Chapter 13

# A MODEL FOR FOXY PEER-TO-PEER NETWORK INVESTIGATIONS

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

**Abstract** In recent years, peer-to-peer (P2P) applications have become the dominant form of Internet traffic. Foxy, a Chinese community focused file-sharing tool, is increasingly being used to disseminate private data and sensitive documents in Hong Kong. Unfortunately, its scattered design and a highly distributed network make it difficult to locate a file originator. This paper proposes an investigative model for analyzing Foxy communications and identifying the first uploaders of files. The model is built on the results of several experiments, which reveal behavior patterns of the Foxy protocol that can be used to expose traces of file originators.

**Keywords:** Peer-to-peer network forensics, Foxy network, Gnutella 2 protocol

## 1. Introduction

Recent surveys report that P2P traffic is responsible for 41% to 90% of all Internet traffic [2, 4]. In 2007, two popular P2P file-sharing applications, BitTorrent and eDonkey, contributed 50% to 70% and 5% to 50% of all P2P traffic, respectively [2]. The Foxy P2P file-sharing protocol is gaining popularity in traditional Chinese character markets such as Hong Kong and Taiwan – approximately 500,000 users are active on the Foxy network at any given time [14]. A Foxy client, which is available free-of-charge, provides a user-friendly traditional Chinese interface. It enables users to connect to the Foxy network without any special configuration and to download free music, movies and software with just a few keystrokes and mouse clicks.

Foxy drew worldwide attention when hundreds of photographs of a local pop icon participating in sex acts with female celebrities were disseminated on the Internet [5]. Other cases involving Foxy include the unintentional sharing of sensitive files in Taiwan and Hong Kong [12]. Locating a file originator is very difficult due to Foxy's distributed network. While Hong Kong Customs and Excise are able to identify and prosecute BitTorrent seeders [3], the techniques used for BitTorrent are not applicable to Foxy.

Chow, *et al.* [7] have published guidelines on the secure use of Foxy clients. However, to our knowledge, no publication describes an investigative process that is applicable to the Foxy network. Without a mechanism to identify file originators, it is impossible to collect digital evidence for prosecuting illegal publishers of digital materials as in the pop icon scandal. This paper addresses the issue by proposing an investigative model for analyzing Foxy communications and identifying the first uploaders of files.

## 2. P2P Forensic Tools

Several digital forensic tools have been developed to identify traces of P2P client execution on computers [1, 16]. However, locating the original uploaders of files remains one of the most challenging problems in P2P network forensics.

The BTM tool [6] was designed to monitor the BitTorrent network and identify initial seeders. BTM monitors public web forums where BitTorrent users communicate and announce their torrent files (the seed files for file download in the network). It mimics a BitTorrent client and collects communication information from trackers and peers. Initial seeders are identified by analyzing the collected data.

However, the BTM approach cannot be applied to the Foxy network because Foxy clients use the Gnutella 2 protocol [9]. Moreover, the mechanisms used to broadcast shared files are different. In BitTorrent, a torrent file is published on a web forum for others to download; no auxiliary seed file is needed in the case of the Foxy network.

Nasraoui, *et al.* [13] have proposed a node-based probing and monitoring mechanism for Freenet and Gnunet. Their approach bears some similarity to our method. However, they only provide a high level framework and do not address the issue of locating initial uploaders.

## 3. Foxy Network Overview

Foxy is a hybrid P2P model based on the Gnutella 2 protocol (G2). In the Foxy network, a peer is either an "ultrapeer" or a "leaf node." Ultrapeers are active nodes that coordinate surrounding nodes, filtering and directing query traffic within the Foxy network. Leaf nodes are the most common nodes; they issue search queries, upload and download files. Ultrapeers are used to relay communications from leaf nodes without flooding the Foxy network.

The search mechanism in the G2 network is initiated when a user enters keywords and clicks the "Search" button. The corresponding search query, referred to as a *Q2* packet, is then submitted to the connected ultrapeers. Each ultrapeer verifies and validates the search, and selectively redirects the query to other leaf nodes or ultrapeers. Nodes that are unlikely to have files matching the keywords may not receive the query.

Ultrapeers use a "query hit table" (QHT) to keep track of the files available on its leaf nodes. For each shared file, the file name is hashed using the QHT algorithm [9]. The algorithm hashes both the file name and the keywords that are maintained separately in the QHT. Whenever a QHT entry matches a query either partially or completely, the ultrapeer considers that the corresponding node has the potential to answer the query and forwards the query to that node.

If the receiving node possesses a file with a name matching the received query, a "query hit packet" (called a *QH2* packet) is transmitted back to the requester. The *QH2* packet contains the IP address of the file source as well as the descriptive name of the file. If several nodes respond, the ultrapeer consolidates the results and sends the requestor a newly constructed *QH2* packet. At this point, the requester can initiate a file download based on the search results.

## 4. Foxy Protocol Analysis

In order to determine and analyze the behavior of the Foxy network, traffic generated and received by several Foxy clients was captured using Wireshark [15], a popular network packet capturing tool. This section describes the experimental results based on the analysis of more than 80 sets of Foxy communication network traffic records involving approximately 3 million packets.

### 4.1 Data Collection

Table 1 lists the five data collections (A through E) used to analyze the Foxy protocol. The collections, which are of varying lengths, were executed over a five-month period. Data collections A and B focus on the search results of popular keywords. Data collection C compares and analyzes the search results of a query between two Foxy clients. Data collections D and E investigate how search queries and results propagate across multiple clients.

*Table 1.* Summary of experiments and tested queries.

| DC | Date | Purpose | Query | Sets |
|---|---|---|---|---|
| A | 05/14/08 to 06/10/08 | Identify the newly-announced keyword (`Pol Record`) and analyze the relevant search results. | `Pol Record` | 10 |
| B | 09/11/08 to 09/29/08 | Analyze the *Q2* and *QH2* packets of a newly-announced keyword and compare the results with those from data collection A. Observe the connectivity of Foxy peers. | `yoshinoya` (in Chinese) | 40 |
| C | 09/29/08 | Monitor and analyze *Q2* and *QH2* packets. | `mov00423` | 7 |
| D | 09/29/08 to 10/08/08 | Investigate the propagation of *Q2* and *QH2* packets among multiple clients. | `bhb_od2`, `mov00423`, `foxy testing2209`, `foxy testing1404`, `04c3a2d1`, `4b9277c6`, `ad28ae6e`, `yoshinoya` (in Chinese) | 24 |
| E | 10/09/08 | Analyze the *Q2* and *QH2* packets of a self-published file.`Foxy_testing2209.mp3` | `foxy testing 2209` | 6 |

## 4.2 Experimental Results

This section describes the experimental results obtained during the analysis of the Foxy network.

**Ultrapeer List Changes** In the case of data collection B, the Foxy client was connected and disconnected from the Foxy network several times. By analyzing the network packets, it was observed that the client connected to a random list of available ultrapeers provided by the Foxy Gnutella web cache. Since the web cache arbitrarily selects ten to fifteen ultrapeers from the ultrapeer pool, leaf nodes connect to different sets of ultrapeers regardless of their physical or network locations. Also, different lists of ultrapeers are returned to a leaf node whenever it is reconnected to the Foxy network regardless of the length of the disconnection.

**Ultrapeer and Leaf Node Connectivity** According to the specifications, ultrapeers have a maximum connectivity of 200 nodes. In the case of data collection B, no ultrapeers were found to exceed this value. On the other hand, leaf nodes were found to be connected to no more than 32 ultrapeers at a time. Because the pool of ultrapeers changes over time, the ultrapeers to which a leaf node is connected also changes over time.

At any time, there are about 500,000 nodes connected to the Foxy network. If each ultrapeer and leaf node is connected to the maximum of 200 leaf nodes and 32 ultrapeers, respectively, then there are at least 80,000 ultrapeers in the network.

**Query Results after Reconnections** In the case of data collection D, five files (F1, F2, F3, F4 and F5) with uncommon names were shared using a Foxy client. Another client (at a remote site) was launched to search for these files. In the first connected session, three files (F1, F3 and F4) were returned in the search results. However, after restarting the client, only F2 was located. This shows that different results are received when identical queries are submitted after a reconnection. The likely cause is that, when new files appear, propagation through the ultrapeers takes a period of time and the propagation is potentially limited to nearby peers.

**Searching for Files with Uncommon Names** In the case of data collection C, several queries were executed with the keyword `MOV00423`. This keyword was part of the name of a video file in a rape case [11]. Before the incident, no requester would expect to receive results for this query. However, after the newspapers reported the case, many results were returned for this keyword query.

These results and those for data collection D suggest that files with uncommon file names are difficult to find at the beginning of the file-sharing period. Only after the keyword is queried a number of times are more results returned.

**Query Filtering at Ultrapeers** In the case of data collection E, a leaf node $X$ with the file `Foxy-testing220.mp3` was connected to the Foxy network. Then, another leaf node $Y$ was connected to the Foxy network; this leaf node submitted a query for the file. After two minutes, the client was disconnected and reconnected to the Foxy network.

Upon analyzing the *Q2* packets arriving at $X$, we discovered that the exact matching query `Foxy-testing220` was sent through the ultrapeers. However, alternatives to the query such as `Foxy-testing22` and

*Table 2.* Percentage of *Q2* packets querying $H_S$ in all *Q2* packets

| Date | Duration (min) | *Q2* Packets ($N_1$) | *Q2* Packets Querying $H_S$($N_2$) | $N_2/N_1$ (%) |
|---|---|---|---|---|
| 11/09/08 | 114 | 833 | 767 | 92.1 |
| 12/09/08 | 110 | 1,097 | 1,043 | 95.1 |
| 13/09/08 | 27 | 219 | 201 | 91.8 |
| 14/09/08 | 58 | 419 | 357 | 85.2 |
| 18/09/08 | 22 | 344 | 285 | 82.8 |
| 08/10/08 | 27 | 560 | 386 | 68.9 |

`Foxy-testing2209` did not arrive at $X$. This indicates that the filtering performed at the ultrapeers does not implement a sub-string match.

It was observed that query filtering does not employ exact matching. Of the 559 *Q2* packets that arrived at $X$, only 26 were initiated by $Y$ requesting the target file. In fact, 95% of the queries were not for the file being shared, indicating that filtering does not use an exact-matching method.

Approximately 70% of the irrelevant queries contained identical hash values or very similar file names. In other words, the majority of the queries were dominated by a small set of extremely popular keywords that were similar to each other.

**Hash Querying after an Incident** Table 2 compares the *Q2* packets captured from September 11, 2008 to September 14, 2008; on September 18, 2008; and on October 8, 2008 in data collections B, D and E, respectively. The majority of the queries pertained to a specific hash value, $H_S$, which was associated with the sex scandal mentioned above.

The large number of *Q2* packets querying $H_S$ suggests that popular queries may occupy the query hash table after the announcement of keywords. Fewer queries for $H_S$ appear after the tide passes and/or more peers have downloaded the file,

## 5. Hypothesized Foxy Network Behavior

Apart from the findings discussed above, some important behavior patterns were not observed. These behavior patterns, which pertain to the publication of a new file, searching for file keywords and massive P2P downloading of a file, are critical to the success of any P2P file monitoring and investigation technique. Therefore, hypotheses have to be derived based on known features of the protocol.

In any file-sharing environment, there are five stages before a downloader obtains a file: preparation, initiation, publication, waiting and downloading [10]. The publication, waiting and downloading stages are the only stages where an investigator can collect digital evidence remotely on the Internet, i.e., before examining the suspect machine.

## 5.1 Publication Stage

After the preparation and initiation stages, the existence of a new file must be announced to the world before it can be disseminated widely. Because the Foxy network does not provide a mechanism for publishing newly-shared files, a file uploader must make the announcement independent of the Foxy network. Therefore, no phenomena can be observed in the Foxy network during the initial publication stage.

## 5.2 Waiting Stage

The waiting stage is critical in the Foxy network protocol. Shortly after the publication of a new file, Foxy users search for the file by submitting keywords. If the keywords arouse sufficient publicity, a sudden increase in the number of queries asking for the same keyword are observed in the Foxy network. This is partially reflected in the comparison of the queries in data collections A and B.

## 5.3 Downloading Stage

File sharing completes with the transfer of the entire file from one peer to another. In the ideal case, the file can be distributed to $nk$ peers in $k$ rounds, where $n$ is the maximum number of connected peers and all peers are assumed to be downloading and uploading at the same speed. Suppose there is only one uploader of the file, i.e., the first uploader who is the target of the investigation. After the first uploader publishes the file, $n$ downloaders immediately download the file from him. Upon completing the transfer, the file is disseminated to $n$ new users by each of the $n$ original users. As a result, $n^2$ users (excluding the first uploader) obtain the file after two rounds.

The following assumptions are made for reasons of simplicity: (i) the download and upload speeds are steady and equal to 500 Kbps; (ii) the file to be transferred has a size of 10 MB; and (iii) the number of simultaneous uploads ($n$) for a Foxy client is five as suggested by the default maximum upload slots for Shareaza [8] (another G2 client). Figure 1 shows the hypothetical file distribution ratio based on these parameters. During the initial round of downloading, all five peers complete the file download in 160 seconds. Afterwards, the download time is shorter as

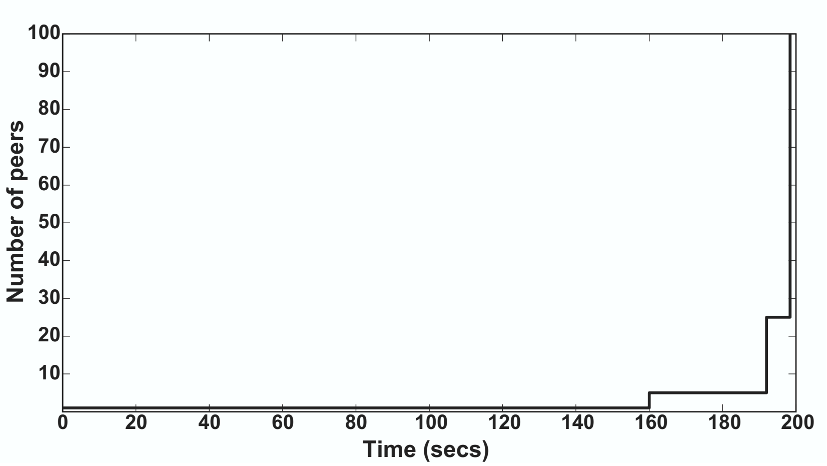

*Figure 1.* Number of peers possessing a file during the first 200 seconds.

more peers upload the file and the number of peers possessing the file increases exponentially. Therefore, it is extremely difficult to identify the first uploader of a shared file after the initial download burst.

## 6. Proposed Investigative Process

Table 3 summarizes the experimental results and the hypothetical behavior of the Foxy network. According to our findings, leaf nodes receive filtered ($O_7$) and frequently-issued queries ($O_8$, $O_9$) from connected ultrapeers. New and popular keyword searches dominate the queries received by clients for similar hash-matched files ($O_{10}$, $O_{11}$).

Publication of file keywords for the Foxy network is usually performed external to the network ($H_1$). Therefore, the publication of new files by the first uploader is rarely detected. However, shortly after the keywords are published, the first batch of downloaders search for and download the file from the first uploader before the file is broadcasted to other users. Based on Hypotheses $H_2$ and $H_3$, the search and download processes initiate two bursts of network traffic: search query traffic inside the Foxy network and file transfer traffic outside the Foxy network.

Digital forensic investigations of suspected uploaders in the Foxy network can only be conducted after identifying their IP addresses. According to $H_3$, the Foxy network has to be monitored and packets have to be captured before the download burst. As long as the IP address of the uploading peer is identified before the file is widely broadcast, the chances of identifying the first uploader are comparatively higher.

*Table 3.* Summary of experimental findings and hypothesized behavior.

| Label | Description |
|---|---|
| $O_1$ | Foxy clients connect to ultrapeer nodes provided by the GWC server. |
| $O_2$ | Each leaf node randomly connects to no more than 32 ultrapeer nodes. |
| $O_3$ | Each ultrapeer connects to no more than 200 leaf nodes. |
| $O_4$ | For each reconnection session, a leaf node connects to different sets of ultrapeers. |
| $O_5$ | Newly-shared file with an uncommon file name can be difficult to find in the file searching process. |
| $O_6$ | Set of query hits returned is affected by connected and neighboring ultrapeers. |
| $O_7$ | Queries from leaf nodes are regulated and filtered at ultrapeer nodes. |
| $O_8$ | Leaf nodes without any shared files also receive queries from neighboring ultrapeers. |
| $O_9$ | Plaintext queries and query hash values are received at leaf nodes. |
| $O_{10}$ | *Q2* packets can arrive at leaf nodes even if they do not possess the file for sharing. |
| $O_{11}$ | *Q2* packets with the relevant hash values dominate the Foxy network shortly after an attractive keyword is announced. |
| $O_{12}$ | Percentage of the frequently-asked query gradually decreases after the majority of Foxy users have downloaded the file. |
| $H_1$ | No observable changes are induced during or shortly after the announcement of a keyword. |
| $H_2$ | Number of *Q2* packets querying a keyword increases after the announcement of a keyword. |
| $H_3$ | Number of peers possessing the newly-shared file increases exponentially after the announcement of a popular keyword. |

Monitoring search query traffic is much more useful than attempting to monitor the downloading of a file that is the subject of an investigation. File transfer occurs via a direct connection between peers; therefore, only the peers involved in the file transfer can identify the IP address of the uploader. On the other hand, query packets (*Q2*) and query hit packets (*QH2*) in search query traffic contain the requester's IP address, ultrapeer's IP address, query hash value or plaintext query keywords, real name of the file, and most importantly, the IP address from where the file can be downloaded.

Figure 2 provides an overview of the simplified Foxy investigation model. In a normal situation (Figure 2(a)), queries received by Foxy clients have different patterns. During a burst (Figure 2(b)), the same queries are received, originating from different leaf nodes. The monitoring nodes use the same query and submit it back to the Foxy network (Figure 2(c)). When the query arrives at the uploader, it returns the full name of the file and its IP address in the QH2 packet (Figure 2(d)).

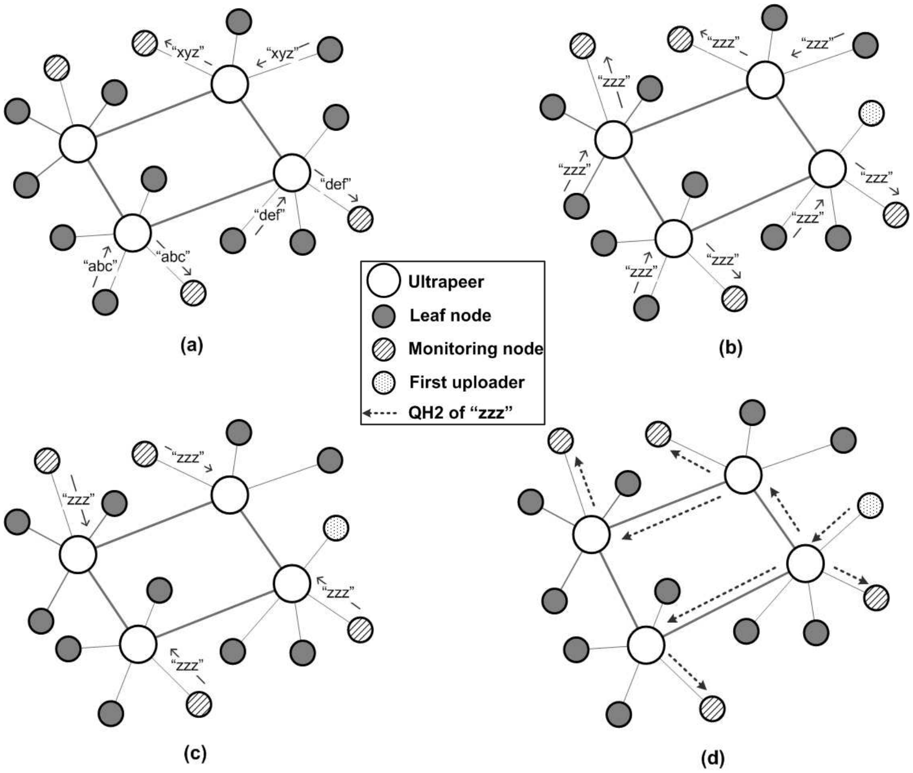

*Figure 2.* Simplified Foxy investigation model.

This enables us to specify the following investigative process using a customized client:

- Connect to the Foxy network and collect all *Q2* packets from the ultrapeers.
- Re-pack and submit the identified query when a large number of *Q2* packets querying the same keyword are observed.
- Extract the source IP address and the matched file name from all collected *QH2* packets for analysis.
- Create a list of source IP addresses and determine the frequency of the returned IP addresses. The shorter the period between the initial identification of the burst and the generation of the list, the greater the likelihood of locating the first uploader.

## 7. Conclusions

Identifying the IP addresses of P2P clients is crucial to P2P network investigations. However, it is extremely difficult to identify the first uploaders of files in the Foxy network. The investigative methodology presented in this paper leverages query packets and IP addresses to help identify uploaders in the Foxy network. The strategy is derived from observations and findings based on captured Foxy network packets. Because the Foxy network uses the G2 network protocol with minor modifications, the strategy should be applicable to general G2 network investigations.

Building on this work, we are customizing a Foxy client to analyze search behavior in an isolated experimental environment. Actual and simulated results of G2 network searches and file downloads will be collected to refine the investigative process. In addition, we plan to extend the investigative process to other "search-based" P2P networks.

## References

[1] Architecture Technology Corporation, P2P Marshal Digital Forensics Software, Eden Prairie, Minnesota (p2pmarshal.atc-nycorp .com).

[2] E. Bangeman, P2P responsible for as much as 90 percent of all 'Net traffic, *Ars Technica*, September 3, 2007.

[3] BBC News, BitTorrent user guilty of piracy (news.bbc.co.uk/1/hi /technology/4374222.stm), October 25, 2005.

[4] J. Cheng, Sandvine: Close to half of all bandwidth sucked up by P2P, *Ars Technica*, June 23, 2008.

[5] M. Chesterton, Edison Chen and 7 HK stars involved in sex photos scandal, *eNews 2.0*, February 21, 2008.

[6] 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.

[7] 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, University of Hong Kong, Hong Kong, 2008.

[8] Discordia, Shareaza, New York (www.shareaza.com).

[9] Gnutella2, Gnutella2 Developer Network (g2.trillinux.org).

[10] R. Ieong, P. Lai, K. Chow, M. Kwan, F. Law, H. Tse and K. Tse, Forensic investigation and analysis of peer-to-peer file-sharing networks (submitted for publication), 2009.

[11] P. Moy, Warning over rape clips, *The Standard*, Hong Kong, September 12, 2008.

[12] P. Moy and N. Patel, Covert cops hit by leaks, *The Standard*, Hong Kong, May 27, 2008.

[13] O. Nasraoui, D. Keeling, A. Elmaghraby, G. Higgins and M. Losavio, Node-based probing and monitoring to investigate the use of peer-to-peer technologies for the distribution of contraband material, *Proceedings of the Third International Workshop on Systematic Approaches to Digital Forensic Engineering*, pp. 135–140, 2008.

[14] Vastel Technology, Foxy, Hong Kong (www.gofoxy.net).

[15] Wireshark Foundation, Wireshark, San Jose, California (www.wire shark.org).

[16] Zemerick Software, Spear Forensics Software, Oak Hill, West Virginia (www.spearforensics.com/products/forensicp2p/index.aspx).