Omnify: Investigating the Visibility and Effectiveness of Copyright Monitors

Rahul Potharaju[‡], Jeff Seibert[‡], Sonia Fahmy, and Cristina Nita-Rotaru
Purdue University
{rpothara, jcseiber, fahmy, crisn}@cs.purdue.edu

Abstract. [1]
The arms race between copyright agencies and P2P users is an ongoing and evolving struggle. On the one hand, content providers are using several techniques to stealthily find unauthorized distribution of copyrighted work in order to deal with the problem of Internet piracy. On the other hand, P2P users are relying increasingly on blacklists and anonymization methods in order to avoid detection. In this work, we propose a number of techniques to reveal copyright monitors' current approaches and evaluate their effectiveness. We apply these techniques on data we collected from more than 2.75 million BitTorrent swarms containing 71 million IP addresses. We provide strong evidence that certain nodes are indeed copyright monitors, show that monitoring is a world-wide phenomenon, and devise a methodology for generating blacklists for *paranoid* and *conservative* P2P users.

1 Introduction

Peer-to-peer (P2P) applications possess fundamental advantages over the traditional client-server model and fixed-infrastructure content distribution networks like Akamai. Specifically, P2P systems offer increased performance, availability, and scalability by leveraging resources (e.g., bandwidth, storage, and computing power) contributed by each peer. As a result, P2P systems enable a wide range of services such as data sharing, voice-over-IP (VoIP), and video streaming. Popular applications that use P2P systems include file sharing systems such as BitTorrent [1] and Gnutella [2], VoIP systems such as Skype [3], and video streaming systems such as PPLive [4]. Several studies have indicated that P2P systems contribute towards more than 60% of the overall network traffic [5].

Users exchange content via P2P file sharing networks for many reasons, ranging from the legal exchange of open source software to the illegal exchange of copyrighted material. The latter activities, however, are perceived as a threat to the business models of the copyright holders. To protect their content, copyright holders monitor P2P networks and the sharing behavior, collecting evidence of infringement, and then issue any infringing user a notice. In the United States, this notice is called a *Digital Millennium*

[1] Entry for PAM Award: This paper contributes a novel *Reverse Infohash* database containing more than 1.75 million infohash → title mappings. The dataset and more information can be obtained at: http://omnify.info/

[‡] Both authors contributed equally.

Copyright Act (DMCA) [6] takedown notice. The notices are formal requests to stop sharing particular data, and are typically sent to the authorities responsible for the IP addresses of the infringing users. These authorities then forward these notices to the respective users inside their network. Unfortunately, this simple approach of monitoring is prone to a wide variety of errors. Piatek *et al.* [7] describe techniques for implicating arbitrary network endpoints in illegal content sharing, and demonstrate their effectiveness by experimentally attracting real DMCA complaints for devices such as IP printers and wireless access points.

Our work was motivated by our recent experience of being mistakenly implicated for copyright infringement, when in fact we were performing performance measurement experiments on the PlanetLab testbed [8], an overlay network for developing and accessing a broad range of network services. This problem can be partly solved by using existing blacklists [9], consisting of IP subnet ranges of clients suspected of monitoring activities. However, blacklists such as those constructed by iBlocklist [9] use help from various user communities and no empirical research exists to prove their integrity or effectiveness. In this paper, we derive a methodology for constructing different types of blacklists. It is important to note that we are *not* encouraging unlawful sharing of copyrighted material, but rather showing that the patterns we reveal can be useful in designing conservative research experiments, as such experiments are critical for improving the BitTorrent ecosystem. Our results can also be leveraged by copyright monitors to improve their detection accuracy, thus enabling them to reduce the rate of false positives.

Contributions: Our work reveals a number of findings, confirming known types of undesirable behavior in the BitTorrent network, and uncovering new patterns that provide strong evidence of monitoring. Our contributions include:

- We develop a systematic methodology for obtaining the file name for a given infohash (a SHA-1 identifier for a BitTorrent file that is globally unique), thereby constructing the first *Reverse Infohash* database containing mappings for 1.18 million infohashes. This dataset has several applications such as understanding the extent of swarm redundancy (how many swarms share the same file), and current trends in file sharing. We are making this data publicly available for the benefit of other researchers.
- We introduce the Ω-factor that utilizes the BitTorrent crawls we collected from more than 2.75 million BitTorrent swarms and the *Reverse Infohash* database to distinguish normal peers from suspicious ones in a P2P system. We then leverage the Ω-factor to identify copyright monitors.
- We reveal a number of patterns in the activity of different hosts (identified by IP addresses), which we use to establish strong evidence supporting the existence of copyright monitors. For instance, we show instances when certain IP addresses participate in hundreds of swarms serving the same file. Leveraging these observations and the Ω-factor, we design and present the methodology for creating two blacklists of suspicious IP addresses: *paranoid*, useful for users who are privacy-conscious and *conservative*, useful for users who are more lenient.

The rest of this paper is organized as follows: Section 2 provides relevant background on our data collection process and the methodology for constructing the *Re-*

verse Infohash database. We then use this database to calculate the participation extent of each IP address using the novel metric we introduce called the Ω-factor. We use two threshold schemes to derive two blacklists. In Section 3, we examine the blacklists in detail and give insights on the effectiveness of copyright monitors. Finally, we summarize related work in Section 4 and conclude in Section 5.

2 Methodology

We seek to study peer activity patterns to identify possibly atypical behavior, with the ultimate goal being to derive blacklists consisting of IP addresses belonging to suspicious clients. To achieve this, we rely on measuring the extent of participation of each IP address in swarms so that we can filter out normal user clients from the list of active IP addresses. We make the key observation that a normal client typically participates in at most one swarm to download a certain file, whereas a copyright monitor would want to participate in as many swarms (that serve the same file) as possible to increase its effectiveness. In order to determine whether multiple swarms are serving the same file, we utilize our *Reverse Infohash* database to groups similar swarms together. We then, to distinguish normal clients from those exhibiting abnormal behavior, introduce the Ω-factor that captures these groupings and the activity pattern of a client.

We first present the background on our data collection methodology. Next, we explain how we construct the *Reverse Infohash* database and use this database to calculate the Ω-factor for each IP address in our dataset. Finally, we use two threshold schemes on the Ω-factor to derive two blacklists.

2.1 Data Collection

The methodology oftentimes used for crawling BitTorrent swarms is to find torrent metafiles by first crawling a BitTorrent aggregator website that hosts them, and then querying the associated tracker for clients [7, 10]. However, as aggregator websites usually only attract users associated with a particular language, this technique would result in only revealing copyright monitoring in certain parts of the world. Therefore, we do the *reverse*: we first crawl a tracker that is not associated with an aggregator website and is highly popular all over the world, called "OpenBitTorrent" [11], and afterwards we discover the content that the swarms we find are serving. We note that while we use a particular tracker, our methodology is extensible to multiple trackers.

When crawling the tracker, we extract a list containing the infohash and number of seeders and leechers for every swarm it is tracking. In BitTorrent lexicon, the infohash is a 20-byte SHA-1 hash of information contained in a `.torrent` metafile. In May 2010, for every hour over 8 days, we obtained this list and crawled each swarm that had at least one leecher. Typically, a single crawl consists of over 5 million IP addresses in one million swarms. We discovered 71 million unique IP addresses in 2.75 million swarms over the 8-day period.

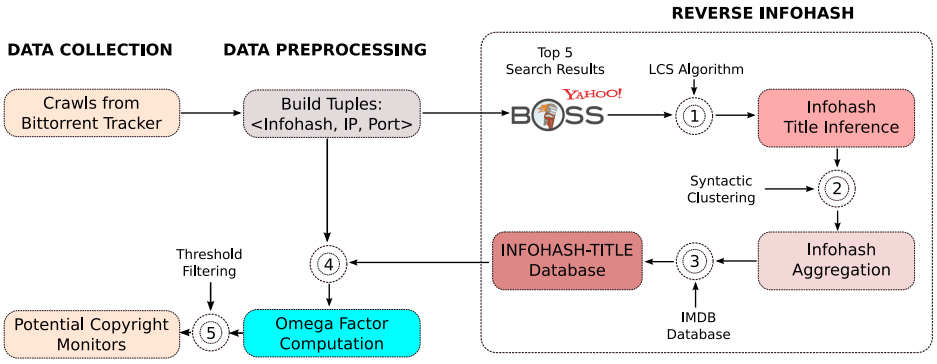


Fig. 1. Schematic overview of the proposed methodology: Starting from our BitTorrent crawls, we use a number of techniques to derive the *Reverse Infohash* database which is used for computing Ω, a metric for distinguishing a normal peer from undesirable peers.

2.2 Building the Reverse Infohash Database

We now focus on finding the file name associated with an infohash and then group similar file names together. We apply a number of heuristics to construct the *Reverse Infohash* database. Figure 1 gives an overview of how we accurately determine the file name associated with an infohash. Our initial step is to query a search engine for the infohash, giving a rough approximation of the file name. As Google rate-limits search queries, we use the Yahoo! Search BOSS API [12], which imposes less stringent constraints on their API usage. We consider the top five results returned by each search query, and use the *Longest Common Subsequence*(LCS) algorithm [13] for constructing a single title using the five search results. Our heuristic performs an exhaustive comparison to find the best possible match. For instance, the algorithm outputs "Iron Man 2 D" for an input of {"Iron Man 2 DivX", "Iron Man 2 DVD", "Iron Man 2 DirectSubs"}.

At this stage, we leverage the Shingling technique [14] to determine the syntactic similarity of the strings in our database. Intuitively, this method helps us determine if two strings are "roughly the same" *i.e.*, for determining when they have the same content except for modifications such as lost characters etc. We view each string in our database as a set of subsequences of tokens $T(S, w)$. A contiguous subsequence contained in S is called a *shingle*. Given a string S, we define its *w-shingling* $T(S, w)$ as the set of all unique shingles of size w in S. For instance, the *2-shingling* or the character bigram shingles of the string "*iron man*" is the following set: {"ir", "ro", "on", "n ", " m", "ma", "an"}.

Given two strings, we build a *shingling* set for each, and then use a distance metric to measure the similarity. As a distance metric, we use the Jaccard Index [15] which is a measure used for comparing the similarity and the diversity of sample sets. It is the ratio of the size of the intersection of the sets to the size of the union of the sets. For instance, if $J(A, B)$ is the Jaccard Index between sets A and B, then:

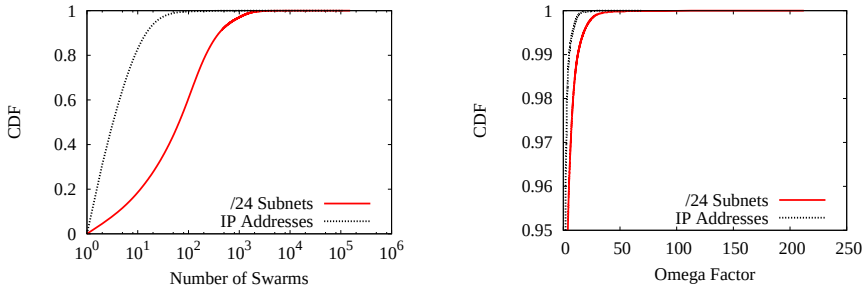
$$J_{(A,B)} = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

We build a similarity index represented by an adjacency matrix for the list of strings in our database, and use a threshold of 0.8 (determined through manual inspection and represents only those strings which are nearly similar to each other) to prepare the final list. At this stage, we have reduced the string grouping problem to an instance of finding *connected subgraphs* [16]. From the result, we can determine if two different infohashes served the same *entity*. We define *entity* as all torrents sharing the same file. For instance, all torrents sharing Iron Man 2 constitute a single *entity*. To verify the accuracy of our methodology, we manually checked 1330 infohashes which were suspected to be serving the same file, "Iron Man 2." We searched Google for information related to an infohash and matched it with ours. We were able to manually match 1265 out of the 1330 (≈95.1% accuracy). The false positives were for entries such as "Irina," a fictional vampire character in the Twilight Saga.

2.3 Identifying Activity Patterns

Our ultimate goal is to distinguish normal P2P users from copyright monitors. Due to the vast number of IP addresses we collected, we do not attempt active measurements or probing of clients, but rely solely on analyzing the data collected. Since oftentimes the IP address, port number, and AS number are not revealing, we must identify patterns that would be indicative of an organization monitoring copyrighted material. We now outline a list of identifiable patterns and why we consider them.

An IP address participates in a large number of swarms: The vast majority of IP addresses seen in our crawls participate in very few swarms. To demonstrate this, we plot in Figure 2(a) a CDF of the number of swarms every IP address participates in (over our eight-day crawl). For example, 84% of IP addresses participate in 10 swarms or fewer. However, some IP addresses appear in hundreds, and even thousands, of swarms. We suspect that since monitors aim to observe the transfer of copyrighted material in the BitTorrent community, they would participate in every relevant swarm.



(a) (b)

Fig. 2. (a) CDF of the number of swarms every IP address participates in over our eight-day crawl. Observe that 84% of the IP addresses participate in 10 swarms or fewer. (b) CDF of the Ω-factor for IP addresses and subnets. Most IP addresses have $\Omega = 1$, meaning they only participate in a single swarm corresponding to a particular *entity*.

An IP address participates in multiple swarms that correspond to the same *entity*: As mentioned earlier, we discovered over a thousand swarms that corresponded to the same "Iron Man 2" movie. We would expect a normal user, if desiring to download such

a movie, to simply participate in one of these swarms. In contrast, a copyright monitor would participate in many swarms, attempting to observe as many downloaders as possible by utilizing all its available resources. Thus, we introduce the Ω-factor, which intuitively measures how effective a host (with a certain IP address) is at monitoring entities, carefully adjusting for entities with a fewer/larger number of swarms. We define the Ω-factor as follows:

$$\Omega_{IPAddress} = \frac{\sum_{i=0}^{n} t_i s_i}{\sum_{i=0}^{n} t_i} \tag{2}$$

where t_i is the total number of torrents corresponding to *entity* i, and s_i is the total number of torrents corresponding to *entity* i where the IP address is actually seen. Let us consider an example of a user downloading "Iron Man 2" and "Valentine's Day" from two swarms. In our dataset, the number of distinct torrents we observed for the former swarm was 1526 and for the latter, 431. Thus, $\Omega = \frac{1526(1)+431(1)}{1526+431} = 1$. A copyright monitor, in contrast, typically aims to monitor multiple torrents corresponding to an *entity*. For the same entities, assuming that the monitor participates in 120 swarms related to "Iron Man 2" and 100 swarms related to "Valentine's Day", we would get $\Omega = \frac{1526(120)+431(100)}{1526+431} = 115.59$. Thus, by using the number of swarms an IP address is participating in as the weight for the total number of torrents corresponding to an *entity*, we are able to detect the outliers.

We evaluate the Ω-factor for every IP address that appears in our database and present the results in Figure 2(b). Unsurprisingly, most IP addresses have a factor of 1, meaning they only participate in a single swarm corresponding to a particular *entity*. However, there are several outliers that have values greater than 40, implying suspicious behavior.

2.4 Generating Blacklists

We now leverage the values of the Ω-factor to derive two blacklists. The first blacklist, called the *paranoid* dataset, is useful for users who are privacy-conscious and do not want to be bothered by any kind of a monitor including a copyright monitor, spambot, or researchers conducting measurements. The second blacklist we constructed, called the *conservative*, is a restrictive subset of the *paranoid* dataset that comprises IP addresses that are all highly likely copyright monitors.

The *paranoid* dataset is obtained by applying a threshold scheme on the Ω factor, and then filtering out all the IP addresses that participate in fewer than 100 distinct swarms. We choose this threshold as most IP addresses participate in fewer than 100 swarms; otherwise, they would seem ineffective as copyright monitors. From Figure 2(b), we chose Ω to be 5.0, therefore including many suspicious IP addresses. The total number of IP addresses in this blacklist is 53,752. The *conservative* dataset is obtained in a similar manner by setting Ω to 20.0 to only select highly suspicious clients participating in more than 100 distinct torrents. The total number of IP addresses in this blacklist is 5,719 – much smaller than *paranoid*. We have verified that these IP addresses do not belong to a known botnet and are not in a spam database.

3 Evaluation

In this section, we show that using our methodology, we can automatically generate blacklists that are able to effectively identify copyright monitors. We also identify interesting characteristics of these monitors and discuss how they can avoid detection in the future.

Identifying Monitors To evaluate if the *conservative* dataset accurately reflects behavior consistent with a copyright monitor, we manually verified the 100 IP addresses with the largest values of Ω. We provide a snapshot of the details of the entities they are monitoring in Table 1. In this table, we provide the AS number and country where the IP address is located, and also the entities that they are heavily monitoring. We determine their *effectiveness* at monitoring copyright infringement by calculating the percentage of swarms that they participate in that correspond to a particular *entity*.

For example, in the first entry in the table, we find that a particular IP address in AS-9167 is heavily monitoring swarms related to the *"Percy Jackson and the Olympians: The Lightning Thief" entity*, which is a popular movie distributed by 20th Century Fox. Using our reverse infohash database, we map 310 different torrents to this *entity*. We observed that this IP address in particular monitors 174 different torrents, giving it an effectiveness of 55.10%. While we do not dig into the details of the other ASes, we show a few more in Table 1.

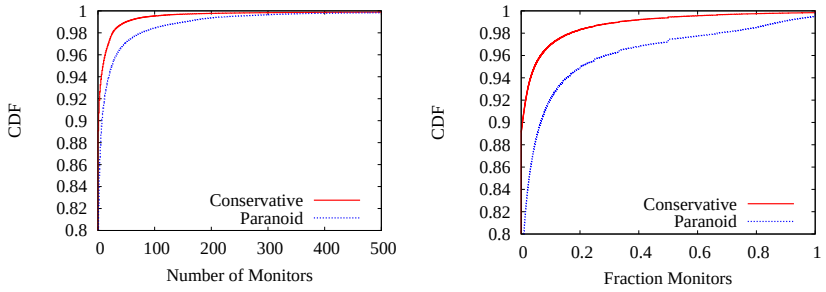
AS Number	Effectiveness %	Country	Entity
9167	55.10	Denmark	Percy Jackson & the Olympians
9167	53.54	Denmark	Percy Jackson & the Olympians
9167	36.5	Denmark	Alvin & the Chipmunks 2 The Squeakquel
33650	84.94	United States	Alex Jones Show
1213	5.4	Ireland	Iron Man 2
30023	17.45	United States	Iron Man 2
30023	15.65	United States	Iron Man 2, Princess and The Frog
30023	6.26	United States	Iron Man 2, Valentine's Day
558	5.81	United States	Iron Man 2, Valentine's Day
9167	49.27	Denmark	Jennifer's Body

Table 1. Using the Ω factor to pinpoint a copyright monitor: The AS numbers shown in this table had IP addresses that exhibited high Ω factors. Upon further inspection, they were found to participate in a large number of torrents serving the same *entity*.

Pervasiveness of Monitoring For every swarm we find how many monitors are participating and also the fraction of monitors in that swarm and plot CDFs of these values in Figure 3(a) and Figure 3(b), respectively. We find that 11% (20%) of the 2.75 million swarms are being monitored in the conservative (paranoid) dataset. We also find that sometimes hundreds of monitors are used and that in 0.6% (3%) of swarms the majority of participants are monitors.

Geolocating Monitors We geolocate the IP addresses in our *conservative* dataset using the MaxMind database [17], and plot the results in Figure 4. As can be seen, most countries in Europe are densely populated with monitors. In addition, several Asian countries, such as Japan, South Korea, and the Philippines are densely populated. While we are not legal experts, we suspect that copyright and privacy laws in each country highly influence the number of monitors there.

Monitors Exhibit Temporal Patterns From the *paranoid dataset*, we find that copyright monitors do indeed own subnets and use them in interesting ways. To illustrate



(a) (b)

Fig. 3. CDF of the number and fraction of IP addresses classified as monitors participating in each swarm in our eight-day crawl.

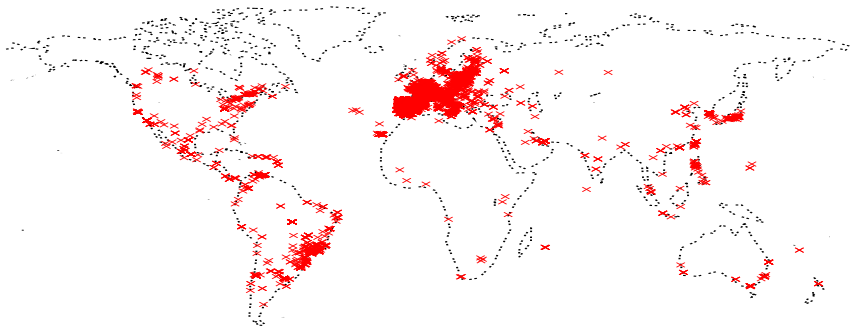
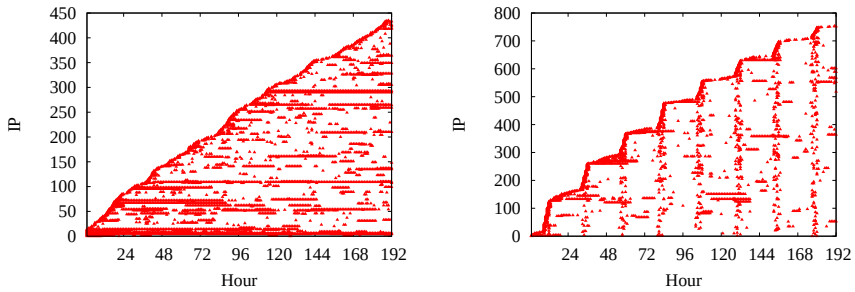


Fig. 4. Map showing the geolocated positions of all IP addresses from our *conservative* dataset.

this, we choose two suspected monitors that have multiple subnets, and give each IP address an identifier in the order of the first time it is observed in the crawls. We plot a point for every crawl where we observe the IP address. The results are depicted in Figure 5. As there is large overlap in the swarms each IP address is monitoring, we suspect that this type of pattern can be attributed to an automated monitor that, when crawling, uses a different subset of IP addresses every time. For example, the monitor in Figure 5(a) seems to use several different IP addresses every hour, while the monitor in Figure 5(b) has a more diurnal pattern, using approximately 100 different IP addresses to crawl once a day.



(a) (b)

Fig. 5. An instance when multiple IP addresses belonging to the same subnet exhibit similar activity patterns.

Spammers Utilize BitTorrent We note that the *paranoid* dataset captures behavior beyond that of a copyright monitor. We found several consecutive subnets in AS-5384 which contained IP addresses that were participating in over 60,000 swarms, which were some of the most often seen. We checked Project Honey Pot [18], which is a database of known email spammers, and found that many of them were contained in that list. We suspect that these clients are harvesting information about BitTorrent users. Thus, while they are not necessarily copyright monitors, we believe most that privacy-conscious users would prefer to avoid contact with such clients.

Comparison with Real-World Blacklists We investigate several blacklists of IP addresses that are suspected to be owned by copyright monitors. These are collected from an aggregator of blacklists [9] and include IP addresses of organizations that are involved with trying to stop file sharing (e.g., MediaDefender, MediaSentry). These blacklists are P2P community-driven and accepted; however, they are usually only extended after users have received a DMCA takedown or similar notice. We find that our *conservative* and *paranoid* datasets overlap with 2,051 and 2,507 IP addresses in these blacklists, respectively. This indicates that algorithmic techniques for determining copyright monitors can be effectively used to supplement existing blacklists.

Improving Stealth and Effectiveness of Monitors We now present a few suggestions for copyright monitors that can improve their stealth and effectiveness. First, they can use our methodology for finding torrents and not rely on crawling aggregator websites. This increases their effectiveness, because they will be able to find more entities. Second, to improve their stealth, they must utilize more hosts (IP addresses) and intelligently use them by distributing among them the swarms that belong to the same *entity*. This will obscure them from most pattern-mining algorithms. Third, using better measurement techniques by leaving a swarm immediately after getting the list of peers, their IP addresses can remain below a detection threshold that relies on hours of activity or the number of times an IP is seen.

4 Related Work

Copyright monitors and the techniques they use to infer copyright infringement in BitTorrent have been studied by Piatek *et al.* [7]. They discover that copyright monitors use indirect and thus inconclusive evidence to serve DMCA takedown notices, and show that it is possible to frame arbitrary IP addresses of infringement. The authors also crawl popular torrent aggregation sites and crawl tens of thousands of swarms. They evaluate the potential of blacklists to identify copyright monitors, and find that there are suspicious IPs participating in many swarms. Our work takes this a step further by crawling millions of swarms, identifying suspicious activity patterns, and narrowing down the list to the most suspicious clients.

Deviant client behavior in BitTorrent swarms was studied by Siganos *et al.* [10]. They implement a client that exchanges control messages with other BitTorrent clients to discover if they exhibit suspicious behavior. They crawl 600 popular torrents over 45 days, and find behavior indicative of copyright monitors and peers involved in botnets. In contrast, our work introduces metrics that can be used to infer copyright monitor behavior.

5 Conclusions

In this paper, we have reported our findings on the effectiveness of monitoring agencies in the wild. We constructed a novel reverse infohash database which we used in computing what we refer to as the Ω-factor, a measure that differentiates entities in a P2P system, to shortlist potential copyright monitors. We applied our techniques on data we collected from more than 2.75 million BitTorrent swarms containing 71 million IP addresses, and discussed our methodology for arriving at a list of potential copyright monitors. In particular, we prepared two datasets: a *paranoid* dataset that contains a list of copyright monitors along with hosts that could be potential spambots, and a *conservative* dataset that contains hosts that are suspected to be copyright monitors with a high probability.

References

1. B. Cohen, "Incentives build robustness in BitTorrent," in *Workshop on Economics of Peer-to-Peer systems*, 2003.
2. R. Matei, A. Iamnitchi, and P. Foster, "Mapping the Gnutella network," *Internet Computing, IEEE*, vol. 6, no. 1, pp. 50–57, 2002.
3. S. Baset and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," *Arxiv preprint cs/0412017*, 2004.
4. Y. Huang, T. Z. Fu, D.-M. Chiu, J. C. Lui, and C. Huang, "Challenges, design and analysis of a large-scale p2p-vod system," in *SIGCOMM*, 2008.
5. K. Cho, K. Fukuda, H. Esaki, and A. Kato, "Observing slow crustal movement in residential user traffic," in *CONEXT*, 2008.
6. A. Kao, "RIAA v. Verizon: Applying the Subpoena Provision of the DMCA," *Sch. L. Rev*, vol. 63, p. 64, 2003.
7. M. Piatek, T. Kohno, and A. Krishnamurthy, "Challenges and directions for monitoring P2P file sharing networks–or–Why my printer received a DMCA takedown notice," in *3rd USENIX Workshop on Hot Topics in Security (HotSec)*, 2008.
8. B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "Planetlab: an overlay testbed for broad-coverage services," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 3–12, 2003.
9. "TBG Blocklist." `http://tbg.iblocklist.com/`.
10. G. Siganos, J. M. Pujol, and P. Rodriguez, "Monitoring the bittorrent monitors: A bird's eye view," in *PAM*, 2009.
11. OpenBittorrent. http://www.openbittorrent.com.
12. "Yahoo! Search Boss." `http://developer.yahoo.com/search/boss/`.
13. D. Hirschberg, "Algorithms for the longest common subsequence problem," *Journal of the ACM (JACM)*, vol. 24, no. 4, pp. 664–675, 1977.
14. A. Broder, S. Glassman, M. Manasse, and G. Zweig, "Syntactic clustering of the web," *Computer Networks and ISDN Systems*, vol. 29, no. 8-13, pp. 1157–1166, 1997.
15. P. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Pearson Addison Wesley Boston, 2006.
16. S. Khuller, "Approximation algorithms for finding highly connected subgraphs," *Approximation algorithms for NP-hard problems*, pp. 236–265, 1996.
17. L. MaxMind, "GeoIP," 2006.
18. "Project Honey Pot." http://www.projecthoneypot.org.