Fake Servers in EDonkey Networks*

ZHANG Min
School of Electronics and Information Engineering
Beijing Jiaotong University
Beijing,100044,China
mia.minzhang@gmail.com

CHEN Changjia
School of Electronics and Information Engineering
Beijing Jiaotong University
Beijing,100044,China
changjiachen@sina.com

JIA Jinkang
School of Electronics and Information Engineering
Beijing Jiaotong University
Beijing,100044,China
jinkangjia@hotmail.com

ABSTRACT

With the growth of the popularity of various P2P file-sharing systems, the tussle between ordinary P2P users and copyright protectors becomes more and more fierce. The contests of servers (supernodes), the key components of these systems, have turned to be the focus of the combat. Some copyright protectors or pollution companies have established their own servers so as to control more users and pollute the whole system, while the users also take measures to identify these fake servers and prevent files from being polluted. To our knowledge, we are the first to study fake servers in eDonkey networks.

We developed a dedicated crawler and traced eDonkey users for over 24 days. Based on our measurements, we find that fake servers, which account for more than 59.4% in number although, don't work well in attracting ordinary users. The users, who have once connected to one of these fake servers, take only 30.9% of all traced users. Even worse, the low stay time ratio of users on fake server shows that fake servers lack mechanisms or incentives to keep users stay longer. However, we cannot underestimate the potential influences of fake servers yet. From our analysis, fake servers indeed disturb users' stay distribution at the rank of normal server. Taking the popularity distribution of servers (Zipf-like) into account, we suggest that copyright protectors should try to control several popular servers instead of setting up many unpopular ones. Furthermore, the probability of potential contacts between good and bad users is high enough (94%). For copyright protectors, utilizing these frequent contacts to spread viruses or polluted files will bring great trouble or even disasters for eDonkey networks.

In addition, the existing method (blacklisting) cannot keep up with the dynamic updates of fake servers, so we proposed an online heuristic feature-based detection method. We think it can be adopted by client software developers for real-time fake server detection.

*This work was supported in part by the China NFSC 60672069, China 973 2007CB307101, Chinese Ministry of Education under grant 20050004033 and Beijing Jiaotong University under grant 2005SM006.

Qshine08 July 28-31, 2008, Hong Kong, China.
 ISBN 978-963-9799-26-4
DOI 10.4108/ICST.QSHINE2008.3836

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Measurement, Experimentation

Keywords

P2P, fake server, eDonkey, pollution

1. INTRODUCTION

Peer-to-Peer (P2P) networks are becoming more and more popular across the Internet in last few years and the flows of P2P traffic have accounted for more than 80% of the total traffic recently. While users are enjoying the easy and fast way to get free files they wanted at home, music and movie industries have taken measures silently to save their potential incomes. The fight on free sharing between widespread network users and copyright protectors has lasted for nearly a decade on these P2P networks. In case of copyright infringement, some copyright protectors like RIAA and MPAA[1], have tried their best to prevent the illegal diffusion of copyrighted products, such as suiting P2P corporation (Napster), spreading fake or polluted files, launching poisoning and viruses attacks etc. Among them file pollution is the widely used scheme to protect the copyrights of video/audio products. In the case, copyright protectors (and/or pollution companies hired) deliberately inject polluted (or fake) files into P2P networks. Consequently, they hope that the polluted files will be propagated across the whole network by the direct file exchanges among users. Up to now, most present studies on this issue put their emphasis on the measurements and modeling of file pollution [2-5].

In fact, it's the infrastructures of these systems that determine the adoption of file pollution scheme instead of other methods. Nowadays the most important and widely-used P2P networks for vast data sharing, such as BitTorrent,

[1]RIAA (Recording Industry Association of America) is the trade group that represents the U.S. recording industry. MPAA (Motion Picture Association of America) are the advocate of the American motion picture, home video and television industries. Both of them commit themselves to protecting intellectual property rights worldwide.

eDonkey (eMule) and PPLive, are all hybrid structured networks coordinated by a small number of trackers. For they aren't fully decentralized P2P systems, copyright protectors can build their own servers to mislead naive users into traps and/or to collect users' important information simultaneously for later suitable legal actions. In this paper we name these servers as fake (or spy) servers. However, compared with the file pollution, which has been discussed in many papers, the problem of fake server has not been well studied to the best of our knowledge. We will pay our attention on these servers and report our measurement-based study on them in the eDonkey networks in the paper.

As we know, the main purposes of fake servers are collecting IP addresses as suiting copyright infringement testimony, polluting files and misleading users. As they aren't genuine servers which try to serve for other peers, these fake ones will behave in different ways. Once one connects to a fake server, some nasty consequences follow. Firstly, the fake servers will record this user's shared files, IP address, and its activity as suiting testimonies. Then more fake servers will be added into the user's server list. Thus the chance of reconnecting to fake servers increases as the user joins the system later. Even worse, the search results returned from the fake server are all non-existent files, or corrupted fake files. As a result, the ordinary user is turned to be the poisoning source unintentionally if it downloads some files. Therefore, it's important to evaluate the effects brought by fake servers. But so far the estimates are mostly based on individual cases or experiences posted in forums and websites. We wish to put forward the research into a more systematic and panoramic scope. A measurement platform is designed and implemented for this goal. Our platform is composed of three components, which will be discussed in detail in section 3. Through analyzing the collected datasets for 24 days, some conclusions are reached. We list the main contributions of the paper as follows:

1. Fake servers dominate in the quantity of system servers. More than 59.4% servers crawled are fake ones, and they have been listed and posted in blacklist on some forums. Besides, adopting our online FSD component, we also observed that 22.1% servers are potential fake ones, which try to hide their identities by changing their IP addresses frequently. They are hard to be detected by ordinary users and recorded in blacklist.

2. Despite their advantages in quantity, fake servers cannot catch primary or expert users. A little to our surprise, the ratio of users who have ever connected to fake servers is just 30.9%, and these users cheated tend to self-rescue themselves by reconnect to other normal servers at last. That is, the fake servers don't work well in destroying and polluting the eDonkey systems.

3. It's revealed that fake servers have obvious characteristics of geographical clustering. Based on our measurement, more than 73.9% fake servers are located in North America. This observation that most fake servers are operated in regions where laws for copyright protections are rigid and integrated, make us believe that most fake servers are manipulated by specific institutions or companies instead of individuals. That is, the establishments of fake servers are mass behaviors instead of personal deed.

4. Users' behaviors on server selection indeed are influenced by fake servers. And the probability of potential contacts between good and bad users is high. These contacts will increase the chances of potential pollution diffusion, so we cannot underestimate the potential troubles brought by fake servers.

5. Last but not the least, based on mining on the features of fake servers, in the paper we propose FSD, a real-time server detecting method. We think the method can be imported as a check module into the client for fake server detection.

The rest of this paper is structured as follows. In section 2, we will introduce some related work on P2P measurements and our motivation for the work. In section 3, the details of our crawler and the collected trace from eDonkey networks will be discussed. Based on our measurements, we will estimate the popularity and the influences brought by fake servers in Section 4. In addition, the behaviors of clients that connect to fake servers are analyzed and some suggestions to users and protectors are raised. Finally, in Section 5 we will draw some conclusions and future work are given.

2. RELATED WORK

With the inborn nature of free sharing of various P2P networks, the combat between the copyright protectors and free users of these networks becomes more and more fierce. As the tussle is still going on in commercial and industry fields, some measurement-based or theoretic works have appeared in academic field. These studies pay great attention on the pollution or anti-pollution of various P2P networks. As far as we know, [3] is the first work which tries to reveal the extent of pollution diffusion in FastTrack networks. It's observed that the pollution is widely spread and is especially pervasive for recent popular songs. In addition, the pollution of these pop songs are much more serious than that of the old classic ones, based on which the authors guess that there are some organizations or companies who try to spread fake files deliberately for protecting the copyright of these new published files. The following studies [2, 4, 5] have done much in modeling of the pollution propagation through theoretic analysis. In [4] the authors modified the well-known model for disease spread. Based on the augmented model, they have also proposed and discussed some counter-measures for the diffusion of polluted files. In [5], some non-linear differential equations are presented and the authors developed a suite of fluid models that reveal the process of pollution proliferation in P2P systems.

As for the eDonkey network , there are many measurement-based work which tries to characterize the system from different points of view. In [6], a 12-day dataset is analyzed and the traffic profiles of eDonkey network are presented. Based on the logs of eDonkey server, [7] presents some staticstial characteristics on clients' behaviors, such as request frequency, relationship among clients, and so on. [8] takes an active-probing measurement method and the characteristics of servers, clients and files are given comprehensively on eDonkey network. Unfortunately, there is little work which tries to evaluate the functionalities and threats faced by the key component of these partially decentralized systems - servers (or super nodes). Even in [8], which have mentioned servers in P2P systems, doesn't differentiate the fake servers

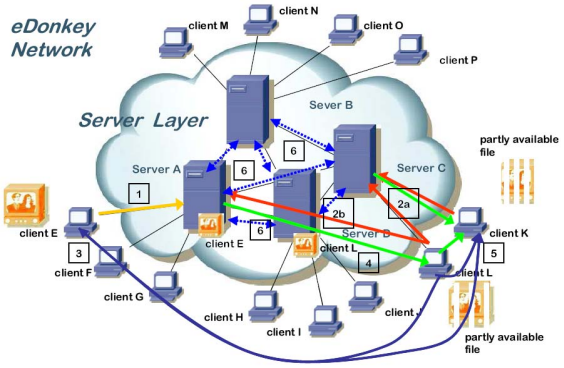

Figure 1: eDonkey Network Architecture

from the normal ones. Their conclusions are some simple statistics, which reflect users' behaviors and preferences to some extent. Our work is the first experimental study, which focuses on fake servers in eDonkey systems. By tracing the long-time switching behaviors of ordinary users, we identified the fake servers and evaluated the effects of different strategies which have been adopted by fake server operators as the counter-measure by copyright protectors.

Generally speaking, there are two kinds of measurement methods which are adopted by most works: the passive monitoring and active crawling. In [7], the authors retrieved datasets directly from the server's database. In these log files, some simple activities of ordinary peers are recorded. The disadvantages of these datasets are obvious. Because these records are mainly used for the maintenance of the system and retracing of vicious users, it's not enough for researchers to study the behavior of each peer deeply. Besides, most server operators are unwilling to open the data. The results based on only one or several servers aren't representative. Because the protocols of eDonkey are opened, the researchers can develop their own clients for data collection. In [8], the active probing of other peers may give a comprehensive view compared with past works. To study the mass behavior of clients as well as servers comprehensively, the crawler which can meet the requirements of our need are developed.

3. MEASUREMENT PLATFORM

3.1 Measurement background

EDonkey is a very popular P2P network for large file replication(mostly movies, pocketed music or softwares, hundreds to thousands of mega-bytes in size in general). For the convenience of peer management and searching, it adopts the two-level hierarchical structure (client and server) which is shown in Fig.1 [6]. One client(user) should firstly join in the eDonkey network by connecting and getting registered to one of the servers. The server is picked from the server list which is recorded in the client's configuration file. The server list can be updated or deleted from time to time automatically or manually. In this paper, we will name the connected server as current server (CS) of this client. After the client successfully connects to the server, the information of clients including its shared files are reported to server. Simultaneously, the client can launch keyword-based search queries to servers to find the files it want to get. The server may respond with the list of files which contains the queried keyword and their owners as well. Then the user can choose the file which he is interested in and begin downloading from the file owners directly. It should be noted that one server may exchange information with others if there is no answer for the queried keyword in its buffer. That is, One server can forward and respond clients' search queries instead of the CS of the client. Compared with the full decentralized system, this search method is much more efficient. In addition, the broadcast storm can be partially avoided by the adoption of two-layer structure. There are hundreds of servers in eDonkey network. Therefore, in such two level structured system, steady running of the servers is of vital importance to the whole system. The sudden shutting down of most servers in September 2007 seriously affected eDonkey network, even triggered the discussion about whether edonkey network could survive.

The fake server problem is noticed for the first time when we are doing measurement-based studies on the file distribution of eDonkey in last year. We found there are some servers in the network which always provided the clients with wrong MD4 checksums. We checked the recent published papers but no relevant explanations for the phenomenon are found. To our surprise, we found that the problem of fake servers in eDonkey network is a rather hot topic discussed at some P2P forums as well as some authoritative websites [11,12]. These discussions can be summarized as:

1. There do exist some organizations, which launch fake servers deliberately in eDonkey network, for example RIAA and MPAA.

2. As a consequence, many eDonkey users, even users in universities, have been caught and monitored by fake servers and received summonses.

3. As a counter measure, a blacklist of fake servers is posted by other communities to protect the eDonkey users from being caught and polluted.

Having these in mind, we plan to study fake servers in these P2P systems comprehensively and thoroughly. After designing a feasible measurement method, we developed our crawler and collected datasets from abundant users. After careful analysis, some insights and useful conclusions are reached.

3.2 Measurement Methodology

To estimate the effects brought by fake servers accurately, large amounts of long time records of users' CSes are needed. Thus the two key issues of our measurements are how to trace and record the servers' long time behaviors for mass users and how to determine the nature of server (fake or normal). To solve these two problems, in the paper we devised our measurement procedures and components to capture the datasets needed, which are described in detail next:

1. User Collection and Sample Selection component ($UCSS$): for the source code of eDonkey is open, it's easy for us to modify the related component to record the information needed. Using our crawler, we send some wildcard queries to CS to collect as much as possible potential eDonkey users' IP addresses. Then we

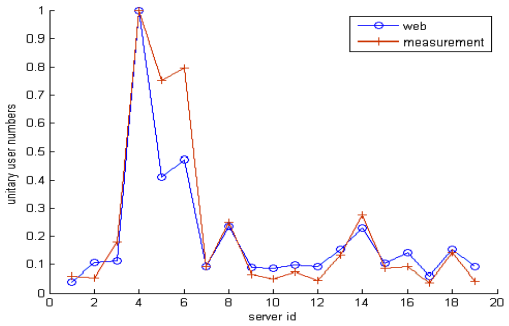

Figure 2: Compare the user numbers proportion of the top 19 servers between web Stat. and our trace

Table 1: Server Proportion

Number of total server	271
Number of fake server	161(59.4%)
Number of potential fake server	60(22.1%)
Number of normal server	50(18.5%)

trace these users one by one and choose a suitable size of reachable sample set for long time tracing.

2. Sample Trace component (ST): After choosing the clients set for tracing, we periodically send queries to each host directly according to the public protocol of eDonkey and records their CS in the database.

3. Fake Server Detection component (FSD): After getting the CS set, we go on to test if each recorded CS is a fake server or not. To recognize the fake servers more accurately, some tricks and heuristics are adopted. After we did some experiments on those fake servers which have been listed in blacklist, two obvious features of fake servers are revealed.

   (a) Because the file hashes (MD4) is the sole identity for each file being shared in eDonkey systems, we can try to query for the files whose MD4 value are already known. Then we can compare the responded MD4 value with known value. It's revealed that fake servers always return wrong MD4 value, which is one of the measures adopted by copyright protectors.

   (b) Comparing with normal servers, the fake servers will return much fewer results when encountering the queries with wildcard characters.

   Based on these two features, we developed the real-time fake server detection component FSD. Then fake servers can be identified if both features can be met in these servers.

3.3 Measurement Setup

As mentioned in last subsection, we firstly modified one of the eDonkey clients (eMule) to collect users' IPs. Different from [9], we launch queries by searching files instead of by nicknames. After recording the results responding from these CSs, we collect more than 100,000 unique IP addresses. Because of some well-known problems, such as NAT, DHCP, firewall or random port selection, not all recorded IP addresses can be reached for a long time. We monitored these users one by one and only 22,885 reachable ones are left for crawling in next step. Secondly, as a trial, we picked up 1757 users randomly for long time cyclical tracing. For each cycle, we will initiate a TCP connection request to each user. Once the TCP connection is established, we will then send *hello* message [10] to users according to open eDonkey protocols. All responses from these users are captured and saved into files for future use. It should be noted that we choose 30 minutes as our crawling interval based on our pre-tracing analysis. We find the average online time of users is much more than 30 minutes, so our crawler can connect to most users with high probability and record the status of their CSes consequently. We ran our crawler for more than 24 days (from July 1, 2007 to July 24, 2007) and gathered 391232 pieces of valid records. Finally we adopt the FSD to judge the nature of the servers. The union of our results and the fake servers which have been posted by forums and websites, composes the fake server base set for our analysis.

All above crawlings and tests are run on a 1.8 GHz PC with 640 MB RAM and a 100 Mb/s NIC.

4. BEHAVIOR ANALYSIS OF USERS AND FAKE SERVERS

To verify the validity and comprehensiveness of our datasets, we made a simple comparison based on the datasets from popular eMule websites [11, 12] and from our crawler. We choose 19 servers which are found and listed in both datasets, and calculate the occupied proportion of the number of users among different servers. The results are shown in Fig.2. It's obvious that both datasets match very well to each other, which confirms the availability of our datasets.

In this section, we'll firstly present our analysis results from servers' view. After the successful identification of fake servers, we will go on to study the influence of these fake servers. Furthermore, the popularity(rank) of fake servers and their geographical characteristics are discussed. Secondly, from ordinary users' perspective, users selection behavior are revealed based on our measurements, and the probability of potential encounters between good users and bad ones are shown. Finally, we'll raise some suggestions to both ordinary users and copyright protectors.

4.1 Characteristics of Servers

What are the features of servers in different categories? Do the fake servers work well? These questions will be answered in this subsection from our analysis.

4.1.1 Classifying servers

In total there are 271 servers which have been captured by our crawler. And the results are listed in table 1. Among them 60 servers cannot be reached by our PC crawler. We name them as potential fake servers because most normal servers will be online for most of the time and are easy to be connected. Only the fake servers will change their IP addresses frequently and go offline at any time. As for the other reachable servers, we adopt the two heuristics, which have been mentioned above to identify the fake ones. With the aid of the blacklist published in forums, we can confirm the nature of the servers which are captured in our trace. It's

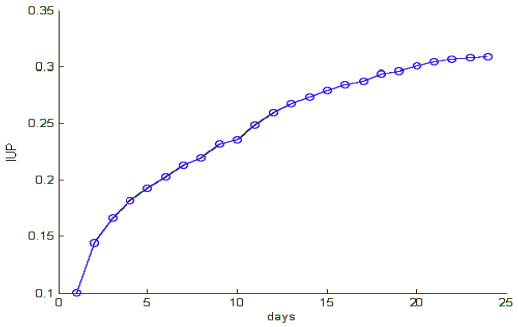

Figure 3: The evolution of IUP(influenced user percentage) in 24 days

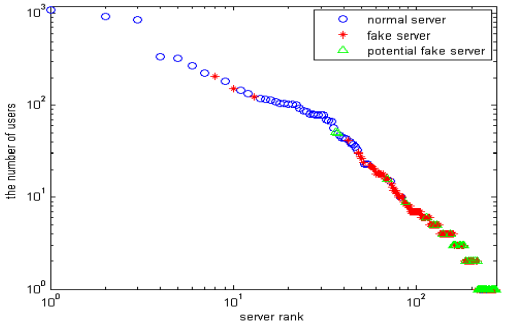

Figure 4: The rank distribution of servers during our observation

revealed that nearly 60% are fake servers, which surprise us very much.

4.1.2 *Effect of fake server*

After identifying fake servers, we go on to study the ratio of users who have been affected by these servers. We think IUP(influenced user percentage), which is defined as the ratio of users who have ever connected to any fake servers, should be a good metric to evaluate the pollution effect of fake servers. In Fig.3 it's plotted the IUP evolution curve with time. It's obvious that the IUP increases at low speed and becomes saturated around 0.30. It means that of all users we collected, only about 30.9% ones have logged on at least one of the fake servers during 24 days. Considering the large portion of fake servers, which is shown in Table 1, we think fake servers don't work well to protect the copyrights of files. The reasons will be explained and proved in detail in next subsection.

4.1.3 *Rank of servers*

We sorted all servers by the number of unique users who have logged on the server at least once. One user who has logged on different servers will be calculated once in each server's statistics. The rank results are shown in Fig.4. One line in the log-log coordinates system indicates that the number of users for each server follows the power-law distribution. The obvious three colored regions denote different categories of servers. From the graph, it's revealed that the normal servers, which have been connected by most users collected in the trace, are more popular than fake servers (red asterisks) and potential fake servers (green triangles). Though there are few fake servers (in top 15) which rank before most normal servers, the users logged on them will not stay for long time (see more in 4.2.2). As for the potential fake servers, the relatively few users and low reputation confirm us their unstable and dynamic characteristics.

4.1.4 *Geographical property of fake servers*

Adopting GeoIP[13], we can easily retrieve the locations by sending queries which contain the IP addresses of these servers. In table 2, it's shown that about 73.9% fake servers are from North America (the location of RIAA and MPAA).

Based on the analysis presented in this subsection, we conclude that fake servers, which are mainly located in North America, failed to connect to and pollute most users, though they account for nearly 60% or more in quantity. In addition, the existing static fake server blacklist is useless for capturing most dynamic fake servers. We recommend that some feature-based real-time heuristics, such as FSD, can be adopted to detect these unstable servers.

Table 2: Geographical Property Of Fake Servers

Geographical region	Number of fake servers
North America	119(73.9%)
Europe	41(25.5%)
Asia	1(0.6%)

4.2 User behaviors on Fake and Normal Servers

In this subsection, we try to answer these questions. How do users select servers? How long do users stay on different categories of servers? Do fake servers affect users' behaviors? For the convenience of analysis, we partition all users into two sets: (1)Bad users who have connected to fake servers. (2)Good users who haven't connected to any fake servers.

4.2.1 *User behaviors on server selection*

In order to display the relationship between users and servers comprehensively, Fig.5 is plotted. In Fig.5(c) the dynamic connection behaviors of all users are plotted. As shown, the two obvious vertical lines at about 50 and 200 divide all servers into three categories: normal servers, fake (listed) servers and unreachable servers from left to right. Similarly, the two fields which are separated by a horizontal line denote two different categories of users: those whose ID below 600 are bad users while others are good ones. It's indicated that 1) users always have strong preference on server selection no matter which category of servers is considered. 2) As for the normal servers, both bad and good users have similar preferences to some well-known servers. Fig.5(a) is the compression mapping and generalizability of Fig.5(c) and similar conclusions can be drawn. We infer that the selection preferences are the consequences of the design of eDonkey client software. The server which has been connected successfully last time will be remembered and have higher priority to be connected this time.

Additionally, it's obvious that even bad users will still tend to connect to normal servers than fake ones though fake servers account for much higher in number, which makes the pollution effect further discounted.

4.2.2 *Users stay property on servers*

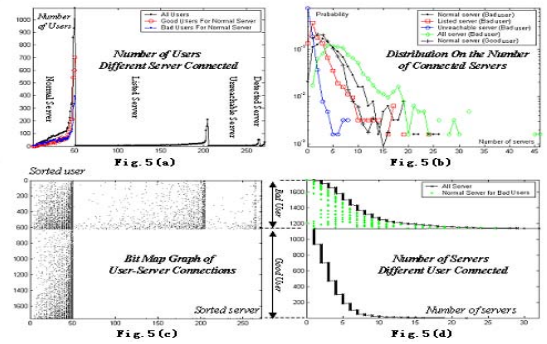

Figure 5: The user-server connections analyzed graphs

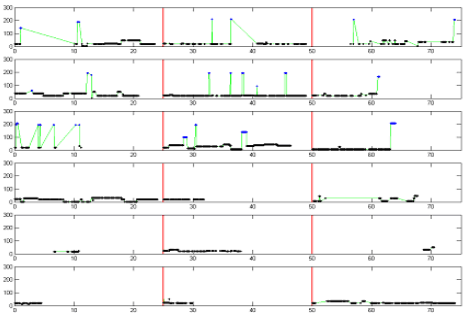

Figure 6: Random selected user traces

In this subsection, we will analyze users' dynamic behaviors in time domain. Fig.6 shows some random selected users' traces, and the x-axis represents the time and y-axis denotes servers' ID. The servers whose IDs are less than 50 are normal servers. From the frequent hops of green lines in top 3 subplots, it can be seen that, even for bad users, they will still stay on normal servers in most of its online time. Fake servers can only disrupt these stay curves but cannot make users attached to them for a long time. The ratio the bad users' stay time on fake servers are calculated, and the distribution is drawn in Fig.7. The x-axis is the time percentage of bad users on fake servers, while the y-axis is the number of users who fall into the interval of this stay time ratio. It seems 50% bad users take less than 10% of their stay time on fake servers.

4.2.3 Potential contacts between good and bad users

From above analysis, it has been proved that the fake servers are failed to attract users to stay for a long time. However, the probability of potential pollution should not be neglected if fake servers make good use of bad users to spread polluted files or viruses. We calculated the co-server time, which is defined as the minimum time interval when one good user meets another bad user on the same normal CS. More formally, let t_g be a given online time of a given good user, S_{tg} be the server this good user connected at time t_g, $T_B(S_{tg})$ be the set of all online times of bad peers that connect to server S_{tg}, then the co-server time is defined as $\mathrm{Min}(t_g(S_{tg})\text{-}T_B(S_{tg}))$. After accumulating enough samples, in Fig.8 we plot the PDF in black dotted line and CDF in blue star line of the co-server time. It's revealed that a good user has over 94% probability to meet a bad user if it stays on a normal server for more than 30 minutes.

From this subsection, we conclude that fake servers cannot attract users to stay longer. However, fake servers do disrupt users' stay time distribution at the rank of normal server. And potential frequent contacts between good and bad users should be noticed and utilized by copyright protectors. We are interested in user behaviors on selecting servers, and we plan to model the users' server selection behaviors in further work.

4.3 Some Suggestions

For users in eDonkey networks, choosing trustworthy servers and refusing to exchange or update server list are passive but secure strategies. The static blacklist method works very limited and cannot keep up with the dynamic update of fake servers. So we think client software developer should import a real-time fake server-detecting module, which is similar to our FSD. The feature-based online detection method can effectively find those dynamic fake servers as well as static ones.

Fake servers do bring some troubles to users, but they can't catch most of users. Considering the rank distribution of servers, make full use of a few popular servers instead of establishing many unpopular ones, is much more effective in manipulating users. Furthermore, utilizing bad users can influence the pollution diffusion in eDonkey networks greatly, for the probability of two users of different kind is high enough.

5. CONCLUSION AND FUTURE WORK

In this paper, we evaluated a typical real tussle[1]–fake server problem between copyright protectors and users in eDonkey networks based on our real measurements. After analyzing the trace collected for more than 24 days, we find that fake servers, which are in the majority in number, have been connected to by only 30.9% users. It seems that fake servers cannot catch primary or expert users. The conclusion can be made that nowadays fakes servers don't work well for protecting the copyright of files. Therefore, usurping popular servers may be a better method to collect a large amount of clients' information, considering the rank distribution (Zipf-like) of servers.

However, fake servers do disrupt user's stay time distribution. In addition, good users have high probability of contacts with bad users. Then copyright protectors can pollute or even destroy the eDonkey networks by utilizing users' potential frequent contact chances. In this way, we could not underestimate the influence of fake servers. For defenders, it's suggested that client software developers should import a feature-based detection module for fake servers. Thus users could avoid being polluted by most of fake servers more effectively.

In future work, we will improve our crawler to trace more users and their long time behaviors more comprehensively. On one hand, these datasets can help us confirm our conclusions more extensively. And on the other hand, we will try to model users' selection behaviors on servers and devise a mechanism for load balancing. Measuring the pollution and attacks in kademlia network[14] also interests us

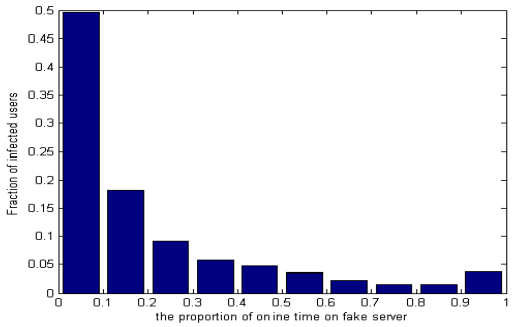

Figure 7: Stay time ratio of bad users on fake servers

very much.

6. REFERENCES

[1] D. Clark, J. Wroclawski, K. Sollins and R. Braden, *Tussle in cyberspace-defining tomorrow's Internet*, In Proc. of SIGCOMM, 2002.

[2] Uichin Lee, Min Choi, Junghoo Cho, M. Y. Sanadidi and Mario Gerla, *Understanding Pollution Dynamics in P2P File Sharing*, IPTPS'06.

[3] Jian Liang, Rakesh Kumar, Yongjian Xi and Keith W. Ross, *Pollution in P2P File Sharing Systems*, Proc. of the IEEE INFOCOM 2005, Vol.2. Miami:IEEE Press, 2005.

[4] Richard Thommes, and Mark Coates, *Epidemiological Moedelling of Peer-to-Peer Viruses and Pollution*, INFOCOM,2006.

[5] Rakesh Kumar, David D. Yao, Amitabha Bagchi, Keith W. Ross and Dan Rubenstein, *Fluid Modeling of Pollution Proliferation in P2P Networks*, ACM SIGMETRICS Performance Evaluation Review, vol. 34, pp.335-346,2006.

[6] K. Tutschku, *A measurement-based traffic profile of the eDonkey filesharing service*, In PAM'04.

[7] J. L.Guillanme and S. Le-Blond, *Statistical properites of exchanges in P2P systems*, In PDPTA'04.

[8] Jia Yang, Hao Ma, Weijia Song, Jian Cui and Changling Zhou, *Crawling the eDonkey Network*, Proceedings of the Fifth International Conference on Grid and Cooperative Computing Workshops,2006.

[9] S. B. Handurukande, A. -M. Kermarrec, F. Le Fessant, Massoulie and S. Patarin, *Peer Sharing Behavior in the eDonkey Network, and Implications for the Design of Server-less File Sharing Systems*, EuroSys,2006.

[10] Yoram Kulbak and Danny Bickson, *The eMule Protocol specification*

[11] http://www.gruk.org/list.php

[12] http://ed2k.2x4u.de

[13] http://www.maxmind.com

[14] P. Maymounkov and D. Mazieres, *Kademlia: a peer-to-peer information system based on the XOR metric*, in Proceedings of the 1st International Workshop on Peer-to-Peer Systems(IPTPS'02)

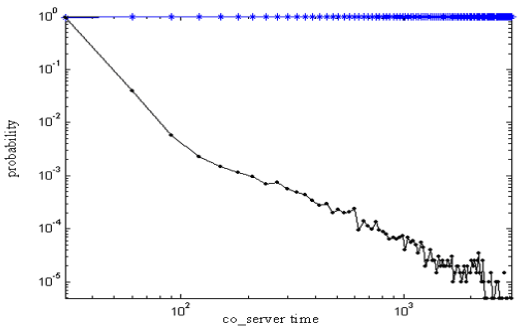

Figure 8: The probability of co-server time