Angling for Big Fish in BitTorrent

Stevens Le Blond, Arnaud Legout, Fabrice Le Fessant, Walid Dabbous

Angling for Big Fish in BitTorrent

Stevens Le Blond, Arnaud Legout, Fabrice Le Fessant and Walid Dabbous
I.N.R.I.A., France

Contact: {stevens.le_blond, arnaud.legout, fabrice.le_fessant, walid.dabbous}@inria.fr

ABSTRACT

BitTorrent piracy is at the core of fierce debates around network neutrality. Most of the legal actions against BitTorrent exchanges are targeted toward torrent indexing sites and trackers. Surprisingly, little is known about the initial seeds that insert contents on BitTorrent and about the highly active peers that are present in a large number of torrents. The main reason is that acquiring this knowledge requires a large-scale continuous crawl of BitTorrent, which is believed to be impractical. However, this information is important for scientist and politics as many unfounded claims are made about BitTorrent piracy.

In this paper, we present a crawl dedicated to initial seeds identification and a large-scale continuous crawl of 103 days during which we collect 148M IP addresses of peers participating in 1.2M of torrents. We present the first in-depth analysis of initial seeds' behavior and of highly active peers. We show that it is possible to identify initial seeds for 70% of torrents, that initial seeds form a small community, and that some of the most active initial seeds are in hosting centers, which make the identification of the location of the human being running those initial seeds complex. In addition, we identified among the highly active peers very different categories including anti-piracy groups and VPNs like iPREDator. We also confirmed that Tor is inefficient for BitTorrent distribution and that VPN solutions are used by well-provisioned peers. Finally, we found an issue affecting initial seeds in the way torrents are queued in BitTorrent clients.

1. INTRODUCTION

"Should one's Internet access be closed for downloading copyrighted material?" This is the key question faced by the European parliament in 2009 when dealing with the three-strikes law proposal [1]. Indeed, many governments and artists associations are fighting copyright infringements on peer-to-peer networks with a clear focus on BitTorrent. For instance, The Pirate Bay, the largest BitTorrent tracker [11], has been stopped in August 2009 following a court order.

Torrent indexing sites and public trackers are the targets of most of the prosecutions because they are the visible parts of the piracy on BitTorrent. However, surprisingly, little is known about who inserts contents, that we call initial seeds, and who are present in the largest number of torrents, that we call highly active peers. We argue that this knowledge is fundamental for scientists and for politics, as it will shed a new light on current BitTorrent usage.

Indeed, an initial seed is the first peer who seeds a new content in BitTorrent. Those peers are the ones who insert contents, but nothing is known about how many they are, their location, their behavior, their characteristics, and their impact on torrents performance. Highly active peers are also unknown. Finding those highly active peers is necessary in order to understand the actors involved in the BitTorrent ecosystem.

The main reason of this lack of knowledge is that collecting the data needed to acquire this understanding is believed to be impractical [4, 8, 11] because it requires a continuous monitoring of peers on a large fraction of all torrents in the Internet. Indeed, previous studies on BitTorrent monitoring reported a high risk of being blacklisted, i.e., banned from torrents indexing sites or tracker per IP address or prefix of IP addresses, when too aggressive in the monitoring. The proposed solution is usually to distribute monitoring machines, at the expense of a much higher complexity, and to reduce the number of torrents monitored [4, 7, 9] or sacrifice the continuous monitoring in favor of a single snapshot monitoring of a large set of torrents [11]. Thus, none of those studies addressed the issue of continuously monitoring a large set of torrents for a large period of time.

In this paper, we present a continuous monitoring performed from May 13, 2009 to August 24, 2009 every two hours of all peers on all torrents tracked by The Pirate Bay trackers. We collected 148M IP addresses, spread over 21k ASes, participating into 1.2M of torrents, and representing 3.6 exabytes of data potentially downloaded by peers. Also, during 48 days in that period we connected to each new torrent, within the first minute of their insertion on The Pirate Bay Web site, and monitored it for 24 hours.

In addition to the collection of this large trace and to the methodology that allowed us to collect this trace from a single machine, our contribution on the analysis of this trace are the following.

i) We show that it is possible to identify the IP address of initial seeds and to map those initial seeds to users who inserted the contents on The Pirate Bay torrent indexing site for 70% of the users. In particular, the notion of users in BitTorrent sites does not provide any anonymity.

ii) We show that a small community of initial seeds inserts most of the contents we observe in our crawl. In particular, the 1,000 most active initial seeds represent 60% of all the torrents seeded. However, deriving the human beings country from the initial seeds IP address would be misleading. Indeed, a large fraction of the most active initial seeds use

foreign hosting centers.

iii) We show that the queuing strategy used in BitTorrent is suboptimal for initial seeds, and that the upload speed of the initial seeds is critical to the performance of the torrents.

iv) We identify 6 different categories of highly active peers, i.e, peers that are seen in a large number of torrents, that are monitors, HTTP and SOCKS proxies, Tor nodes, VPNs like iPREDator, and heavy peers. In addition, we show that using those categories we can track anti-piracy groups.

v) We confirm the belief that Tor is inefficient for BitTorrent distribution and find that well-provisioned peers use VPN solutions like iPREDator.

vi) We show that a continuous crawl is superior to a single snapshot to identify highly active peers.

In section 2 we present the crawlers used in this study and discuss the representativeness of our crawls. In section 3 we profile initial seeds, and we profile highly active peers in section 4. We present the related work in section 5 and conclude in section 6.

2. LARGE-SCALE CONTINUOUS MONITORING

In this section, we justify the choice of the torrent indexing sites and trackers we decided to monitor. Then, we describe the crawler of initial seeds and the large-scale crawler that are used respectively to identify and monitor initial seeds, and to perform a large-scale continuous crawl of a large fraction of the torrents in the Internet. However, those two crawlers work on overlapping data, in particular the .torrent files and meta data information, and run in parallel.

In the following, we refer to the notion of infohash as a unique identifier of a torrent. The infohash is a hash of the info_key field of the .torrent file.

2.1 The BitTorrent Ecosystem

The BitTorrent ecosystem consists of *torrent indexing sites*, *trackers*, and peers [11]. The focus of this work is to explore the behavior of specific peers, in particular initial seeds and highly active peers, by monitoring torrent indexing sites and trackers.

Torrent indexing Web sites maintain a database of meta data on torrents including for each torrent the torrent name, a link to the .torrent file, the login of the *user* who inserted the meta data on the torrent indexing site, comments on the torrents, etc. Those sites can be public, i.e., they require no authentication, or private, i.e., they require credentials to log into the site. The public torrent indexing sites represent the largest community, and we specifically focus on those sites. In particular, we consider the three most popular public torrent indexing sites in English that are mininova, The Pirate Bay, and IsoHunt. However, those sites are not restricted to torrents in English. Indeed a significant fraction of the torrents are in other languages, in particular European and Asian languages. Moreover, as we collected 148M unique IP addresses in the large-scale crawl described in section 2.3, we deem that we already cover a representative fraction of all Internet users. Zhang et al. [11] show that the redundancy of those sites is high, and therefore taking a few of them is enough to cover most of the torrents.

The trackers are dedicated servers maintaining a list of active peers for each torrent registered to those trackers. The Pirate Bay tracker is by far the largest tracker with more than 10M of peers and 1M of torrents. The second largest BitTorrent tracker is one order of magnitude smaller in number of peers and torrents tracked [11]. In this study we specifically focus on The Pirate Bay tracker. This tracker was stopped on August 24, 2009 due to a legal action on its bandwidth provider. By that date, we had already monitored The Pirate Bay tracker for more than 100 days, which was enough for our study. Therefore, we decided to stop the collection of data when the tracker was stopped, instead of moving to another tracker.

2.2 Initial Seeds Crawler Description

The goal of this crawler is to detect and monitor initial seeds. To achieve this goal, we implemented a crawler that connects to each new torrent inserted on The Pirate Bay Web site and monitors that torrent during 24 hours. We ran this crawler continuously for 48 days, between July 8, 2009 and August 24, 2009. We describe in the following the details of this crawler.

The Pirate Bay Web site maintains a page of newly inserted torrents. Our crawler retrieves this page every minute and, most importantly, joins each of the newly added torrents it has not yet joined. Then, the crawler acts as a regular BitTorrent client. In particular, it exchanges signaling with neighbors, downloads pieces, and checks the integrity of those pieces using the cryptographic SHA-1 hashes contained in the .torrent file, which is used to fingerprint each piece. However, the downloaded pieces are never written on disk; once the SHA-1 is verified the piece is discarded. In addition, the crawler saves the IP address of all its neighbors, and all HAVE and BITFIELD received messages.

When the crawler joins a torrent, it asks for 200 neighbors to the tracker, which is the maximum number of neighbors returned by The Pirate Bay tracker during our experiments. As the tracker keeps state on all peers subscribed to each torrent, it is able to detect peers connected to an abnormally large number of torrents and blacklist their IP addresses. We observed that the tracker blacklists the /24 prefix of IP addresses connected to a large number of torrents. In order to prevent this blacklisting, the crawler, once it gets a list of neighbors for a torrent, unsubscribe from this torrent at the tracker, but continues to monitor the peers for 24 hours.

During the 48 days of crawl, we collected 39,298 unique infohashes uploaded by 6,210 users, excluding infohashes of torrents for which we never succeeded to connect to a tracker and torrents that stayed on The Pirate Bay Web site less than 24 hours. In addition, we observed with the list of peers returned by the tracker 9,102,817 unique IP addresses connected to those torrents.

2.3 Large-Scale Crawler Description

The goal of this crawler is to perform a large-scale continuous crawl of a large fraction of the torrents in the Internet. On May 13, 2009, we collected all the .torrent files and meta data available on mininova and The Pirate Bay. The collected meta data is typically the content name, the type of contents, the name of the user who inserted it, comments, etc. We discovered 1,411,940 unique .torrent files on mininova and 974,980 on The Pirate Bay. The Overlap between both sites is only 227,620 .torrents files.

Then, from May 13, 2009 to August 24, 2009, we performed the following three tasks. First, every 24 hours, we collected all the new .torrent files and the associated meta data added during the previous 24 hours period. Both mininova and The Pirate Bay provide a dedicated interface to retrieve the new torrents only. This phase is fast because

there are at most a few thousands of new torrents added per day [11].

Second, we performed *scrape-all* requests on The Pirate Bay tracker every 24 hours. Upon receiving this request, the tracker returns the infohashes of all its torrents. We found that all URLs pointing to The Pirate Bay tracker resolved to 8 IP addresses. Those 8 IP addresses correspond to a cluster of machines that are most of the time synchronized, but not always. In order to do not miss torrents, we performed our scrape-all requests on all 8 IP addresses and then extracted the unique infohashes. Those requests generate 1GB of data and take 15 minutes. In addition to infohashes, scrape-all requests return the number of seeds and leechers in each torrent.

We tried to map the corresponding .torrent files and meta information to all infohashes collected during this phase. When we did not find the corresponding .torrent file from the data retrieved on The Pirate Bay and mininova, we made a request on IsoHunt. We found 365,441 additional .torrent files on IsoHunt.

Third, every two hours, we crawled The Pirate Bay tracker in order to retrieve the list of couples *(IP,port)* for all peers that were in torrents discovered in the scrape-all requests with at least one seed and one leecher. This task works as follows. For each infohash, the crawler computes how many independent requests R must be performed in order to retrieve at least 90% of the peers in the torrent when each request results in 200 peers retrieved at random from the tracker, which is the maximum number of peers returned by The Pirate Bay tracker. Then, the crawler starts a round of R parallel instances of a dummy BitTorrent client, each client started on a different port number, whose only one goal is to get a list of peers from the tracker. Once a round is completed, the task removes all duplicate couples *(IP,port)*, makes sure that indeed 90% of the peers of the torrent were retrieved, and saves the list of couples *(IP,port)*. In the case where fewer than 90% of the peers were discovered during the first round, additional rounds are performed until 90% of the peers are retrieved. The list of *(IP,port)* for the largest torrent takes less than 1 second to retrieve, and this third task is completed on all torrents (between 500k and 800k torrents depending on the crawls) in half an hour. Thus, each tracker crawl can be considered as an instantaneous *snapshot* of the torrents. In the following, we call each of such crawls a snapshot. As for the initial seed crawler, this crawler unsubscribes from each torrent at the tracker in order to prevent being blacklisted.

We ran this crawler from a single computer with a dual core processor, 32 GByte of RAM and 10 TByte NAS. Our crawler is extremely lightweight and scalable for the following reasons. First, we do not use a real BitTorrent client to crawl the tracker, but our own implementation of an optimized lightweight BitTorrent client whose only one goal is to retrieve a list of couples *(IP,port)*. Therefore, we are able to run several thousands of those clients at the same time on a single machine. Second, all crawls to the tracker are made using infohashes retrieved with the scrape-all requests, which is much faster than working on .torrent files directly. Indeed, to crawl the tracker we do not need any disk access to read .torrent files and we do not need to compute the infohash on each .torrent file. Finally, we create 100 processes at the beginning of each crawl and then we give to each process 10% of the infohashes to crawl. Therefore, we do not have any process creation during a crawl.

During the first crawl of our measurement period, we found 1,870,662 unique infohashes returned by the scrape-all request. For all those infohashes, 675,161 corresponds to effective torrents that are torrents with at least one seed and one leecher. We were able to map 501,725 of the effective torrents to .torrent files. For the 675,161 effective torrents, the tracker announced 12,124,600 peers, and we retrieved the couples *(IP,port)* for 11,263,752 (93%) of them. During the last crawl of our measurement, we found 2,048,517 unique infohashes, 715,063 effective torrents, and we were able to map 540,993 of those torrents to .torrent files. For the 715,063 effective torrents, the tracker announced 13,845,696 peers, and we retrieved the couples *(IP,port)* for 12,894,258 (93%) of them.

For all crawls, we retrieved 148 millions unique IP addresses spread among 21,257 ASes, representing 1.98 billions of content downloads and 3.6 exabytes of data downloaded by those peers. We identified 2,524,741 unique infohashes among which 1,196,678 represent torrents with at least one seed and one leecher.

3. PROFILING OF INITIAL SEEDS

In this section, we perform a detailed analysis of initial seeds. We start by defining a methodology to identify initial seeds of torrents and show that we can find the initial seeds corresponding to the most active users. Then, we characterize the initial seeds in terms of type, volume, and number of contents. Finally, we characterize the impact of initial seeds on torrents performance.

In the following, we focus on the infrastructure used by initial seeds that is best identified by IP addresses rather than by the couple (IP,port). Indeed, the stability of the port number varies greatly with BitTorrent clients. For instance, the most popular BitTorrent clients can optionally change the port number each time the client is restarted. However, the IP address is not enough to identify uniquely initial seeds. Indeed, several initial seeds can be behind the same NAT or proxy leading to an erroneous identification of a large initial seed, whereas it consists of several small initial seeds. However, we show at the end of section 3.1 that using IP addresses is reasonable in our context.

3.1 Identification of Initial Seeds

We start to introduce two methods, the active and passive ones, to identify initial seeds. Then, we present three categories of users used to identify initial seeds with the passive method. Finally, we discuss the two methods.

3.1.1 Two Methods to Identify Initial Seeds

Identifying initial seeds is complex. Indeed, most BitTorrent clients implement intelligent initial seeding, usually called super seeding, which makes initial seeds announce themselves as leechers with no or few pieces. The rationale of this strategy is to force leechers to download specific pieces from the initial seed. Then the seed can monitor whether the served pieces are indeed replicated among the leechers. If this is not the case, the initial seed will stop sending pieces to the leechers that do not contribute. Therefore, it is not possible to identify initial seeds by joining a torrent as soon as it appears and looking for the only one seed of the torrent.

In order to identify initial seeds, we developed two methods. We call the first method the *active* one. This method identifies a peer as an initial seed when, joining a new torrent, the crawler finds a single peer in the torrent. This

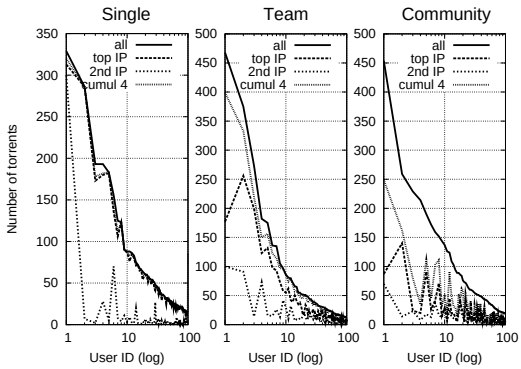

Figure 1: Users classification according to initial seeds strategy. *all* is the number of torrents inserted by each user. *top IP*, resp. *2nd IP*, is the IP address that we find in the largest, resp. second largest, number of torrents for a given user. *cumul 4* shows the number of unique torrents in which the 4 most active IP addresses are found for each user.

peer can announce itself as a seed or a leecher, and we verified that, due to the super seeding mode, very few peers announce themselves as a seed. As long as the peer is the only one in the torrent, this method considers it as the initial seed. Indeed, as our crawler joins the torrent within the first minute of its addition to The Pirate Bay Web site, it is likely that this peer is the initial seed. Using this method, we identified an initial seed for 21,544 torrents. However, this method does not work when there are more than one peer when the crawler joins a torrent.

We introduced a second method, called the *passive* one, to identify initial seeds even for torrents joined too late for the active method to be applicable. Then we use the active method to validate the accuracy of the passive one, because both methods identify initial seeds for torrents in a completely different way. The passive method groups torrents according to the users that inserted them in The Pirate Bay Web site. Then, for each user, we order the IP addresses collected as described in section 2.2 according to the number of torrents they appear in. The hypothesis we make, that we validate in the following, is that, for each user, there is one or a small community of initial seeds that seed all contents for this user. By finding the IP addresses that appear the most frequently per user, we are able to identify the initial seeds for that user.

For the passive method, we only consider IP addresses returned by the tracker during the first connection of the crawler to the torrent. The rationale is that if an IP address is not present during the first connection to the torrent, it is unlikely it is an initial seed. The first IP address, that we call *top IP* in the following, is the IP address that we observe in the largest number of torrents inserted by a user. The second IP address is the second most observed IP address, etc. Using this methodology we identified three main categories of users.

3.1.2 The Three Categories of Users

Fig. 1 shows the three categories of users identified with the passive method. The first category of users is called *single* user and is shown on the left plot. This category is the one for which the top IP address for each user is present

Table 1: Validation of the passive method using the active one. This table shows the success of the active and passive methods to identify the same initial seeds for the same torrents. *Cat* represents one of the three categories of users. *Active*, resp. *Passive*, represents the number of torrents for which the active, resp. passive, method has identified an initial seed. The active method is independent of the category, thus the same number for each category. *Active* ∩ *Passive* is the number of torrents for which both methods identified an initial seed. *Success* is the number of such torrents for which both methods identified the same initial seed.

Cat	|Active|	|Passive|	|Active ∩ Passive|	Success
single	21,544	9,125	5,796	100%
team	21,544	4,334	2,723	99.96%
community	21,544	1,849	724	100%

in more than 80% of its torrents. We only show the first 100 users ordered according to the number of torrents they inserted, but we found 1,712 users in this category excluding all users that inserted strictly fewer than two torrents. The most striking result is that, except for the first user, the second IP address to appear the most frequently is present in just a few torrents for each user. Moreover, looking at the mapping (top IP, user), we found that each user has a different top IP address, which means that the presence of a top IP address is not due to a peer that joined a large number of torrents on The Pirate Bay Web site, but due to a peer that is tightly linked to a user. We validate the accuracy of the passive method for the single user category using Table 1. Indeed, we see that when both the active and passive methods find an initial seed for the same torrent, it is always the same initial seed. In conclusion, for the single user category, each user has a dedicated initial seed that we identified.

The only one exception to the single user category is the first user known as *eztv*. For this user, the first 10 IP addresses are present in more than 150 torrents each. In fact, this user fosters peers that contribute faster than 10 Mbit/s by giving them contents in preview keeping the torrents private to them for a given period of time. Therefore, when a torrent becomes public, there is already several fast peers, always the same fast ones, in the torrent. For the *eztv* user, we consider the top IP address as the initial seed. On the contrary to the other users of the single user category, we cannot be sure it is the initial seed, but it is anyway a peer with a very important role for this user.

The second category of users is called *team* of users and is shown in middle plot in Fig. 1. This category represents the users with a small team of initial seeds. Indeed, for each user, the number of unique torrents to which the 4 most active IP addresses (*cumul 4* curve) are connected to represents at least 80% of all torrents for this user. We only show in Fig. 1 the first 100 users ordered according to the number of torrents they inserted, but we identified 129 users in this category, excluding all users that inserted fewer than 10 torrents. Those users are typical of Web sites maintaining several types of torrents, one initial seed specialized per type. Here we do not claim to find all initial seeds for this category, but that the threshold of 4 IP addresses per user is reasonable because it gives a very low number of false positive as shown in Table 1. Indeed, considering the four top IP addresses in this category as initial seeds, we have a success

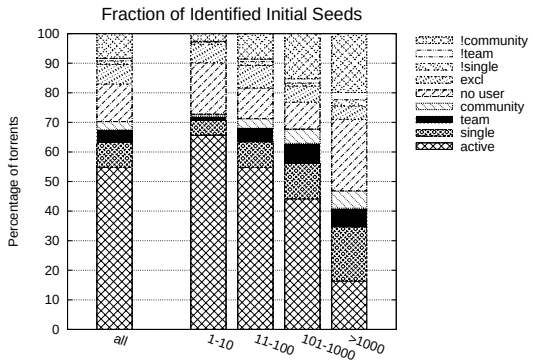

Figure 2: Fraction torrents for which the initial seed is identified by each method. On the x-axis, *all* is for all torrents, *a-b* is for torrents between a and b peers 24 hours after the first connection of the crawler to the torrent, and *>1000* for torrents with more than 1,000 peers after 24 hours.

of 99.96% of initial seeds identified with the passive method also identified with the active one for the same torrents. In the following we consider the four first IP addresses for each team of users to be initial seeds.

The last category of users is called *community* of users and is shown in the right plot in Fig. 1. This category represents users with a large community of initial seeds. Indeed, for each user, the number of unique torrents to which the 4 most active IP addresses (*cumul 4* curve) are connected to represents less than 80% of all torrents for this user. Those users are typical of a large number of initial seeds federated around a community of interest, usually a Web site allowing its members to upload contents. We just represent in Fig. 1 the first 100 users ordered according to the number of torrents they inserted, but we identified 193 users for this category, excluding all users that inserted fewer than 10 torrents. In order to limit the number of false positive, we decided to just consider the top IP address as the initial seed per community of users. Indeed, we observe a 100% success in Table 1.

3.1.3 Discussion of the Methods

Using the passive method, we have shown that for each user some IP addresses appear frequently and that we can consider the most frequent ones to be initial seeds. However, this method can be misled by some peers connecting to all torrents independently of the users who inserted them, those users can be crawlers, highly active peers, etc. Fortunately, such peers are rare and do not impact our results. Indeed, only 77 initial seeds appear for more than one user, and only 8 for more than 3 users. Some initial seeds appear for a few users because they usually participate in parallel to different communities. As we will see in section 3.2, initial seeds are specialized in one or a few types of contents. As the number of suspicious initial seeds is very small, compared to the 2,206 initial seeds identified with the passive method, those initial seeds will not impact our results.

We have seen in Table 1 that with the passive method we can identify the initial seed for a large number of torrents that we do not identify with the active one. Fig. 2 shows the fraction of initial seeds identified and missed by each method for different torrent sizes. The active method (*active*) identifies the initial seed for 55% of all torrents, but this fraction decreases with the torrent size, down to 16% for torrent with more than 1,000 peers. The percentage of torrents detected with the passive method in Fig. 2 excludes all torrents already identified with the active one. We see that the passive method (*single, team, community*) discovers the initial seed for around 15% of all torrents, but the detection ratio improves with the torrent size. In particular, for torrents with more then 1,000 peers, the passive method detects initial seeds for 30% of the torrents, better than the active one that detects initial seeds for 17% of the torrents. On all torrents, the combination of the active and passive methods succeed to detect the initial seed for 70% of all torrents and 47% of large torrents with more than 1,000 peers.

We now focus on which torrents are missed by the passive method. We see in Fig. 2 that there are 13% of torrents that are not associated to any user (*no user*). Thus the passive method does not work for such torrents. There are 7% of torrents that are missed because they are excluded (*excl*) from the classification in the category single user (users that inserted fewer than 2 torrents), and the categories team and community of users (users that inserted fewer than 10 torrents). Finally, the number of torrents in each category, but for which the passive method identified the initial seed is negligible for the categories single user (*!single*) and team of users (*!team*), but important for the category community of users (*!community*), up to 20% for large torrents. In fact, our methodology is conservative for the category community of users as we just consider the top IP address as an initial seed. As our goal is not to identify all initial seeds, but a significant fraction of them well spread over all torrent sizes and all users, we did not try to improve further the methodology for the category community of users. This is an area of improvement for future work.

The correlation we observe in Table 1 is not trivial as the active and passive methods detect seeds in a very different way. In particular, the passive method significantly improves the number of torrents for which an initial seed is identified. However, the passive method detects less than 5% of initial seeds in addition to the active method. Indeed, with the active method we identified 9,184 different initial seeds, and 2,206 with the passive one among which 432 only (less than 5%) were new initial seeds not yet identified with the active method. The reason is that large torrents, the most challenging for the active method, are usually seeded by highly active initial seeds. As those seeds appear in small and large torrents, the active method can identify those initial seeds for at least a few small torrents.

An important limitation of our methodology is that we use IP addresses as identifiers for initial seeds. Indeed, it is possible to aggregate several initial seeds behind the same NAT or proxy and incorrectly identify those seeds as a single one with the IP address of the NAT or proxy. However, we have several evidences that this is not the case. First, initial seeds do not overlap on different users. With initial seeds behind a NAT or proxy we would have expected to see initial seeds for different users behind the same IP address, which is not the case. Second, we will see in section 3.2 that part of the most active initial seeds are on European hosting centers. In that case, all IP addresses are static and public. Finally, for all initial seeds we have checked the BitTorrent client version with time. We found that for each initial seed, the client ID remains the same for all torrents of this seed, or

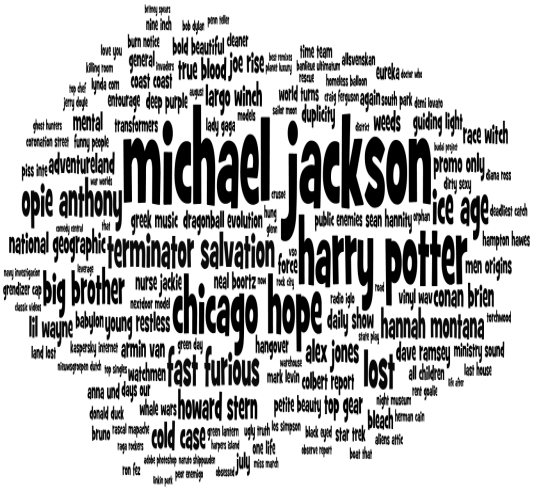

Figure 3: Tags cloud of torrents seeded by initial seeds during the 48 days of crawl. We extract the two most significant keywords from the torrent names and vary their police size to reflect the number of torrents whose name matches those keywords, the largest the keywords, the more frequent those keywords appear in name of torrents.

that the client is updated at one point in time. If the NAT or proxy had impacted our results, we would have observed much more variability in BitTorrent client IDs and we would have observed a much larger number of initial seeds serving different users.

In conclusion, we have developed a methodology to identify a large fraction of initial seeds. Indeed, we have identified the initial seed for 70% of all torrents inserted during our 48 days crawl. A striking outcome of the passive method is that we can identify the initial seed or a small community of initial seeds for a large number of active users. In the following we consider as initial seeds all seeds discovered using the active and passive methods.

3.2 Characterization of Initial Seeds

In this section, we focus on the type of contents served by initial seeds, on the localization of those initial seeds, and on their overall impact in terms of seeded contents.

Fig. 3 shows a tags cloud[1] of significant keywords in torrent names for all torrents seeded by the initial seeds. Here we see that the name of most of the contents inserted by initial seeds contains keywords referring to copyrighted material. However, as we do not analyze the content inserted by initial seeds, we just check that each piece is valid using SHA-1 fingerprint contained in .torrent files, we cannot conclude on the amount of contents with a copyright. It is anyway interesting to see that the contents inserted by initial seeds closely follow events. Indeed, two weeks before we started our crawl Michael Jackson died, and one week after starting our crawl the latest Harry Potter movie was released.

We surprisingly observe in Fig. 4 that a small number of initial seeds, compared to the 9M unique IP addresses

[1]All tags cloud in this paper were generated using http://www.wordle.net with the Creative Commons Attribution 3.0 license.

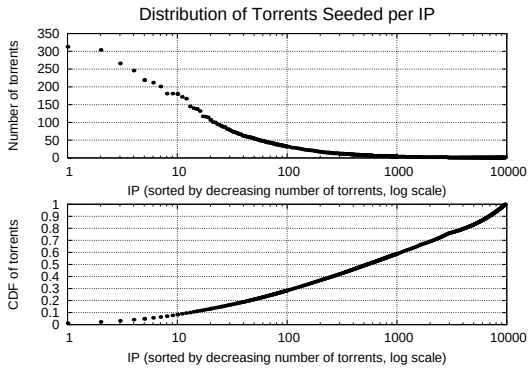

Figure 4: Distribution of the number of torrents seeded by each initial seed. The top plot shows the number of contents per initial seed and the bottom plot shows the cumulative distribution of torrents.

Table 2: Rank, number of contents, volume of contents (GB), country code, and AS name for the top 20 initial seeds.

Rank	# contents	Volume	CC	AS name
1	313	136	NZ	Vodafone
2	304	79	FR	OVH
3	266	152	DE	Keyweb
4	246	34	FR	OVH
5	219	186	FR	OVH
6	212	247	DE	Keyweb
7	201	535	FR	OVH
8	181	73	US	HV
9	181	17	CA	Wightman
10	180	7	SK	Energotel
11	172	161	FR	OVH
12	167	23	RU	Corgina
13	145	197	DE	Keyweb
14	140	11	FR	OVH
15	138	109	US	Aaron
16	132	12	US	Charter
17	117	119	FR	OVH
18	116	109	FR	OVH
19	114	79	NL	Telfort
20	107	225	RU	Matrix

we have seen during our crawl, seed most of the torrents. Indeed, we observe in Fig. 4, top plot, that the most active initial seeds are seeds for a large number of torrents during the 48 days of our crawl. Moreover, according to Fig. 4, bottom plot, those most active initial seeds are initial seeds for a large fraction of the torrents seeded; the top 100 initial seeds represent 30% of all the torrents seeded, and the top 1,000 initial seeds represent 60% of all the torrents seeded.

Focusing on the top 20 initial seeds in Table 2, we observe that half of those initial seeds are located in France and Germany. In fact, OVH that is located in France and Keyweb that is located in Germany are very large hosting centers offering, for a cheap price, high profile servers with a high speed, typically 100Mbit/s, network connectivity. Those servers are used to run what is called a *seedbox* that is a dedicated server specialized in file sharing. Those seedboxes come with a dedicated support, installed BitTorrent clients, and can be remotely administered using, for instance, a Web interface like the one offered by TorrentFlux [3]. Therefore, those seedboxes attract initial seeds that require a high speed network connectivity.

However, concluding that the human beings behind those initial seeds live in the countries of the hosting centers would be a mistake. Indeed, those servers are rented by people abroad from those countries. For the 1,515 torrents seeded

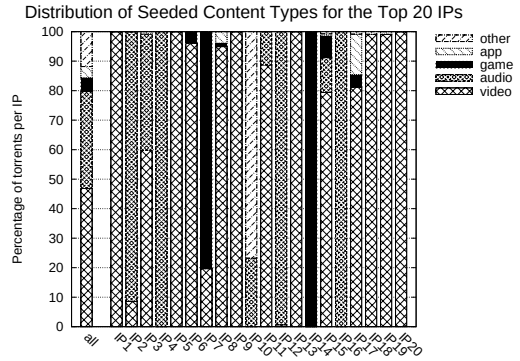

Figure 5: Repartition in types of torrents seeded by the top 20 initial seeds. *all* represents the repartition for all initial seeds found in the crawl.

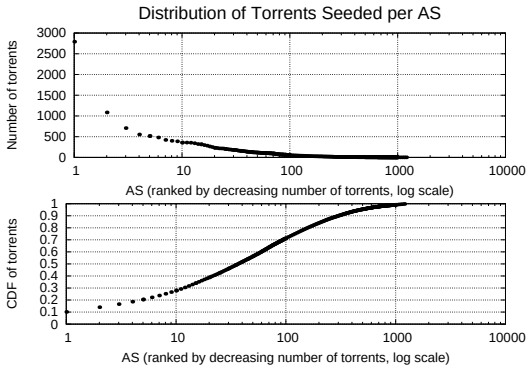

Figure 6: Distribution of the number of torrents seeded per AS. The top plot shows the number of contents per AS and the bottom plot shows the cumulative distribution.

Table 3: Rank, number of contents, volume of contents (GB), country code, and AS name for the top 20 ASes.

Rank	# contents	Volume	CC	AS name
1	2,791	2,984	FR	OVH
2	1,084	1,343	EU	Telenor
3	707	756	DE	Keyweb
4	503	403	US	AT&T
5	516	385	US	Verizon
6	480	811	NL	Zigoo
7	422	317	US	Charter
8	400	393	EU	Telia
9	388	373	AT	UPC
10	359	316	EU	NTL
11	358	380	NL	Telfort
12	354	161	US	High Velocity
13	342	300	GB	BE
14	323	197	GB	iNet
15	316	136	NZ	Vodafone
16	298	221	IT	Telecom Italia
17	284	44	AU	MPX
18	264	230	SE	ComHem
19	250	118	US	Qwest
20	232	130	ES	Telefonica

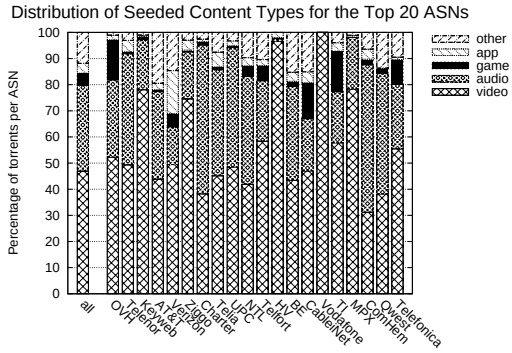

Figure 7: Repartition in types of torrents seeded by the top 20 ASes. *all* represents the repartition for all ASes found in the crawl.

by the initial seeds in OVH among the top 20 initial seeds, we parsed the content names to find some specific strings, all matches are case insensitive. We found 552 times the word *spanish*, and after a manual inspection found that it always refers to a content translated in Spanish, or to a content with Spanish subtitles. Looking for the string *fr*, which is typical for contents in French, and *ge* or *de*, for contents in German, we only found after manual inspection 13 contents in French (all related to learning material for children), and 5 in German. Concerning Keyweb, for the 623 torrents seeded by initial seeds in Keyweb among the top 20, we found 228 content names matching the string *spanish* and confirmed that all those names refer to contents in Spanish. However, we did not find, after manual inspection, any content matching the strings *fr*, *ge*, or *de* referring to contents in French or German.

Looking at the repartition per content type of contents seeded by the first 20 initial seeds in Fig. 5, we observe that those seeds are highly specialized in one or two types of contents. This specialization in types of contents explains why we do not observe any correlation in Table 2 between the number of contents and the volume of contents. Indeed, some types of contents, like app or audio, are much smaller than other types, like video.

We now focus on the distribution of the initial seeds per AS. We observe in Fig. 6 that the top AS represents 10% of all the contents seeded during our crawl, and the top 10 ASes represent 30% of all the contents seeded.

Indeed, we see in Table 3 that OVH is, by far, the most popular AS for initial seeds. Among the other top 20 ASes there is only one other hosting center, Keyweb, that is ranked third. The other ASes are classical Internet (ADSL, or optical) providers. Therefore, top ranked initial seeds can be hosted using a broadband home connection, even if the dedicated solution offered by hosting centers attracts a large fraction of highly active initial seeds.

Unlike for initial seeds, Fig. 7 shows that all types of contents are seeded from most of the ASes. Therefore, there is no correlation between the type of content seeded and the AS of the initial seed. This is due to the large number of initial seeds per AS. The main exception is Vodafone (ranked 15) for which most of the content (313 over 316) are seeded by a single initial seed that we identified as an initial seed of the user *eztv* that is specialized in TV show contents.

In conclusion, we have seen that whereas contents are downloaded by millions of peers, most contents are seeded by a small community of highly active initial seeds. However, deriving the country of the human being operating an initial seed based on the IP address of this initial seed would be misleading. Therefore, carefully building maps of BitTorrent usage has to be done. We provided here new results and

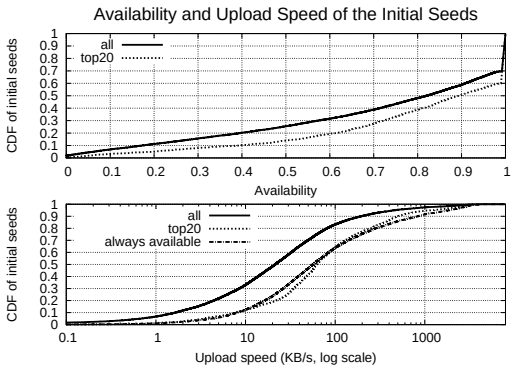

Figure 8: Availability (top) and upload speed (bottom) of initial seeds. *all* and *top20* represent the availability of for all initial seeds and of the top 20 initial seeds for each torrent they seed. *always available* is for all initial seeds that are always available.

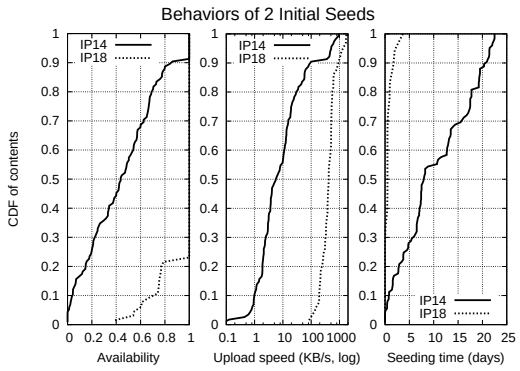

Figure 9: Availability, upload speed, and seeding time for two initial seeds running μTorrent. The two initial seeds are seed 14 and 18 as shown in Table 2, both are hosted by OVH.

a new methodology in that direction.

3.3 Performance Issues with Initial Seeds

In this section, we exhibit a fundamental issue for initial seeds in the way torrents are queued in BitTorrent clients, then we explore the impact of initial seeds upload speed on torrents performance.

In the following, we only consider the initial seeds discovered with the active method. The reason is that finding the initial seed with the active method guarantees that the crawler joined the torrent when it was created. Therefore, all performance measures can be conducted during the entire period of initial seeding, i.e., up to the moment the initial seed has uploaded an entire copy of the content.

For a given torrent, let T be a period of time ranging from the time the crawler joined this torrent to the time the initial seed of the torrent has uploaded a full copy of the content. A new piece is a piece that was never observed by the crawler in its neighborhood, except on the initial seed. As the initial seed is the only one to have new pieces, when the crawler receives at least one new piece or one HAVE message for a new piece during a 10 minutes period, we say that the initial seed has been active during this period. By definition, the presence of the initial seed during a period T is the sum of the duration of all 10 minutes periods included in T during which the initial seed has been active. We have also tested with periods of 20 and 30 minutes without any significant differences. The availability of an initial seed for each torrent is its time of presence divided by T.

To monitor the upload speed of an initial seed, we count the number of unique HAVE messages for new pieces received during the period T, multiply it by the size of a piece, and divide it by T. Therefore, we compute the average upload speed of initial seeds during a period T.

To monitor the download speed of a neighbor, we count the number of HAVE messages it sent to the crawler up to the moment the initial seed has uploaded a full copy of the content or to the moment the neighbor has left the peer set of the crawler. The calculation is similar as for the upload speed of the initial seed.

3.3.1 Queuing Issues of Initial Seeds

Surprisingly, we see in Fig. 8 top plot, that only 30% of the initial seeds are always connected to the torrent they seed (*all* curve) from the instant the crawler joined the torrent up to the instant the initial seed has uploaded a full copy of the content. We also see that during the period of time required to upload a content, 50% of the initial seeds were disconnected from the torrent 20% of the time. Moreover, we see that the availability of the top 20 initial seeds (*top20* curve) is not dramatically better, whereas we expected to observe highly available initial seeds. To compute the availability, we have also tested periods of 20 and 30 minutes, instead of 10 minutes, without any significant difference.

We observe that the average upload speed on all initial seeds, *all* in Fig. 8 bottom plot, is highly variable depending on the torrents, 35% of the torrents are uploaded at less than 10 kByte/s by the initial seed, and 15% of the torrents uploaded at more than 100 kByte/s. We observed that the average upload speed for the top 20 initial seeds (*top20* curve) and the seeds always available (*always available* curve) is very similar, but the average upload speed is not as high as we expected. Indeed, we see for the top 20 initial seeds that 65% of the torrents are uploaded at less than 100 kByte/s, whereas half of them are hosted by dedicated hosting centers with high speed network connectivity.

In order to understand the large difference we observe between the actual availability and upload speed per torrent, and what we expected, we focus on the top 20 initial seeds. Fig. 9 shows the availability, upload speed, and seeding time of two initial seeds hosted by OVH. As those seeds are hosted in the same location, we expected to observe a similar performance, which is not the case. Indeed, we observe that seed 18 is typical of a high performing initial seed. It has a high availability, always available for 77% of the torrents, and offer an excellent average upload speed, larger than 100 kByte/s for all torrents and larger than 1,000 kByte/s for 10% of the torrents. However, seed 14 is performing poorly. It is always available for less than 10% of its torrents, and the average upload speed is lower than 10 kByte/s for 55% of the torrents.

The reason of this behavior is explained by the seeding time of those initial seeds and reveals a fundamental issue with the queuing algorithm used in BitTorrent clients. Fig. 9, right plot, shows the seeding time for both initial seeds. We define the seeding time of a torrent as the period of time starting from the first time we saw the initial seed

to the last time the initial seed was in that torrent[2]. We observe that seed 18, the one performing best, seeds torrent just a few hours and at most 4 days in the worst case. In the contrary, seed 14 seeds 28% of its torrents more than 5 days, up to 23 days.

In order to understand the impact of the seeding time on availability and upload speed, we need to introduce the concept of queuing used by BitTorrent clients. The two most popular BitTorrent clients, μTorrent and Vuze [11], use a similar notion of queuing, but here we focus on μTorrent. Although it is possible to add to a BitTorrent client a large number of .torrent files, the client will only connect to a few torrents that we call the active torrents. The rationale is to guarantee a good upload speed per torrent. Therefore, the number of active torrents in parallel is a function of the client upload capacity. With μTorrent, an upload speed of 1 Mbit/s corresponds to 6 active torrents, and 10 Mbit/s to 15. Which torrent will be active is determined by the queuing algorithm of the client. This algorithm is a collection of many heuristics. In particular, a newly started torrent will have the highest priority for 60 minutes, and a torrent with no seed will have a high priority in the queue. If the first copy of the content takes more than 60 minutes to complete, which is likely, and that the μTorrent client has many other unpopular torrents with no seed in the queue, then it is possible that the newly added torrent will be stopped and that an unpopular torrent will become active instead. In fact, as there is no notion of initial seeds in μTorrent, there is no way for the client to discover that the content has not yet been uploaded at least once. This pathological behavior is likely to appear with initial seeds with several unpopular contents queued.

Whereas the initial seed 18 leaves fast the torrents it seeds, thus a small number of torrents queued in the BitTorrent client, the initial seed 14 stays in the torrents it seeds for a long time, thus a large list of torrents queued. As a consequence, torrents for initial seed 14 are likely to become inactive, due to the queuing heuristics we described, before a first copy of the content is uploaded. This period of inactivity will have a dramatic impact on the availability and average upload speed of the initial seeds for those torrents.

Our conclusion is confirmed on all torrents seeded by the top 20 initial seeds in Fig. 10. Indeed, we observe that the torrent with the highest average upload speed are torrents always available (top plot) and torrents with a small seeding time (bottom plot).

In summary, whereas the queuing heuristics in BitTorrent clients seeks to maximize the upload utilization of a peer, it does not appear to be adapted to the case of initial seeds. Indeed, for initial seeds, it is fundamental to upload a full copy of a content as soon as possible even if it is at the expense of a lower overall utilization of the upload capacity on the client.

One way to solve this issue would be to create an initial seed property per torrent. Then the BitTorrent client should give the highest priority to torrents for which it is the initial seed when one copy of the content has not been fully uploaded. It is beyond the scope of this study to perform a detailed analysis of the queuing algorithms used in BitTorrent clients, but it is undoubtedly an interesting area of future investigation. In particular, there is a significant area of improvement for initial seeds queuing strategy.

[2]This information is not obtained by the initial seeds crawl that just collects information for the first 24 hours of the torrent's life, but by the large-scale crawl that we describe in section 2.3 restricted to the same 48 days as the initial seeds crawl.

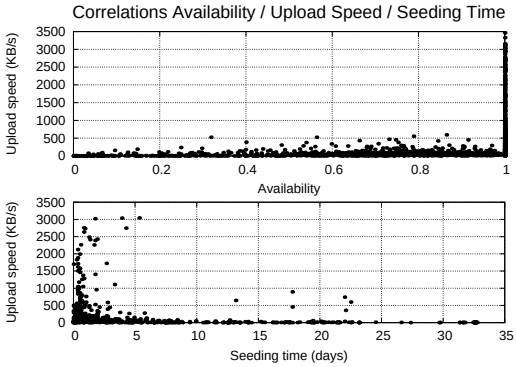

Figure 10: Correlation Availability, upload speed, and seeding time for all torrents seeded by the top 20 initial seeds. Each dot represent the correlation of a single torrent seeded by one of the top 20 initial seeds.

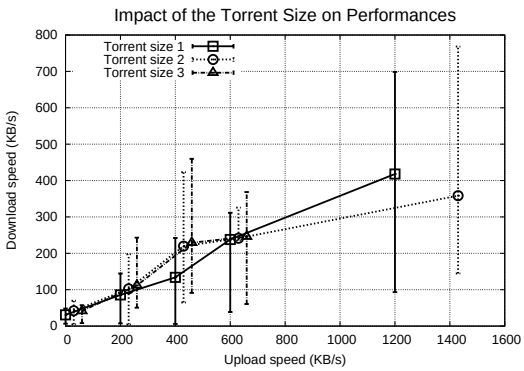

Figure 11: Impact of initial seeds upload speed on torrent performance with torrent size. The lines represent the correlation between the initial seeds' upload speed and average peers' download speed for torrents with $1-10$ peers (torrent size 1), $11-100$ peers (torrent size 2), and more than 100 peers (torrent size 3). The error bars represent the 30th and 80th percentiles computed on all peers.

3.3.2 *Impact of the Upload Speed of Initial Seeds on Torrents Performance*

In this section, we only consider initial seeds that are always available. Interestingly, the upload speed of initial seeds is correlated to torrents performance independently of the torrent size. Indeed, we see in Fig. 11 that the larger the upload speed of the initial seeds, the larger the average download speed on peers. We also observe that the gap between the 30th and the 80th percentiles increases. This is due to the heterogeneity of the download speed of peers. Some peers have low download speed, therefore, increasing the insertion of new pieces in the torrent will not bring much benefits. However, other peers are very fast and they will benefit from an larger upload of the initial seed.

This is confirmed by Fig. 12. We see that the fast peers benefit more from a high upload speed of the initial seed than slow ones.

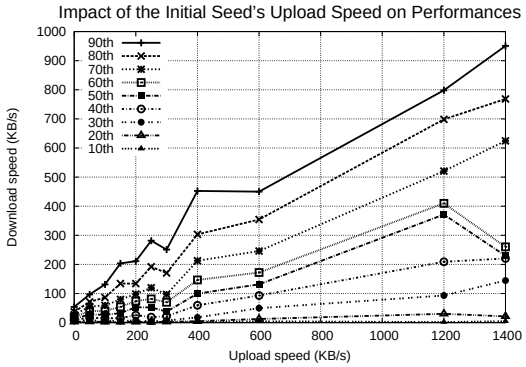

Figure 12: Impact of initial seeds upload speed on peers download speed. We give the peers download speed statistics for all torrent sizes. The lines represent the correlation between the initial seeds' upload speed and a given percentile of peers' download speed.

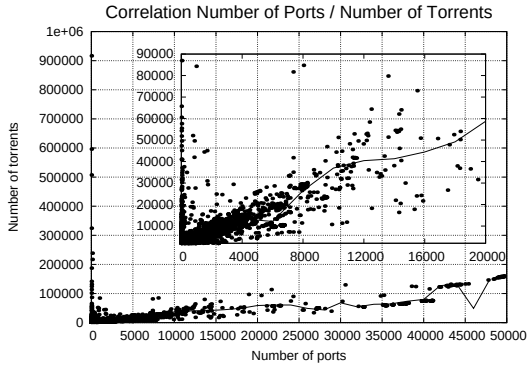

Figure 13: Correlation of the number of ports per IP address and of the number of torrents for the top $10,000$ IP addresses. Each dot represents an IP address. The solid line is the average number of torrents on the 148M IP addresses computed per interval of $2,000$ ports.

In conclusion, we have shown that on 24,544 torrent inserted during a 48 days period, the upload speed of the initial seeds is critical to the download speed of peers. Whereas it is known that the upload speed of initial seeds is important for the performance of torrents [6] and that models [10] predict that the torrent size does not significantly affect the peers download completion time, it is fundamental to validate that on a large number of real torrents and with real peers those findings are true.

4. PROFILING OF HIGHLY ACTIVE PEERS

In this section, we focus on the characterization of highly active peers, i.e., peers that are seen in the largest number of torrents. However, this characterization is challenging because of the huge size of the trace generated by the crawler described in section 2.3. In particular, we found 148M IP addresses and more than 510M couples *(IP,port)* collected on a period of 103 days.

Ordering the IP addresses according to the total number of unique torrents in which we see them on the 103 days of the large-scale crawl, we observe a power law relation. In particular, the top 10,000 IP addresses that we observed in the largest number of torrents were present in at least 1,636 torrents each during the 103 days, the top 100,000 IP addresses were present in at least 309 torrents. In the remaining part of this section, we specifically focus on the top 10,000 IP addresses, as we want to understand the behavior of highly active peers. Therefore, those top 10,000 IP addresses are the ones that are seen in the largest number of torrents on the 103 days period.

However, characterizing highly active peers with a large-scale continuous crawl is challenging. Indeed, it is hard to identify a peer using its IP address or the couple *(IP,port)*, because the meaning of this information is different depending on the way the peer connects to torrents. A peer can connect through a NAT operated by its ISP due to scarcity of IP addresses allocated to its ISP. It may want to hide its identity from anti-piracy groups using HTTP or SOCKS proxies, using paid-for BitTorrent proxies like iPREDator operated by The Pirate Bay that are usually SOCKS proxies offering a VPN service, or using anonymous networks like Tor [2]. In all those cases, a large number of peers can be seen using the same IP address. Furthermore, the couple *(IP,port)* cannot be used to uniquely identify a peer, as there is no guarantee that a port or IP associated to a peer will remain the same over time. Therefore, in many cases, just using the IP address or the couple *(IP,port)* can erroneously make a peer be identified as a highly active peer. We focus on those false positives in the following. On the contrary, a highly active peer using a dynamic IP address can be identified as a regular peer. We do not consider this case of false negatives in the following, as we believe that false positives are more frequent than false negatives. However, this is a possible area of improvement in the future.

We confirm this complexity in Fig. 13 where we can see that neither an IP address nor the couple *(IP,port)* can be used to uniquely identify a peer in all cases. Indeed, we see in Fig. 13 that for most of the IP addresses the number of torrents increases linearly with the number of ports. Moreover, the slope of this increase corresponds to the slope of the average number of torrents per IP over all 148M IP addresses (solid line). Each new port corresponds to between 2 and 3 additional contents per IP address. Therefore, it is likely that those IP addresses correspond to NATs with a large number of users behind them. There are also many IP addresses for which the number of torrents is much larger. Thus those IP addresses do not correspond to the typical behavior of a NAT.

In order to understand what are the IP addresses that significantly differ from NATs, we identify 6 categories of highly active peers. The two first categories are HTTP and SOCKS public proxies for which we retrieved from the sites hidemyass.com and proxy.org a list of IP addresses. We found in those lists 81 HTTP proxies and 62 SOCKS proxies within the top 10,000 IP addresses.

The third category is composed of Tor nodes for which we performed a reverse DNS lookup for the top 10,000 IP addresses and extracted all names containing the *tor* keyword and manually filtered the results to make sure they are indeed Tor nodes. We also retrieved a list of nodes on the Web site proxy.org. We found a total of 174 Tor nodes within the top 10,000 IP addresses.

The fourth category is composed of monitors. A monitor corresponds to a peer that spies a huge number of torrents

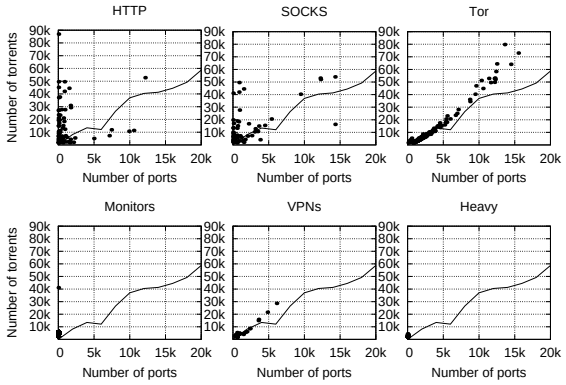

Figure 14: Correlation of the number of ports per IP address and of the number of torrents among the top $10,000$ **IP addresses for six categories of highly active peers.** Each dot represents an IP address. The solid line is the average number of torrents on the 148M IP addresses computed per interval of $2,000$ ports.

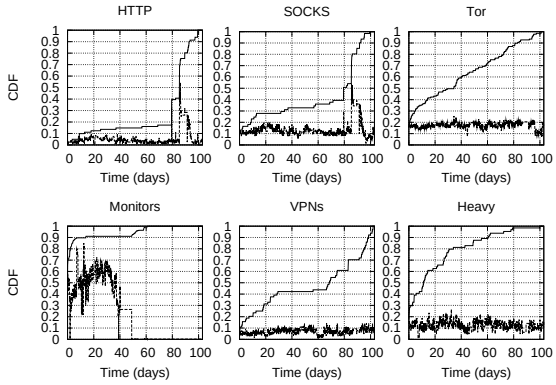

Figure 15: Benefit of the continuous crawl compared to a single snapshot to identify highly active peers of 6 categories. For each snapshot we compute the top $10,000$ IP addresses that we observed in the largest number of torrents. The dashed line represents the number of IP addresses of a given category discovered in the top $10,000$ IP addresses of each snapshot that are also among the top $10,000$ IP addresses that downloaded the largest number of torrents on all snapshots. The solid line represents the cumulative number of those discovered IP addresses.

without downloading anything. We identified two ASes, corresponding to hosting centers located in the USA and in UK, containing a large number of IP addresses within the top 10,000 with the same behavior. Indeed, we were never able to download any piece from those IP addresses, and all those IP addresses always use a single port in the 103 days. Therefore, those IP addresses do not look like regular peers, but like a permanent infrastructure. We found 1052 such IP addresses within only two ASes.

The fifth category is composed of paid-for BitTorrent proxies that we call VPN in the following to make a clear distinction with the generic purpose HTTP and SOCKS proxies. To find VPNs, we performed a reverse DNS lookup for the top 10,000 IP addresses and extracted all names containing the *itshidden*, *cyberghostvpn*, *peer2me*, *ipredate*, *mullvad*, and *perfect-privacy* keywords and manually filtered the results to make sure they are indeed the corresponding VPNs. Those keywords correspond to well know paid-for VPN services. We found 30 VPNs within the top 10,000 IP addresses.

The last category is composed of heavy peers that corresponds to IP addresses that used fewer than 10 different ports during the 103 days of the trace, and from which the crawler downloaded pieces. Therefore, those highly active peers cannot be a large NAT due to the small number of ports, or a monitor as data is downloaded from them. They correspond to real peers that we saw in a large number of torrents. We found 77 such heavy peers.

We do not claim that we have found all peers in each category. Instead, we have identified a few peers in each category within the top 10,000 peers that we use to characterize the behavior of the highly active peers.

We see in Fig. 14 that for HTTP and SOCKS proxies the number of torrents per IP address is much larger than for NATed IP addresses (solid line). Considering the huge number of torrents in which we see those IP addresses, compared to the case of heavy peers, it is likely that the proxies are used by anti-piracy groups. Indeed, we see in Fig. 15 that the crawler suddenly stops discovering new IP addresses for the monitors category after day 50. In fact, by that date, The Pirate Bay tracker changed its blacklisting strategy to reject IP addresses that are present in a large number of torrents with a much lower threshold (which was not an issue for our crawler as we unsubscribe regularly from the tracker, see section 2.3). The result is that monitors that we identified were unable to continue to spy torrents using this strategy. However, we observe at day 80 that the number of HTTP and SOCKS proxies suddenly increased, probably corresponding to anti-piracy groups migrating their monitoring infrastructure from dedicated hosting centers to proxies. Considering, the coordination we observe in Fig. 15 in the increase of the HTTP and SOCKS proxies, it is likely that those proxies were used in a coordinated effort.

The correlation for monitors and heavy users in Fig. 14 does not show any striking result, therefore we do not discuss it further. However, we observe in Fig. 14 that for Tor nodes and VPNs the number of torrents per IP address is close to the NATed IP addresses (solid line). For large number of ports, Tor nodes deviate from the standard behavior of NATed IP addresses. In fact, we found that just a few IP addresses are responsible of this deviation, all other Tor nodes following the trend of the solid line. We guess that those few IP addresses responsible for the deviation are also used to spy torrents. Indeed, it is unlikely that a highly active peer will use Tor nodes due to their poor performance. Indeed, using the initial seeds crawl we performed measures of download speed of peers for 3,191,145 different peers. Those measures are performed by counting HAVE messages received by the crawler. We see in Fig. 16 that the performance of Tor nodes is really poor. However, the performance of the heavy peers is better, but still lower than the average. In fact, we see that the fastest heavy peers are very fast and that some peers are very slow. This behavior is typical of the queuing issues discussed in section 3.3.1 when there are a lot of torrents queued in a BitTorrent client. But, this issue is not as critical as it is for initial seeds, because even if the performance per torrent is poor, the overall performance of the BitTorrent client might be good. Also, we observe that the performance is the best for VPNs. In fact, that means that peers using VPNs are usually very

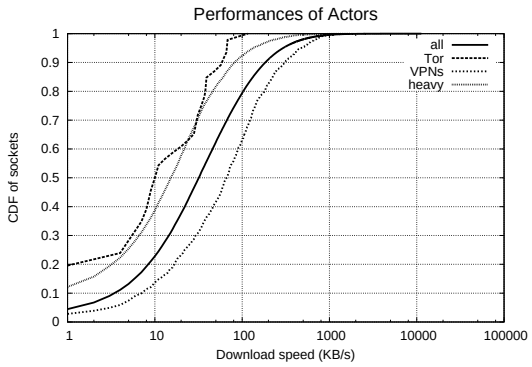

Figure 16: Performance of peers. *all* is for the 3M of peers for which we performed a download measure.

fast peers, and that VPNs do not dramatically decrease the performance of those peers.

Finally, we see in Fig. 15 that the information obtained with a continuous crawl is much larger that with a single snapshot of the peers. Indeed, for instance, whereas we take 12 snapshots per day (one every two hours), we need more than 20 days of continuous crawl to discover 60% of the heavy peers. The sawtooth behavior is due to the diurnal pattern, and the lack of sawtooth behavior is due to some snapshots that we failed to capture after a crash of the machine performing our crawls.

In summary, we identified 6 different categories of highly active peers using our large-scale continuous crawl. In particular, we have been able to identify anti-piracy groups spying torrents, and we even identified a change in their monitoring infrastructure, which make us confident that our crawl covers most of the highly active peers. We confirmed the belief that Tor is inefficient for BitTorrent distribution and found that VPN solutions are usually used by well-provisioned peers. Last, we have shown that capturing a single snapshot, while much easier, will bring significantly less information than a continuous crawl.

5. RELATED WORK

Our work differs from the related work in scale, time span, and focus. As for the scale, Siganos et al. crawled the top 600 torrents from The Pirate Bay [9] Web site during 45 days collecting 37M IP addresses. They do not give the number of simultaneously observed IP addresses, but using a small number of parallel torrents prevents from identifying top initial seeds and highly active peers. Choffnes et al. [4] monitored 3,029 simultaneous peers on average, and Piatek et al. *partially* crawled 55,523 torrents [7]. It is unclear how many simultaneous peers Piatek and al. have monitored as they reported being detected and blocked when being too aggressive [8]. Dan et al., who reported to have crawled 2.4M torrents with 37M peers, used a different terminology for crawling [5]. Indeed, they performed scrape requests to tracker, which only gives the number of peers per torrent, but not the IP addresses of those peers. The data they collected is much easier to get and completely different in the focus. Zhang et al. [11] collected .torrent files for 4.6M torrents during a nine month period. However, they only present a single crawl of trackers, using an infrastructure of 35 machines, that consists in 5M IP addresses collected on a 12 hours period.

None of those studies addresses the identification of initial seeds and highly active peers, nor they present the traces to perform such a work.

6. CONCLUSION

In this paper, using two crawls of BitTorrent, we perform the first in-depth analysis of initial seeds and highly active peers. We show that it is possible to identify the IP address of initial seeds and to map those seeds to users that inserted torrents in torrent indexing sites for 70% of the torrents. We show that the initial seeds form a small community and that some of the most active initial seeds use hosting centers, which make the identification of the location of the human being running those initial seeds complex. Finally, we show that there is a fundamental issue in the way torrents are queued in BitTorrent clients for initial seeds.

We also show that highly active peers represent several very different categories of peers including monitors, NATs, VPNs, proxies, and heavy peers. Using the large-scale continuous crawl we were able to track anti-piracy groups, to confirm that Tor is inefficient for BitTorrent distribution and that VPN solutions like iPREDator are used by well-provisioned peers.

We have shown that it is possible to continuously monitor a significant fraction of all torrents of the Internet from a single machine. This undoubtedly raises many concerns on the privacy of users running BitTorrent. However, fixing this privacy issue is not trivial as it partly comes from the very open design of the trackers. Exploring how to improve the privacy of users running BitTorrent is an interesting area for future work.

7. REFERENCES

[1] Three-strikes law. http://en.wikipedia.org/wiki/Three_strikes_law.
[2] Tor. https://www.torproject.org/.
[3] Torrentflux. http://www.torrentflux.com/.
[4] D. Choffnes, J. Duch, D. Malmgren, R. Guiermá, F. E. Bustamante, and L. Amaral. Swarmscreen: Privacy through plausible deniability in p2p systems. Technical report, Northwestern University, March 2009.
[5] G. Dán and G. Carlsson. Dynamic swarm management for improved bittorrent performance. In *IPTPS'09*, Boston, MA, USA, 2009.
[6] A. Legout, N. Liogkas, E. Kohler, and L. Zhang. Clustering and Sharing Incentives in BitTorrent Systems. In *SIGMETRICS'07*, San Diego, CA, USA, June 2007.
[7] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson. One hop reputations for peer to peer file sharing workloads. In *NSDI'08*, San Franciso, CA, USA, 2008.
[8] 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 *HotSec'08*, San Jose, CA, USA, July 2008.
[9] G. Siganos, J. Pujol, and P. Rodriguez. Monitoring the bittorrent monitors: A bird's eye view. In *Proc. of PAM'09*, Seoul, South Korea, April 2009.
[10] X. Yang and G. de Veciana. Service capacity of peer-to-peer networks. In *Proc. of INFOCOM*, Hong-Kong, China, March 2004.
[11] C. Zhang, P. Dunghel, D. Wu, and K. Ross. Unraveling the bittorrent ecosystem. Technical report, Polytechnic Institute of NYU, 2009.