure mechanism based on quality of service in P2P network, Computers and Mathematics with Applications, 60 (2010) 224-233.

[22] Wei, X., Ahmed, T., Chen, M., Pathan, A. -S. K. (2012), PeerMate: A malicious peer detection algorithm for P2P Systems based on MSPCA, *In*: Proceedings IEEE Int. Conf. on Computing, Networking and Communications (ICNC), Lahaina, HI, USA, January 2012, 815-819.

[23] Wei, X., Fan, J., Chen, M., Ahmed, T., Pathan, A.-S.K. (2013). SMART: A Subspace based Malicious Peers Detection algorithm for P2P Systems. *International Journal of Communication Networks and Information Security*, 5, (1), April 2013, 1-9.

[24] Montgomery,D. C. (1991).Introduction to Statistical Quality Control*, Second Edition*, New York: John Wiley & Sons, Inc. 1991.

[25] Meulpolder, M., Pouwelse, J. A., Epema, D. H. J., andSips, H. J. (2009). BarterCast: A practical approach to prevent lazy freeriding in P2P networks. *In*: Proceedings IEEE *International Symposium on Parallel & Distributed Processing* (IPDPS'09), 2009 1-8.

[26] Jiang, H., Jin, H., Guo, S., Liao, X. (2012). A measurement-based study on user management in private BitTorrent communities. Concurrency and Computation: Practice and Experience.

[27] Gummadi, K., Dunn, R., Saroiu, S., Gribble, S., Levy, H., Zahorjan, J. (2003). Measurement, modeling,

and analysis of a peer-to-peer file-sharing workload. In 19th ACM Symposium on Operating Systems Principles, Bolton Landing, NY, USA, October 2003.

[28] Boley, D. L. (1998). Principal direction divisive partitioning. Data Mining and Knowledge Discovery , 2(4):325-344, 1998.

[29] Tasoulis, S. K., Tasoulis, D. K. Plagianakos, V. P. (2010). Enhancing principal direction divisive clustering, Pattern Recognition, (2010) 43:3391-3411.

[30] Rencher, A. C. (1998). Multivariate Statistical Inference and Applications. Wiley, 1998.

[31] Xu, W., Liu, X.,Gong,Y. (2003). Document clustering based on non-negative matrix factorization, ACM SIGIR, Toronto, Canada, 2003.

[32] Ester, M., Kriegel, H.-P., Sander, J., Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, in: Proc. the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96), 1996.

[33] K-Means Clustering, http://en.wikipedia.org/wiki/K-means_clustering.

[34] Golub, G., Loan, C. F. V. (1996). Matrix Computations, The Johns Hopkins University Press.

[35] Liu, Z., Chen, X., Yang, J., Jia, C.,You, I. (2014). New Order Preserving Encryption Model for Outsourced Databases in Cloud Environments. Journal of Network and Computer Applications, Elsevier, 2014. DOI: 10.1016/j.jnca.2014.07.001.

Author biographies

Xianglin Wei was born in Anhui province, China. He received a Bachelor's degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China in 2007, and his Ph. D. degree in PLA University of Science and Technology, Nanjing, China. He is now working as a researcher at the PLA University of Science and Technology, Nanjing, China. His research interests include Cloud Computing, Peer-to-Peer network, network anomaly detection, network measurement, and distributed system design and optimization. He has served as editorial member of many international journals and TPC member of international conferences.

Jianhua Fan was born in Anhui province, China. He received the Ph. D. degree from Institute of Communication Engineering, Nanjing, China in 1999. Now he is working as a senior researcher at PLA University of Science and Technology, Nanjing, China. His research interests include wireless ad hoc network, cognitive network and delay/disruption tolerant network.

Tongxiang Wang was born in Shangdong province, China. He received a Bachelor's degree from the PLA University of Science and Technology, Nanjing, China in 2012. He is now a Ph. D. candiate in PLA University of Science and Technology.

Qiping Wang was born in Jiangxi province, China. She is now a postgraduate in PLA University of Science and Technology.