Chapter 11

IDENTIFYING FIRST SEEDERS IN FOXY PEER-TO-PEER NETWORKS

Ricci Ieong, Pierre Lai, Kam-Pui Chow, Michael Kwan and Frank Law

Abstract This paper describes a new approach for identifying first seeders in illegal file sharing investigations involving Foxy, one of the most popular Chinese peer-to-peer networks. In identifying first seeders, the approach focuses on determining the slow-rising period of the cumulative seeder curve instead of merely measuring the number of seeders. The relationships between file popularity, number of packets and the maximum upload limit during the time that the first seeder is connected to the network are also analyzed. These relationships are used to specify rules that investigators can use to determine if an identified seeder is, in fact, the first seeder.

Keywords: Peer-to-peer network forensics, Foxy network, initial seeder

1. Introduction

The Foxy peer-to-peer (P2P) network is very popular in Chinese markets such as Hong Kong and Taiwan. It has approximately over 500,000 Chinese language users at any given time, including users from Asia, Australia, Europe and America. Foxy rapidly disseminates images, music and television programs throughout the global Chinese community. However, Foxy has also been used to distribute racy photographs, pornographic movies and sensitive documents. Once a file is shared on the network, it is practically impossible to completely remove the file, even if the owner wishes to control or block its distribution.

The identification of the first seeder is always the ultimate goal of a P2P file sharing investigation. One approach for verifying if an identified seeder is the first seeder is to determine if the seeder was connected during the slow-rising period of the cumulative seeder curve. However, it is difficult to identify the slow-rising period in a precise manner.

This paper describes an alternative approach to identify the slow-rising period instead of measuring the number of seeders. The approach is based on the observation that the file download duration fluctuates but reduces during the slow-rising period and becomes steady throughout the rapid-rising period; after this time, it increases again when peers leave the network. The paper also examines and clarifies the relationships between file popularity, number of packets and the maximum upload limit during the time that the first seeder was connected. These relationships are used to derive rules that help determine the first seeder in a P2P network.

2. Background

In the Foxy network, file contents are distributed from source uploaders or seeders. A seeder can either be a leaf node or hub node. Leaf nodes are ordinary nodes that contribute the majority of files shared in the Foxy network. Leaf nodes link to hub nodes, which act as "big brothers" who regulate and distribute relevant queries to the peers.

A node searching for a file sends the query to a hub node, which passes the query to the connected nodes. When a connected node indicates that it has the file available for download, the requesting node proceeds to download the file from the node, which becomes the seeder of the requested file. If the seeder is the first node in the Foxy network to upload a particular file, it is known as the first seeder of the file.

The first node that indicates the availability of a file for download could be the first seeder. The propagation of a file can be restricted by identifying and dealing with the first (and early) seeders before the file is widely disseminated. Moreover, the identification of the first seeder is a key goal in investigations involving the illegal distribution of files in P2P networks [1, 2, 5, 6]. This is because only the initial seeders of a file can be prosecuted for their intention to distribute the file under the Hong Kong legal system.

2.1 Identifying Initial Seeders

The initial seeder or first seeder is the seeder or seeders who initiate the distribution of a file in a P2P network. Peers download the file from this seeder. Observations of the download scenario from the beginning – before the file is distributed on the network – should make it easy to identify the first seeder. However, aside from the first uploader of the file, no one can know exactly when the file began to be distributed. The identification of first seeder is not a trivial task.

In some P2P networks (e.g., BitTorrent), it is required to announce the file name and/or download location; this simplifies the task of identifying the first seeder. However, announcements are not mandatory in protocols such as eDonkey, Gnutella and Gnutella 2. The publication of the availability of a file is performed via a searching mechanism within the protocols. Consequently, even if the keyword of a shared file is identified in a forum, no direct link can be drawn between the forum and the first seeder as in BitTorrent.

The identification of the first seeder is also affected by the time when the search function is initiated. If a single seeder is found at the beginning of the file distribution process, the seeder is likely to be the first seeder. However, without a reference time for file publication, an investigator would not be able to confirm when the file distribution started and how long the seeder was active.

Recently, two methods for identifying an initial seeder (or one of the first few seeders) have been published. The first method (Method 1) [5] repeatedly issues identical query patterns submitted by requesting addresses within a short period of time in the Foxy network for the query hits of interest that are returned. The second method (Method 2) [6] is based on the assumption that the growth of seeders in a P2P network follows the cumulative amount of seeders ("seeder curve"). Method 2 engages two rules:

- **Rule 1:** If a seeding peer is found to be reachable and connectable during the slow-rising period, the seeding peer is the first uploader.
- **Rule 2:** If a seeding peer is found after the file distribution level-off period, it is impossible to confirm that the seeding peer is the first uploader. If the single seeder is found during the slow-rising period, then it is very likely that the single seeder is the first seeder.

2.2 Identification Challenges

Several experiments were performed to verify the accuracy of the two methods. The experiments were performed using a modified Shareaza client, an open source P2P client that supports the Gnutella 2 protocol [8]. During the Foxy file sharing process, a SHA-1 hash value generated by the client is used as the identity of the file being shared. Thus, when the SHA-1 hash value is found, the associated file has already been uploaded.

As reported elsewhere [5, 6], our experiments confirmed that the number of query packets increase shortly before and after the appearance of the first seeder. Increases in the number of queries with SHA-1 values generated from multiple IP addresses were also observed. This proves

that Method 1 could be used to identify the first seeder. However, there are two practical difficulties in implementing this method:

- Numerous hash values are generated in the Foxy network each day. We collected 100,000 to 650,000 new hash values per day from our monitored hubs alone. Monitoring all the hash values in the network would require a massive amount of computational resources.
- The growth rate of duplicate hash values is low. The number of identical hash values recorded for some files is as low as three or four hits over a 24-hour period, unless the files are very popular. Continuously monitoring all the files in the Foxy network would require significant resources.

It would appear that Method 2 is practical because the investigator only has to determine if the seeder is found in the slow-rising period. However, several questions must be answered:

- How can one confirm that the network monitoring was performed before the file of interest was widely distributed?
- How likely is the identified seeder one of the first few seeders?
- How can one confirm that the seeder was, in fact, identified during the slow-rising period?

For these reasons, Methods 1 and 2 may not be completely applicable to the Foxy network.

2.3 Simulation Challenges

Observations of file distribution in the Foxy network for clients using Gnutella 2 are not easily performed. The Gnutella 2 protocol used in Foxy is a decentralized P2P protocol. Every Foxy hub behaves as a searching server, so the returned results may only represent a portion of the search results from the entire Foxy network. Even if a suspected seeder is spotted, it is only possible to observe the localized query hit results. Also, it is not possible to confirm if the suspected first seeder is truly the first seeder or just one seeder in the swarm of seeders in the entire Foxy network during the file sharing process.

Peer nodes enter and leave the Foxy network very frequently. However, when a hub node leaves Foxy, the connected nodes restructure themselves by connecting to hubs and leaf nodes. This restructuring affects the search results returned to a leaf node.

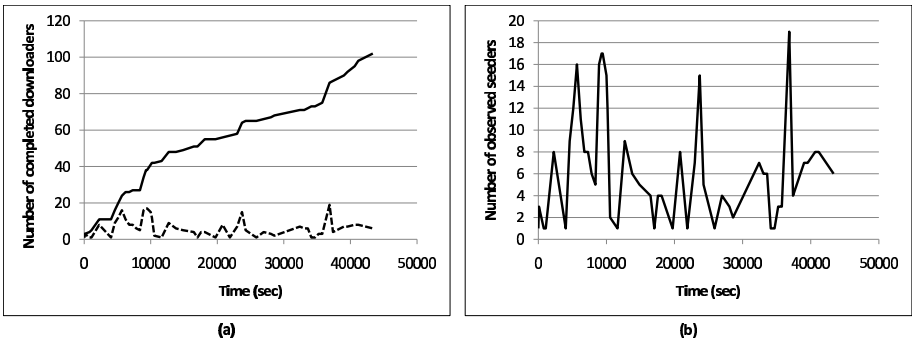

Figure 1. Seeder curves observed for a Shareaza-Foxy client.

A potential problem arises when a file has already been broadcasted and downloaded in an unmonitored part of the Foxy network. An identified sole seeder from a Foxy client could be one of the completed downloaders in the unmonitored network. If it recently connected to a monitored client as a seeder, it should not be considered as the first seeder.

Figure 1 presents the seeder curves for a Shareaza-Foxy client. Figure 1(a) shows the cumulative number of seeders (seeder curve) for a small but popular file (top plot). Note that the cumulative number of seeders increases throughout the period. The bottom plot (dashed line) in Figure 1(a) shows the actual number of seeders for the same file; its "zoomed-in" version is shown in Figure 1(b). The plot in Figure 1(b) shows nine instances where the number of observed seeders is one. However, if the number of seeders really drops to one, the overall download rate experienced by downloaders should be much less than the observed rate.

No matter how extensive the experiment, the number of hubs that can be monitored is only a small fraction of the total number of hubs in the Foxy network. Consequently, we decided to observe the file sharing behavior in a simulated Foxy network. By controlling the simulation environment, the behavior during the slow-rising period can be analyzed.

3. Simulation Experiments

This section describes the simulation experiments involving the Foxy network.

3.1 SimFoxy

Several researchers have conducted simulations of P2P networks [4, 7]. However, most of them use queuing models to simulate P2P network

performance or focus on the Gnutella and BitTorrent protocols rather than the Gnutella 2 protocol.

Instead of relying on existing simulation programs, we built our own program, SimFoxy, to simulate the behavior of Foxy network clients. SimFoxy focuses on the download process after a file has been identified. It may be used to simulate a Foxy network that shares a single file from a single seeder with a predefined number of peers. The numbers of seeders and peers are recorded throughout a simulation. Various parameters (e.g., file packet number, packet size, upload and download connection limits, upload and download rates, etc.) that affect file propagation behavior are implemented as adjustable parameters in SimFoxy.

3.2 Simulation System

In most P2P downloads, file sharing initiation can be modeled as discrete events with Poisson inter-arrival times [4, 7]. The simplest way to analyze the initiation stage behavior of file distribution in the Foxy network is to develop a Foxy simulation environment using a discrete event simulation package. We employ SimFoxy, a Python-based implementation that uses the SimPy simulation module [9]. SimPy is an object-oriented, process-based discrete event simulation language developed in Python 2.x that supports the simulation of multiple processes.

All file sharing and download processes in SimFoxy are simulated at the packet level. A packet object is the basic unit that is uploaded and downloaded during the file sharing process. The packet-level simulation is performed by limiting the resource capacity of packets such as download connections by adding the amount of time expected to be used in the download process. Download activities are initiated by packets after receiving requests from peers. After one download activity involving a packet is completed, the packet pauses until it is invoked by another downloading peer.

The downloading and uploading of packets by a peer are regulated by the server object. To simplify the architecture and to reduce resource usage, instead of simulating the server as a hub in the Foxy network, the entire peer list, partial peer list, seeder list and peer availability for downloading are added as accessible resources in the server object. This reduces the resources required for simulation.

3.3 Simulation Assumptions

Our SimFoxy implementation incorporated several assumptions to ensure that it would be possible to analyze the effects of various environmental parameters in the Foxy network. First, we assumed that one

original seeder exists in the simulated Foxy network and no new seeder connects to the Foxy network during the simulation. Second, all the peer nodes participate in uploading and downloading the target file only; this reduces the effect on the download bandwidth due to the distribution of other files. Third, all peers are only able to upload the file as a seeder after they have completely downloaded all the packets from the seeder; this captures the behavior of a Foxy 1.9.7 client, for which partial downloads of incomplete seeds are not supported.

The purpose of the simulation was to study the initial stage of peer upload during file distribution. In order to shorten the simulation time, we concentrated on the simulation of the first 100 to 1,000 peers during the upload and download periods. Consequently, in the simulations, the maximum number of preset peers in SimFoxy was limited to 1,000.

The upload connectivity and download connectivity are preset in Foxy clients. To reduce the complexity of downloads, all the Foxy clients should be configured to support a maximum of five downloads and ten uploads. These values were tested in our simulation experiments.

The downloading of file fragments in the network is controlled by the Foxy client. To simplify the simulation setup, download and upload fragments were defined to be 500 KB per packet, which is the packet size observed in the real Foxy download packet request query. Therefore, the complete download of a 10 MB file requires twenty 500 KB packets to be downloaded by a node.

The connection speed between an uploader node A and a downloader node B is assumed to be the minimum of the upload rate of A and the download rate of B. Usually, this is the upload speed of node A because the upload speed is normally less than the download speed. When one additional node is connected to node A, the upload speed is divided equally among the two nodes.

3.4 Simulation Sets

More than 100 simulation experiments were performed using SimFoxy. The simulations were performed by varying five parameters: (i) average inter-arrival time (T_{arr}); (ii) number of peers interested in the target file during the simulation period (N_p); (ii) simultaneous upload peer limit (N_u); (iii) simultaneous download peer limit (N_d); (iv) average inter-departure time (T_{dep}); and (v) file size expressed as the number of 500 KB packets (N_{pkt}). The upload and download rates of all the peers and the seeder were set to 1,280 KB/s (1 Mbps) and 2,560 KB/s (2 Mbps), respectively. Fixing the upload and download rates of all the peers reduces the effect of randomness on the parameter measurements.

The simulation experiments were divided into four sets defined below.

Set 1 This set of simulations investigated the effects of changes in the file size (i.e., number of packets (N_{pkt})). In Sets 1(a), 1(b) and 1(c), experiments were performed by varying N_{pkt} only. In Set 1(d), N_{pkt} was fixed, but different upload (N_u) and download (N_d) limits were used. The inter-arrival (T_{arr}) and inter-departure (T_{dep}) times were set to 5 seconds and 10 seconds, respectively.

- **Set 1(a):** Simulations involving different numbers of peers; N_{pkt} = 20 (∼10 MB); N_p = 1 to 1,000; N_u = 5; N_d = 10.
- **Set 1(b):** Simulations involving different numbers of packets; N_{pkt} = 20 to 400 (∼200 MB); N_p = 5; N_u = 10.
- **Set 1(c):** Simulations involving the sharing of large files; N_{pkt} = 800 (∼0.4 GB), 1,600 (∼0.8 GB), 3,200 (∼1.6 GB); N_p = 1, 3, 4; N_u = 5; N_d = 10.
- **Set 1(d):** Simulations involving different upload and download limits; N_{pkt} = 20; N_p = 1, 2, 5, 10, 25, 50; N_u = 5, 10, 20, 40; N_d = 10, 20, 40, 50.

Set 2 This set of simulations investigated the effects of changes in the inter-departure time (T_{dep}) after download completion. In Sets 2(a), 2(b), 2(c), 2(d) and 2(e), experiments were performed by varying T_{dep} from 10 to 1,200 seconds with the upload (N_u) and download (N_d) limits fixed at 5 and 10, respectively; the number of packets (N_{pkt}) limited to 100 pieces; and the number of peers (N_p) fixed at 100 nodes.

- **Set 2(a):** Simulation involving different inter-departure times; N_{pkt} = 100; N_p = 100; N_u = 5; N_d = 10; T_{dep} = 10 seconds.
- **Set 2(b):** Simulation involving different inter-departure times; N_{pkt} = 100; N_p = 100; N_u = 5; N_d = 10; T_{dep} = 400 seconds.
- **Set 2(c):** Simulation involving different inter-departure times; N_{pkt} = 100; N_p = 100; N_u = 5; N_d = 10; T_{dep} = 800 seconds.
- **Set 2(d):** Simulation involving different inter-departure times; N_{pkt} = 100; N_p = 100; N_u = 5; N_d = 10; T_{dep} = 1,000 seconds.
- **Set 2(e):** Simulation involving different inter-departure times; N_{pkt} = 100; N_p = 100; N_u = 5; N_d = 10; T_{dep} = 1,200 seconds.

Set 3 This set of simulations investigated the effects of changes in the inter-arrival time (T_{arr}). In Sets 3(a), 3(b), 3(c) and 3(d), experiments were performed by varying the inter-arrival time patterns (all at once, periodic, random and uniform random) and sharing 100 packets with 100 peers with an inter-departure time (T_{dep}) of 1,000 seconds. The upload (N_u) and download (N_d) limits were fixed at 5 and 10, respectively.

- **Set 3(a):** Simulation involving 100 peers downloading simultaneously; $T_{arr} = 0$ seconds.
- **Set 3(b):** Simulation involving 100 peers starting their downloading at different inter-arrival times; $T_{arr} = 5$, 10, 20, 40 seconds.
- **Set 3(c):** Simulation involving 100 peers starting their downloading at random times; T_{arr} = random: 0 to 1,200 seconds.
- **Set 3(d):** Simulation involving 100 peers starting their downloading at uniform random times; T_{arr} = uniform random: 0 to 1,200 seconds.

Set 4 This set of simulations investigated the effects of changes in the inter-arrival time (T_{arr}) patterns (periodic, random, uniform random and Poisson random). In Sets 4(a), 4(b) and 4(c), experiments were performed by sharing 200 pieces of packets (N_{pkt}) with different inter-arrival time patterns (all at once, random and uniform random). In Sets 4(d) and 4(e), experiments were conducted by sharing of 20 pieces of packets with $T_{dep} = 1$ to 10 seconds and $T_{arr} = 100$ seconds; and by sharing 20 and 40 pieces of packets with Poisson random T_{arr} ($\lambda = 0.25$), respectively. The upload (N_u) and download (N_d) limits were fixed at 5 and 10, respectively.

- **Set 4(a):** Simulation involving a popular file being downloaded by all the peers simultaneously; $N_{pkt} = 200$; $N_p = 100$; $T_{arr} = 0$ seconds; $T_{dep} = 100$ seconds.
- **Set 4(b):** Simulation involving random incoming peers; $N_{pkt} = 200$; $N_p = 100$; T_{arr} = random: 0 to 1,200 seconds; $T_{dep} = 100$ seconds.
- **Set 4(c):** Simulation involving uniform random incoming peers; $N_{pkt} = 200$; $N_p = 100$; T_{arr} = uniform random: 0 to 3,600 seconds; $T_{dep} = 0$, 120, 2,000, 3,000, 8,000 seconds.
- **Set 4(d):** Simulation involving slow inter-arrival times; $N_{pkt} = 20$; $N_p = 1$ peer/second, 1 peer/10 seconds; T_{arr} = periodic: 1 peer/second; $T_{dep} = 100$ seconds.

- **Set 4(e):** Simulation involving Poisson random incoming peers with different inter-departure times; $N_{pkt} = 20,\ 40$; $N_p = 100$; T_{arr} = Poisson random ($\lambda = 0.25$); $T_{dep} = 100,\ 1{,}000$ seconds.

These four sets of simulation experiments facilitated the systematic analysis of the effects of various parameters on the download duration, slow-rising period and the time required for the appearance of the second seeder.

3.5 Observations

Several observations can be made based on the simulation experiments.

Comparison of Experimental and Simulation Results Experiments performed in the actual Foxy network cannot reflect the file distribution over the entire network. Therefore, the behavior in the real and simulated Foxy networks may not match completely.

To demonstrate how closely SimFoxy simulates the real Foxy network, some simulations were conducted using parameters obtained from real-world environments. Actual Foxy network download scenarios were captured by conducting two file downloads at different instants. In both cases, observations based on our modified Foxy client revealed that the incident was initiated by one observable seeder. Following this, seeder growth curves were constructed using SimFoxy with similar criteria.

Figure 2 shows four seeder curves for real and simulated Foxy networks. Figure 2(a) shows the observed number of seeders (dashed line) and the cumulative number of seeders (unbroken line) for a small, popular file. Figure 2(b) shows the simulation results for a small to medium sized file with a rapid arrival rate in SimFoxy. Figure 2(c) shows the behavior during the first 12 hours for a large, popular file in the Foxy network. Figure 2(d) shows the simulation results for a large file with slow peer departure and rapid arrival rates in SimFoxy. Note that the curves in Figures 2(b) and 2(d) match the majority of the actual completed downloader growth rate curves (dashed lines) in Figures 2(a) and 2(c).

Relationship between File Size and Download Time After clarifying the relationship between file size and the download completion time of the first downloader, we attempted to determine how the number of competing peers affects the download time based on the results obtained in Sets 1(c) and 1(d).

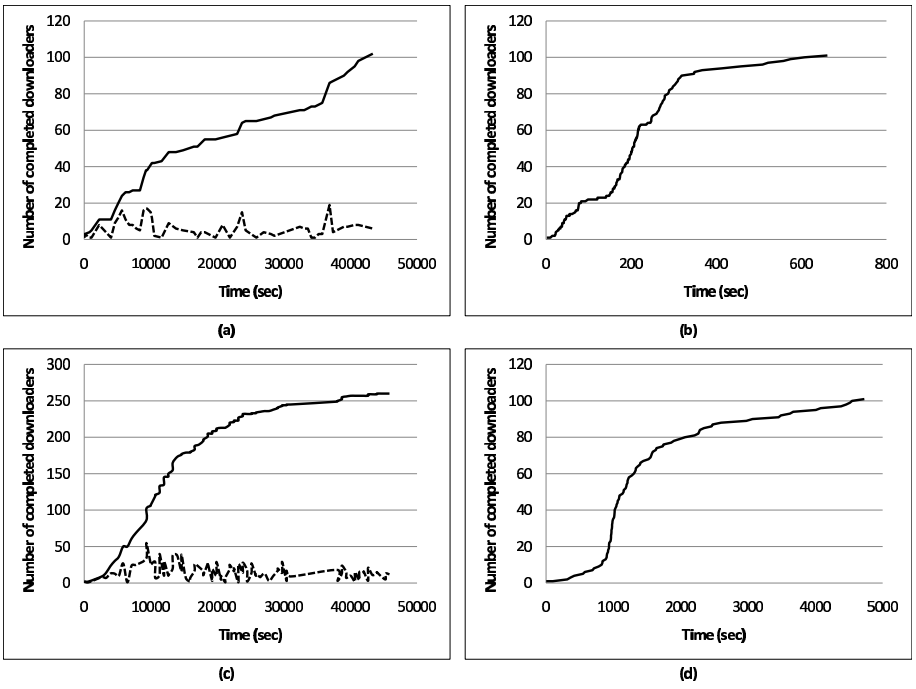

Figure 2. Four seeder curves for real and simulated Foxy networks.

Th download time is affected by the number of peers competing for a single file as well as by the file size (Set 1 simulations). If the file download duration for one downloader is T_x, the increment due to additional peers, which we call the "download increment ratio" (R) is $(T_y - T_x)/T_x$, where T_y is the download duration time when the number of downloaders is greater than one.

Instead of varying the file size, we plotted the download increment ratio against increments in file size for three peers and four peers downloading simultaneously (Figure 3(a)). The x-axis is R and the y-axis is the size of the target file (in MB). The curve in Figure 3(a) shows the effect of the number of peers competing for same file at one time (P_i means that i peers are competing). Note that R is affected by the file size but eventually levels off.

Effect of Upload Limit on Download Time Instead of comparing the number of competing peers, the download increment ratio R for different upload limits was compared based on the Set 1(d) results. The download increment ratio R corresponding to the same file source with the same competing peers was measured for two different upload limits

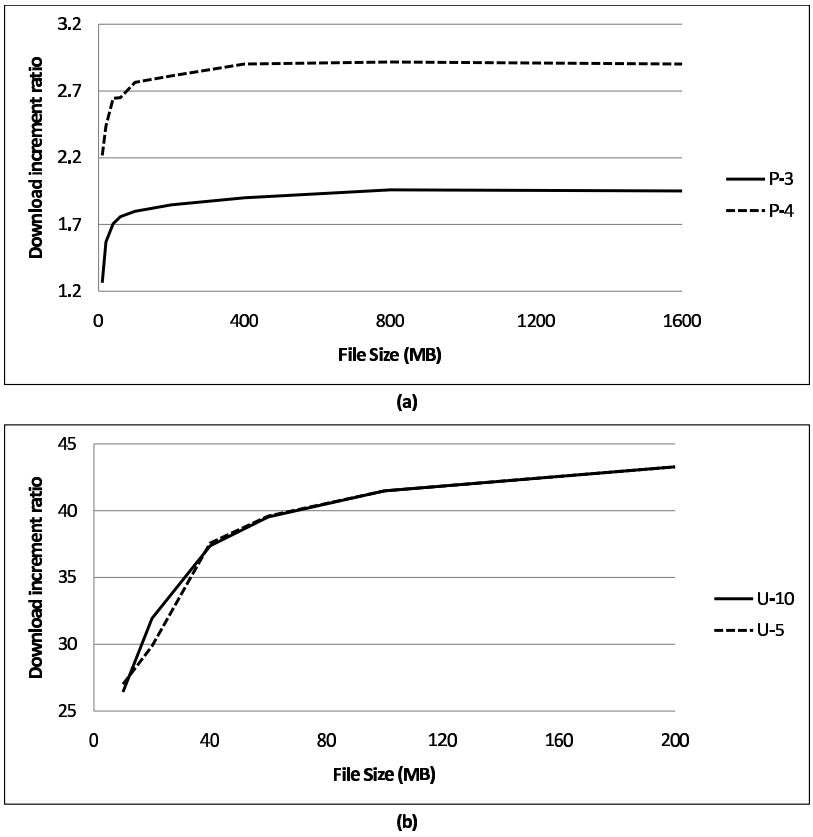

Figure 3. Download increment ratio versus file size.

(5 peers and 10 peers). Figure 3(b) highlights the effect of the upload limit on the download increment ratio (U_j means the upload limit is j peers). Note that the upload limit has almost no effect on the download increment ratio R.

Effect of Departure Rate on the First Downloader Completion Time The departure rate of peers affects the overall file distribution behavior. The Set 2 simulations show that the departure rate of peers definitely affects the download time of successive peers. With a higher departure rate, the download speed after the rapid-rising period would be greatly reduced. However, because the first downloader must complete the download from the first seeder, it is only affected by the behavior of the first uploader. Thus, the departure rate of peers was found to have no effect on the completion time of the first downloader.

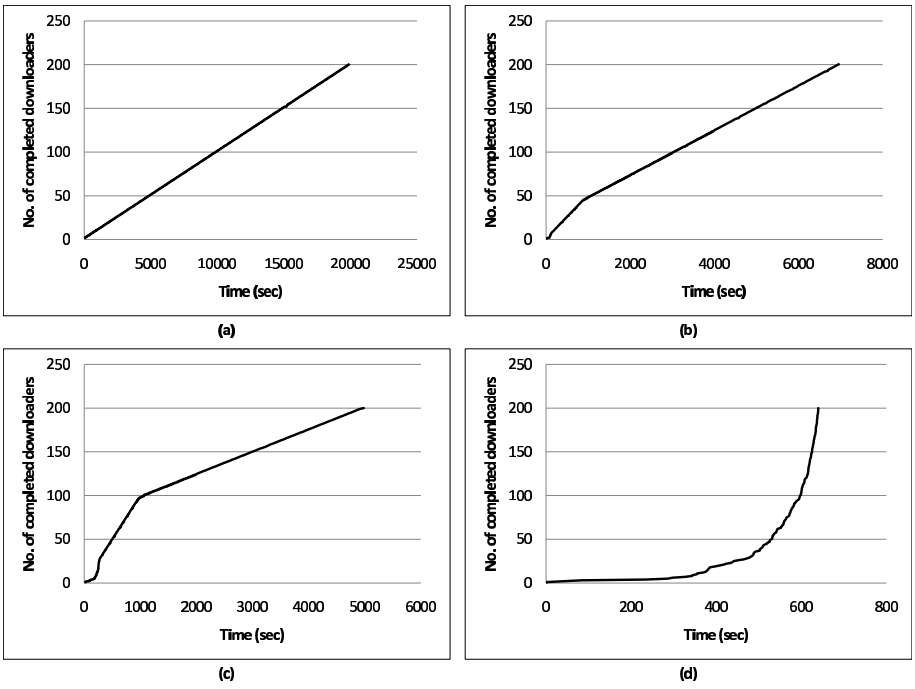

Figure 4. Cumulative number of downloaded copies over time.

Effect of Arrival Rate on Peer Download Time Using the data from the Set 3 and Set 4 simulations, the growth rate of seeders was measured against different peer inter-arrival times from 0 to 100 seconds. Figure 4 shows the cumulative number of downloaded copies over time for 200 peers initiated with different inter-arrival times (Figure 4(a): 100 seconds; Figure 4(b): 20 seconds; Figure 4(c): 10 seconds; Figure 4(d): 0 seconds). The download duration is not affected by the peer inter-arrival time if the inter-arrival time is greater than the download duration. However, when the inter-arrival time is reduced, the download duration is affected and the relationship changes from a straight line to a curve (Figure 4).

Effect of Peer Download Time Variation Instead of simply measuring the growth rate of seeders as in the Set 3 simulations, we measured the first connected time of peers and calculated their file download completion duration (download duration). In the Set 4 simulations, all the peers had the same simulated upload and download speeds and the download duration was plotted against the first connected time as in Figures 5 and 6. The two figures show the cumulative number of seeders

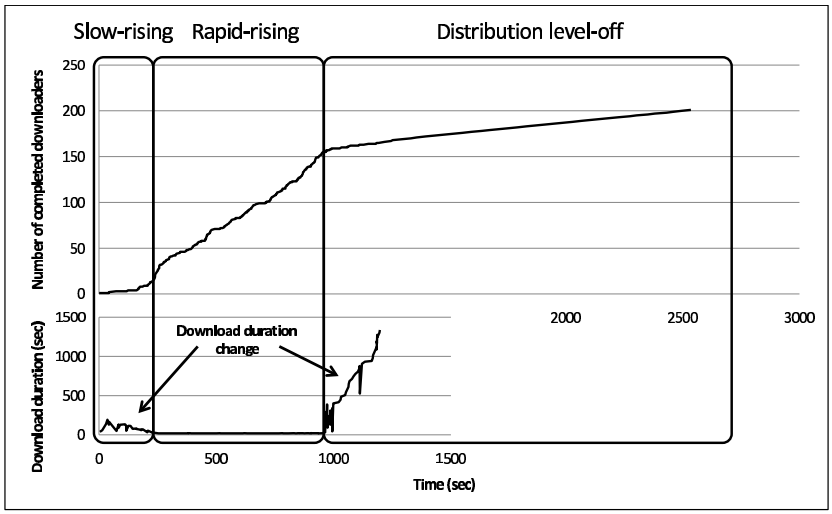

Figure 5. Downloader (upper) and download duration curves (lower) (random).

(seeder curve) and the corresponding download duration curve for two peer inter-arrival patterns. Figure 5 shows the peer download behavior when the peers arrive randomly between 0 to 1,200 seconds based on Set 4(b).

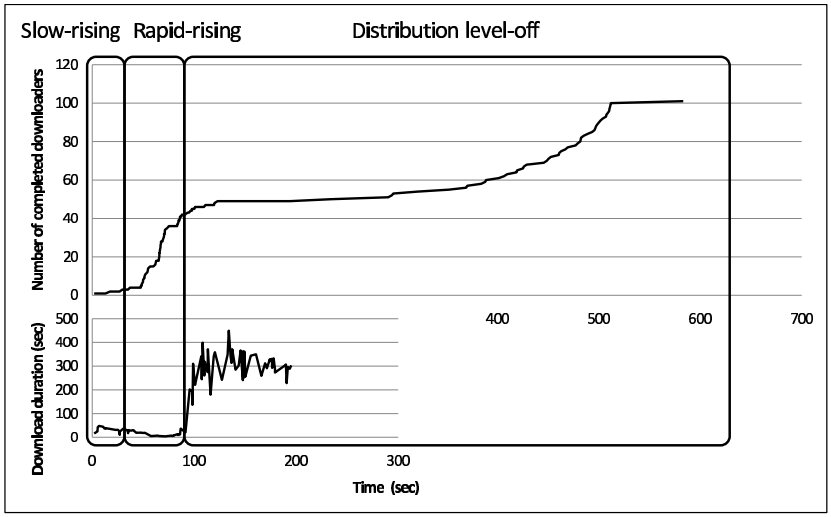

Figure 6. Downloader (upper) and download duration curves (lower) (Poisson).

Figure 6 shows the peer download behavior for a Poisson inter-arrival time [3] with $\lambda = 0.25$ according to Set 4(e). The value of λ was set to

0.25 to ensure that the peers connected to the network for download were spread across the entire download period instead of being concentrated at the beginning of the download period. Note that the download duration of peers exhibits a strong relationship with the slow-rising period and the rapid-rising period.

The download duration (lower) curves in Figures 5 and 6 show that the download time reduces throughout the slow-rising period. The download duration is rather short throughout the rapid-rising period, after which it rises when peers that complete their downloads leave the Foxy network.

4. Simulation Results

This section discusses the main results obtained in the simulation experiments.

The download duration change is another indicator of the slow-rising period. Figures 5 and 6 show that the download duration T_d is affected by the time when the download request is initiated. During the slow-rising period, peers experience long download times and many download interruptions because a single file resource is shared by multiple peers $P_1, \ldots, P_n$.

When multiple peers provide a resource, a peer P_i may obtain packets from peers other than the seeder S and the download duration T_d is reduced. At the same time, the number of completed downloaders increases much faster – this is observed as the rapid-rising period. The download duration remains approximately the same because the download connectivity is pre-defined in the client.

The equilibrium is disrupted when more peers disconnect after completing their downloads than the increment in the number of completed downloaders. As the number of available seeders goes down, T_d increases again until no more peers join or leave the Foxy network.

The popularity of a resource increases the first peer download time. Analysis of the simulation results reveals that the peer inter-arrival time and the file size greatly affect the seeder growth rate. When no peers compete for the same seeder, the peer download time is essentially the same as the time required to download the resource from a single server. As more peers download from a source simultaneously, the download completion time increases.

In the actual Foxy network, peers search for a popular file after it is published and announced. Peers $P_1, \ldots, P_n$ could attempt to connect to the same file source simultaneously. Seeder S uploads a packet PK_j to Peer P_i when the request for PK_j by P_i is accepted by S.

However, the acceptance of a request is limited by the upload limit of S. When the number of requests from peers $P_i, \ldots, P_n$ exceeds the maximum upload limit, the peers whose requests were accepted earlier by S are granted packet PK_j while the other peers have to wait until the earlier download requests are completed. As more peers with same download rate request the same file from S, the probability of obtaining PK_j by P_i drops. Thus, the download duration of the second seeder from the first seeder is increased. According to our experiments, if all the peers who request the file have the same configuration, then the number of peers is directly proportional to the lengthening of the download duration.

5. Seeder Identification Rules

A single seeder in a Foxy network can be identified, but verifying that the seeder is one of the initial seeders is not a simple task. In our previous research [6], we showed that a seeder can be identified as one of the first few seeders if the sharing of the resource falls within the slow-rising period of the seeder curve. However, confirming whether or not the period of interest falls within the slow-rising period is also difficult.

Our simulation results reveal that the download duration reflects the behavior of seeder curve. Instead of observing the number of seeders in the Foxy network, investigators could perform multiple downloads of the same file at different times from different clients. Then, the results could be analyzed using the following rules.

- **Rule 1:** If two or more observed download durations T_d drop during consecutive downloads, the observed seeders should be collected within the slow-rising period.
- **Rule 2:** If T_d remains roughly steady at the stable download duration, the observed seeders should be collected during the rapid-rising period.
- **Rule 3:** In the case of a popular file, the download duration T_d for the second seeder is lengthened. This duration is directly proportional to the number of peers that simultaneously download the file. The slow-rising period is lengthened by the number of requesting peers. Therefore, the period for identifying initial seeders is lengthened by the popularity of the file.
- **Rule 4:** If the file download completion time T_d is less than the peer inter-arrival time T_{arr}, then it is impossible to confirm the appearance of first seeder.

Rules 1 and 2 can help confirm that the seeder is collected during the slow-rising period. Rules 3 and 4 can help verify that the chance of mistakenly identifying the seeder as the first seeder is reduced.

6. Conclusions

The identification of the first seeder is a crucial task in investigations of illegal file sharing in P2P networks. The approach for identifying first seeders in the popular Foxy network based on the slow-rising period of the cumulative seeder curve can be very helpful in investigations. Furthermore, rules derived from the relationships between key parameters – such as file popularity, number of packets and the maximum upload limit when the first seeder is connected to the network – assist investigators in determining if an identified seeder is, in fact, the first seeder.

The work presented in this paper is experimental in nature. Our future research will develop and validate a mathematical model that expresses the relationships between the number of hubs, number of peers, seeder growth rate and download duration. Such a model would support network forensic investigations as well as the design and implementation of strategies for controlling illegal file sharing in P2P networks.

References

[1] K. Chow, K. Cheng, L. Man, P. Lai, L. Hui, C. Chong, K. Pun, W. Tsang, H. Chan and S. Yiu, BTM – An automated rule-based BT monitoring system for piracy detection, *Proceedings of the Second International Conference on Internet Monitoring and Protection*, p. 2, 2007.

[2] K. Chow, R. Ieong, M. Kwan, P. Lai, F. Law, H. Tse and K. Tse, Security Analysis of the Foxy Peer-to-Peer File Sharing Tool, Technical Report TR-2008-09, Department of Computer Science, Hong Kong University, Hong Kong, 2008.

[3] P. Consul, *Generalized Poisson Distributions: Properties and Applications*, Marcel Dekker, New York, 1989.

[4] B. Fan, D. Chiu and J. Lui, Stochastic differential equation approach to model BitTorrent-like P2P systems, *Proceedings of the IEEE International Conference on Communications*, pp. 915–920, 2006.

[5] R. Ieong, P. Lai, K. Chow, F. Law, M. Kwan and K. Tse, A model for Foxy peer-to-peer network investigations, in *Advances in Digital Forensics V*, G. Peterson and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 175–186, 2009.

[6] R. Ieong, P. Lai, K. Chow, M. Kwan, F. Law, H. Tse and K. Tse, Forensic investigation and analysis of peer-to-peer networks, to appear in *Handbook of Research on Computational Forensics, Digital Crime and Investigation: Methods and Solutions*, C. Li (Ed.), Information Science Reference, Hershey, Pennsylvania, 2010.

[7] D. Qiu and W. Sang, Global stability of peer-to-peer file sharing systems, *Computer Communications*, vol. 31(2), pp. 212–219, 2008.

[8] Shareaza, Shareaza 2.5.2.0 (shareaza.sourceforge.net), 2010.

[9] SimPy, SimPy Simulation Package (version 2.0.1) (simpy.sourceforge.net), 2009.