Peeking through the BitTorrent Seedbox Hosting Ecosystem

Dario Rossi, Guilhem Pujol, Xiao Wang, Fabien Mathieu

Peeking Through the BitTorrent Seedbox Hosting Ecosystem

Dario Rossi[1,2], Guilhem Pujol[2], Xiao Wang[2], Fabien Mathieu[3]

[1]Telecom ParisTech, dario.rossi@enst.fr
[2]Ecole Polytecnique, first.last@polytechnique.edu
[3]Alcatel Lucent Bell Labs, fabien.mathieu@alcatel-lucent.com

Abstract. In this paper, we propose a lightweight method for detecting and classifying BitTorrent content providers with a minimal amount of resources. While heavy methodologies are typically used (which require long term observation and data exchange with peers of the swarm and/or a semantic analysis of torrent websites), we instead argue that such complexity can be avoided by analyzing the correlations between peers and torrents. We apply our methodology to study over 50K torrents injected in ThePirateBay during one month, collecting more than 400K IPs addresses. Shortly, we find that exploiting the correlations not only enhances the classification accuracy keeping the technique lightweight (our methodology reliably identifies about 150 seedboxes), but also uncovers seeding behaviors that were not previously noticed (e.g., as multi-port and multi-host seeding). Finally, we correlate the popularity of seedbox hosting in our dataset to criteria (e.g., cost, storage space, Web popularity) that can bias the selection process of BitTorrent content providers.

1 Introduction

Being one of the most successful P2P applications, BitTorrent has been dissected under many angles, from focused performance analysis to broad studies of the whole ecosystem. The present work focuses on the identification and characterization of the peers that inject content in P2P systems. A simple method consists in joining a swarm just after it has been advertised on a torrent website, in the hope that the monitoring peer finds the swarm populated with one unique seeder. However, this method fails when multiple peers are found, e.g., due to the injection of fake IPs or the use of multiple initial seeders. In such cases, heavier methodologies are usually used, which require long term observation and data exchange with peers of the swarm and/or a semantic analysis of torrent websites. These techniques require the exchange of significant amounts of data (see Sec. 2 for details). We instead advocate a lightweight technique that exploits multiple sources of correlations between peers and swarms (see Sec. 3).

Obvious tensions exist between BitTorrent users, content copyright holders and government agencies – recent studies indicate an increased uptake in the use of foreign seedboxes to bypass local jurisdictions [1]. Following the authors of [2, 3], we do not take sides in this struggle. Detecting the content provider can be used for multiple purposes: torrent Websites could automatically detect producers of sensitive, unsolicited or non-compliant content, making it more efficient to remove content and accounts

(which indeed happens but at a relatively long timescale [4], suggesting humans are in the loop); governmental agencies could use similar techniques to narrow down the list of potential suspects, thereby reducing the risk of generating false alarms (i.e., targeting users printer and WiFi access point as in [2]).

This paper makes a number of contributions. We develop a lightweight methodology exploiting multiple sources of correlation (Sec. 3), pinpointing a small number of peers (4K) responsible for a large fraction of content (60%). We confirm that our methodology correctly identifies seedboxes by performing reverse DNS lookups and extensive manual verification: we can associate about 150 peers, responsible for about 40% of all torrents, to known seedbox services (Sec. 4). Our methodology also exposes two interesting, yet previously unnoticed, seeding behaviors:

- seedboxes using multiple ports for the same IP address (that others have generally considered to be multiple peers behind a NAT box [4, 5]);
- groups of heterogeneous seedboxes (e.g., using different IPs, hosting providers, ASs), that consistently seed the same set of torrents, and that are thus managed by a single BitTorrent content provider.

We then estimate the seeding cost incurred by BitTorrent content providers, and correlate the popularity of seedbox hosting in our dataset to criteria (e.g., cost, storage space, Web popularity) that can bias their selection (Sec. 5). Finally, we conclude the paper outlining also future directions (Sec. 6).

2 Related work

The study of alleged content providers in BitTorrent started with the seminal work of Piatek et al. [2], in turn a byproduct of another work from the same authors [3]: while studying BitTorrent performance in the wild, they managed to attract a number of (false positive) Digital Millennium Copyright Acts (DMCA) takedown notices. Authors showed that simple techniques may implicate arbitrary network endpoints in illegal content sharing (e.g., as a tracker may let peers specify an arbitrary IP in their announce), effectively managing to frame printers and wireless APs into getting DMCA notices. Since the study of BitTorrent requires some active crawling, authors seldom provide a broad view of the ecosystem [6], usually preferring to focus on some specific aspect [2–5, 7–11].

We report a brief summary of the closest work to ours in Tab. 1. As just said, [2, 3] focus on DMCA notices. Authors in [8] are interested in application- and network-layer heuristics to find clients with deviant behaviors (e.g., monitoring peers) to possibly construct blacklists on-the-fly to enforce user privacy. Authors in [4, 11], classify content providers in fake, profit-driven and altruistic categories. Both work agree that roughly half of the top-100 producers are profit-driven [4, 11], and [11] further points out that fake publishers are dominant among the top-861 producers, which are responsible for an estimated 60% of the total downloaded BitTorrent content. Hence, effective filtering of fake publishers could reduce network resource waste. Finally, [5] provides heuristics to classify the user type (e.g., proxy, Tor, monitors) while [10] does so by using PageRank-like algorithms on the user-content graph.

Table 1. Comparison of Related Work

Ref	Year	Duration	Torrents	Peers	Focus
[3]	2007	30 days	55K	-	DMCA notice
[2]	2008	30 days	27K	-	DMCA notice
[8]	2009	45 days	top 600	37M	240 deviant clients
[5]	2009	48 days	39K	148M	top 10,000 users
[4]	2010	80 days	55K	35M	top 100 publisher (37% of content)
[11]	2011	38 days	52K	16M	top 861 publisher (67% content)
this work	*2013*	*37 days*	*57K*	*443K*	*150 seedboxes*

The above work generally relies on direct data exchanges with discovered peers, to verify that they actually own copies of the content (as otherwise the problems noticed in [2] may appear). Such resource-consuming approaches are hard to avoid if one aims at studying with precision and certainty one given peer or one given swarm, but we advocate that a preliminary filtering may significantly reduce the amount of work needed. For example, by joining a swarm immediately after its torrent has been published, we significantly reduce the number of IPs collected to some 443K (as opposite to 10M-100M in other works). Also, we employ an aggressive filtering phase that reduces the false alarm rate to a minimum, thus pinpointing 150 seedboxes responsible for about 40% of the content. Direct techniques such as those proposed in [2, 4, 5, 11] could then be used on this more reduced, and more interesting, producer subset.

The present work also differs from [2, 4, 5, 11] by not relying on cross-checking with external sources (e.g., user ID in the PirateBay portal) as they can easily be gamed (e.g., a sybil attack creating multiple user IDs). Conversely, we argue that network level data is less easily modifiable and thus more reliable: for instance, frequent changes of IP address involve either non-trivial techniques as the use of botnets or IP forging through BGP hijacking (due to the necessity of receiving traffic), or negotiations with multiple hosting providers, which may be slow, costly and thus impractical (due to monthly service fees). Another important contribution of this work is to correlate multiple observations of individual torrents along several dimensions, which brings a significant improvement while maintaining a desired lightweight property. Additionally, by exploiting correlations, we expose previously unnoticed seeding behaviors, partly countering common wisdom [4, 5].

A final, notable, contribution of this work beyond the state of the art is a systematic study of the BitTorrent *seedbox ecosystem*, which has been previously only hinted to by [4, 7] but never thoroughly assessed. We point out that, for the time being, we are not interested in addressing whether the content is legitimate or fake. Our analysis of torrent seeding costs holds irrespectively of whether the costs have to be sustained by a producer of real torrents, or by a polluter of fake ones (though this could be easily extended as discussed next).

3 Classification Methodology

Our detection methodology works as follows: we collect data by periodically (every 10 seconds) scraping the "recent torrents" page `http://thepiratebay.se/recent` at ThePiratebay. The page, whose average size is 57KB, is parsed for new torrents[1]. As soon as a new torrent is added to the list, we fetch the torrent (35KB average) and connect to the tracker to get the peer list (1 UDP packet).

We then rely on heuristics, described in what follows, to classify the torrent producer. Since we are not interested in discriminating between real vs fake torrents, we avoid checking whether the torrent exists for several hours/days after it is first injected (since in case the torrent quickly disappears or is banned, this can be used as a reliable indication of fake torrents [4]), though this would be a natural next step.

3.1 Unique seed (S)

In case the content that has been added to ThePirateBay (TPB) is genuinely new, then there are chances that the peer list is reduced to one *unique seed*. As done in previous work [4, 5, 8–11], this simple heuristic allows us to conclude that the seed is likely to be the content originator. Formally, whenever a peer (identified by an endpoint IP:port) matches this simple heuristic for a torrent, we label the peer (and the corresponding torrent) as "S".

However, there are multiple reasons why this heuristic may fail. First, peers may add a torrent to the ThePirateBay that is already published somewhere else, so that the tracker may return multiple peers/seeds ([6] reports this to be often the case for ThePirateBay). Second, content originators may use some strategies to disguise themselves, such as using (i) injecting fake IPs to the tracker, as exploited in [2]; (ii) purposely using multiple ports per IP, to trick monitors in believing the observed IP is that of a NAT box; (iii) using multiple distinct dedicated servers, known as *seedboxes*, per torrent.

Yet, while the "S" classification (*individual observation over single torrents*) may fail due to the above reasons, we argue that *multiple observations over several torrents* can leverage the wealth of additional information to identify the largest fraction of the above instances. More precisely, we propose to exploit correlation in (i) time, (ii) TCP/IP space and (iii) content.

3.2 Correlation in time (T)

We use the classification "T" to denote a correlation in time between swarms. When the tracker returns a list with more than one seeder for a newly injected torrent, a single observation is not enough to isolate the actual content provider(s). However, if a peer that has been previously labeled "S" for another torrent belongs to this list, it is reasonable to assume this peer to be the original content provider of this torrent as well, following a label propagation approach usually done in the classification literature [12]. Notice that in this way we may find peers disguised among other nonexistent peers (e.g., due

[1] Alternatively, we could subscribe and parse the RSS feed on all new torrents `rss.thepiratebay.se/0`

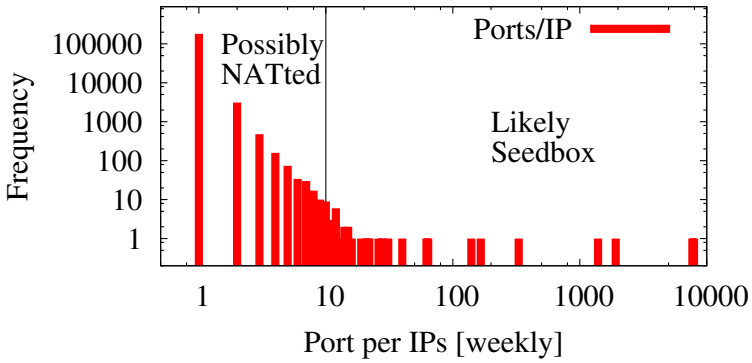

Fig. 1. Number of ports per IP (weekly observation interval)

to fake injected IPs) or among other legitimate peers (e.g., when torrents are added to ThePirateBay after being added to other torrent Website, thus after a swarm is already formed, as observed in [6]).

3.3 Correlation in the TCP/IP space (P)

Let us then address the correlation in the TCP/IP space, and more precisely focus on the TCP port number (denoted as "P" in the following). It is typically considered that whenever multiple ports are observed for a single IP address, this is due to peers behind a NAT, so that the IP address observed is that of the NAT device [4, 5]. Additionally, authors in [5] notice that the number of ports grows proportionally to the number of torrents, suggesting that this is indeed due to multiple users downloading torrents with clients configured with different random ports.

We argue that while this reasoning is correct in most of the cases, some content providers purposely exploit this belief to disguise themselves. To support this argument, Fig. 1 shows the frequency of port number per peer in the typical week in our dataset. It can be seen that while in a large majority of cases, a single TCP port corresponds to a single IP, a number of endpoints deviate from this rule (likely due to NAT) and furthermore some endpoints exhibit *large* deviation (topping to almost 10000 ports behind the same IP). Based on Fig. 1, whenever we find that a newly injected content is seeded by peers with the same IP but different port, we label the torrent (and the IP) as "P".

3.4 Correlation in content space (C)

Finally, let us consider the correlation in the content space (denoted "C"), whereby we may observe multiple endpoints that are disjoint in the IP address space, but that are clustered in the content space.

A common practice in the content diffusion business consists in using multiple CDN services for resilience and performance (e.g., Netflix employs 3 CDN operators). It is

reasonable to assume that similar good practices are adopted by professional BitTorrent content providers. Yet, in case BitTorrent content providers employ multiple seedhosting services, endpoints will be totally unrelated in the IP address space[2]. However, in case such unrelated IPs systematically seed the same (or similar) groups of torrents, they can be easily clustered using typical Community of Interests (CoI) techniques. Of course, BitTorrent consumers can also exhibit some affinity and show a form of clusterization, but the content providers that use multiple seedboxes create communities that drift significantly from human behavior in terms of sizes: seedbox clusters are smaller (in terms of IPs) and much more correlated than human communities. Specifically, a group of torrents and peers is flagged as "C" when they are consistently observed seeding a group of at least C_{min} torrents. We tolerate slight differences in the set composition as the peer list may be partial or all seedboxes from a given cluster may not be active at the same time. This is managed by setting a threshold on a distance between peer lists.

More formally, we denote with $T(p)$ the set of torrents associated to a peer p. $J(p,q)$ is the Jaccard distance between peers $T(p)$ and $T(q)$, that is:

$$J(p,q) := 1 - \frac{|T(p) \cap T(q)|}{|T(p) \cup T(q)|}$$

The peers flagged "C" are selected as follows: we first restrict ourself to the peers p such that $|T(p)| \geq C_{\min}$ for some $C_{\min}$. We then select a maximal peer $p_0 = argmax_p|T(p)|$. If $B(p_0,\epsilon) := \{p : J(p,p_0) \leq \epsilon\}$ contains more than p_0, the whole set is classified as "C", along with corresponding torrents. Then $B(p_0,\epsilon)$ is discarded and we iterate the process.

Though the algorithm complexity may be quadratic in the number of torrents, practical complexity significantly reduces when the input is preliminary sorted by decreasing set size. Thanks to the parameter $C_{\min}$, which we set to 5 based on preliminary tests, input size shrinks (57K to 1875) and so does the running time (2hr to 1min). We also set empirically the maximum Jaccard distance to $\epsilon = 1/C_{\min}$ (i.e., 4/5 torrents in common). We point out that results qualitatively hold for other parameter settings, that we are however unable to fully report due to lack of space.

4 Classification Results

We apply the above heuristic in series to the whole dataset $\mathcal{D}$: we first apply the single-observation heuristic S to obtain a dataset of matching peers and torrent $\mathcal{S}$; we then apply the time-correlation heuristic T, gathering a $\mathcal{T}$ dataset made of a subset of peers from $\mathcal{S}$ and torrents not in $\mathcal{S}$. We next obtain $\mathcal{P}$ by applying the space-correlation heuristic P to $\mathcal{D}\backslash(\mathcal{S} \cup \mathcal{T})$ and finally obtain $\mathcal{C}$ applying C to $\mathcal{D}\backslash(\mathcal{S} \cup \mathcal{T} \cup \mathcal{P})$.

[2] Unless the provider is renting multiple seedboxes of the same hosting facility, so that IPs would share a common prefix. Yet, as correlation in the IP space only covers a subset of the cases we consider, we neglect it in the following.

4.1 Ground truth

To assess the quality of the classification, we perform a reverse DNS lookup of the IP addresses of the peers individuated as content originator. The most reliable way to assess whether a seed (peer) owns torrent data is to download all (some) data chunks and verify their MD5 signatures.

Our university policy forbids us to engage in direct exchange of illegal content via BitTorrent, for which reason we cannot join torrents as a means of verification (besides, the use of super-seeding techniques would possibly void the usefulness of checks based on meta-data).

Yet, we point out that this step is unnecessary in terms of our verification. Recall that we are more interested in detecting "seedboxes" (as opposite to detecting generic "seeds"): this is because the use of a seedbox is correlated with continuous and sustained seeding, typical of professional activities (as opposite to sporadic seeding activities).

We then manually inspect the reverse names to find known seedbox-hosting services (labeled as "sbx" in Tab. 2, see Sec. 5 for an excerpt of this list) or known ISP providers. We also browse the websites of seedbox-hosting and of (many previously unknown) ISP providers as an additional check. Finding Web pages explicitly offering seedbox services for monthly fees (that we study in more details later in the paper) completes the DNS ground truth, making it very reliable.

In corner cases, e.g., whenever the DNS fails to return any result (i.e., no PTR record), we label the peer as *unknown* ("unk" in Tab. 2). When we gather a DNS PTR record but cannot find any *explicit* evidence of seedbox-hosting services, we prefer a conservative approach and do not *not* label the peer as "sbx", even though we cannot find any *explicit* evidence of legitimate ISP and, rather, we do find some hint of suspicious activities (e.g., as13285.net, outo.asia). Otherwise stated, some of these unknown boxes may be actually seedboxes that are simply hard to confirm as such via DNS, but that are instead captured by the above heuristics.

Two further points are worth stressing. First, as in any classification study, result accuracy is bound to the quality of the ground truth. Our interest with this regard is not to precisely calibrate these heuristics with the available ground truth, which leads to the inevitable tradeoff between false alarms (legitimate users believed to be seedboxes) vs. false negatives (seedboxes that remain undetected). Rather, we aim at showing that exploiting correlation along different dimensions enables light-weight seedbox detection.

Second, it could be argued that, reverse DNS queries (ignoring for the time being manual Web page verification) could be used not only as a ground truth, but also as a classification technique (e.g., by means of simple pattern matching on the DNS name). Yet, we point out that this approach would be bound to failure, in that as soon as DNS names would be used to detect (and possibly block) seedboxes, a simple countermeasure would be to remove reverse DNS entries (or use domain names that bear no relationship to BitTorrent seeding).

4.2 Classification performance

Results in terms of torrents, peers and torrent/peer are reported in Tab. 2 for each heuristic, as well as for their combination (boldface, STPC row). In line with previous results [4, 11], we gather that overall only 4K/430K peers are responsible for 35K/57K

Table 2. BitTorrent provider detection performance

	Peers					Torrents					Torrents/peer				
$\mathcal{X}$	sbx	unk	$\lvert\mathcal{X}\rvert$	$\lvert\mathcal{X}\rvert/all$	$sbx/\lvert\mathcal{X}\rvert$	sbx	unk	$\lvert\mathcal{X}\rvert$	$\lvert\mathcal{X}\rvert/all$	$sbx/\lvert\mathcal{X}\rvert$	sbx	unk	$\lvert\mathcal{X}\rvert$	$\lvert\mathcal{X}\rvert/all$	$sbx/\lvert\mathcal{X}\rvert$
S	121	691	2941	*0.66%*	*4%*	4972	4525	14630	*26%*	*34%*	41.09	6.55	4.97	*39*	*8*
T	121	148	925	*0.21%*	*13%*	7207	4832	18355	*32%*	*39%*	44.75	32.65	19.84	*154*	*3*
P	5	0	85	*0.02%*	*6%*	467	0	572	*1%*	*82%*	93.40	0	6.73	*52*	*14*
C	17	25	125	*0.03%*	*14%*	1284	298	1875	*3%*	*68%*	75.53	11.92	15.00	*116*	*5*
STPC	143	716	3151	**0.71%**	**5%**	13930	9655	35432	**62%**	**39%**	97.41	13.48	11.24	**87**	**9**
All			443217					57081					0.13		

torrents; furthermore, about 150 seedboxes are responsible for about 40% of the content.

For any heuristic, we report its recall $|\mathcal{X}|/all$ and seedbox rate computed as $sbx/|\mathcal{X}|$, in terms of both peers and torrents. As for the number of torrents, we find that S accounts for 26% of the observations (less than in [5]), that temporal correlation T is able to explain an additional 32% of torrents while P and C account for a small percentage of torrents (1% and 3% respectively). Interestingly though, we see that the simplest S heuristic has the lowest true positive ratio, as only 34% of torrents can be reconducted to known seedbox via reverse DNS queries, while seed ratio is higher for the other heuristics T=39%, P=82% and C=68%.

4.3 Emerging behaviors

Notably, the very high seed rate for the P heuristic suggests that the use of multiple ports is not uncommon practice in professional seedboxes, debunking a common myth. A possible reason for this behavior is that most seedbox services do not shape the uplink bandwidth among servers in the same rack (see Tab. 3 in Sec. 5 for details): as such, the use of multiple "virtual server" per seed opportunistically increases the amount of aggregate bandwidth that the seedbox is able to obtain (until the point at which the server CPU becomes a bottleneck due to the high number of concurrent applications running on the same physical core). We note that port usage can be either restricted to narrow ranges, or uniformly span a rather large port interval. We examplify such behavior in Fig. 2, where a single IP address is responsible for seeding 39 torrents (grouped by a token of their name in the picture) using over 4000 ports, in generally restricted ranges (unless for the AXXP token). As it clearly emerges from the picture, the IP appears to be running multiple seedboxes, that are likely managed and configured by different individuals (or organizations).

While the low-level details of the virtualization techniques used are not openly advertised by seedbox hosting services (see Sec. 5), available offers however range from managing custom full-blown virtual machines, to simply running pre-configured copies of popular BitTorrent software (e.g., ruTorrent, rtorrent) possibly employing container-based emulation techniques.

Additionally, we also find that 125 peers are organized in groups of 2.5 distinct seedboxes on average. We report a scatter plot of the number of common torrents seeded by different IPs in Fig. 3. To be conservative, let us neglect cases where we observe at most 10 torrents to be seeded by no more than 10 IPs (gray shaded region): still, two

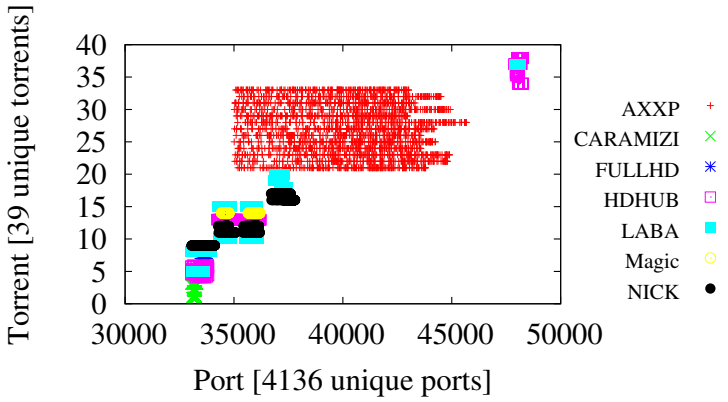

Fig. 2. Emerging behavior: multi-port seedboxes

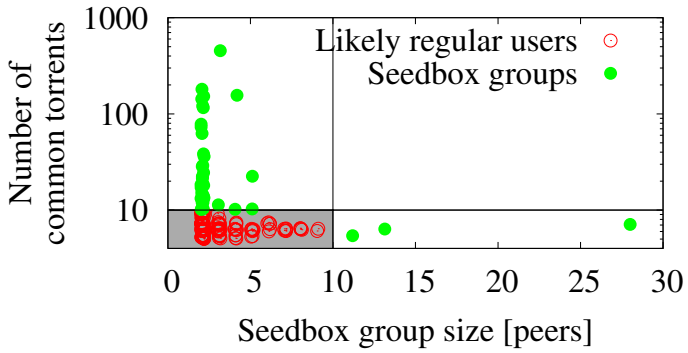

Fig. 3. Emerging behavior: seedbox groups

interesting areas of the plot emerges. In the top-left area (common case), small groups seed a high number of torrent: to point out some relevant example, one of such groups seeds over 300 torrents using 3 seedboxes (seedhost.eu, mshost.ws plus an unknown provider), while another group seeds over 100 torrents with 5 boxes (2 on kimsufi.com, 2 on novalayer.net and 1 on blazinseedboxes.com). In the bottom left area, rather large groups seed a smaller number of torrents (uncommon case): a closer look suggests one of such groups to be a monitor, whereas further investigation would be needed to better understand the structure of these fewer uncommon groups.

4.4 Significant contributors

Finally, going back to Tab. 2, we see that the number of torrents seeded by each peer is significantly higher for seedboxes (40 to 90) than for other peers (5 to 20). Overall, peers matched by any of the S,T,C or P heuristic seed 87 times more torrents than the average peer (0.13 torrent/peer since multiple peers are possibly found per torrent). Furthermore, seedboxes seed 9 times more torrents than other peers in the S,T,C or P sets. Hence, peers exist that have quite serious commitments, requiring a significant

Table 3. Seedbox hosting ecosystem: Popularity, Alexa Ranking and Service Features and Cost

Hosting	**Popularity**			**Alexa**			**Service: min vs. max storage, bandwidth and cost**					
Provider	T/P	T%	P%	Rank	Local	Indegree	minTB	minBW	minUSD	maxTB	maxBW	maxUSD
kimsufi.com	106.58	42.75	17.56	4610	437	2602	0.5	-	13	2	-	52
seed.st	41.44	16.38	17.3	136517	11540	787	0.1	2MBps	6	0.2	0.4TB/mo	40
novalayer.net	78.69	5.95	3.31	173014	0	13	0.2	-	15	4	-	141
nforce.nl	340.33	5.94	0.76	1.99E+07	0	399	0.2	1TB/mo	19.5	6	5TB/mo	221
xirvik.com	24.97	5.08	8.91	267295	170077	76	0.05	-	18	3	-	120
ovh.net	21.6	5.02	10.18	609	103	38844	4	-	90	8	-	117
leaseweb.nl	35.62	3.31	4.07	27388	2549	116	0.5	5TB/mo	39	4	100TB/mo	180
blazinseedboxes.com	35.54	2.69	3.31	958176	165834	34	0.2	-	15	8	-	150
voxility.net	51.43	2.09	1.78	72840	52522	47	-	-	-	-	-	-
seedmybox.com	24.77	1.87	3.31	221282	2126	205	0.3	-	30	1	-	57
leaseweb.com	20.21	1.65	3.56	3157	2642	1496	0.5	5TB/mo	39	4	100TB/mo	180
secureboxes.net	12.05	1.4	5.09	380643	18774	55	0.05	-	6	0.4	-	26
estroweb.in	17.5	1.22	3.05	5683848	0	7	0.12	-	19	0.5	-	49
aireservers.com	16.64	1.06	2.8	6010443	0	4	0.25	-	13	1	-	57
pulsedmedia.com	10.71	1.06	4.33	324688	9467	190	0.1	-	12	1	-	62

amount of work (e.g., to package torrents, transfer them to multiple seed-hosting services, handling the contracts), as noted in [4]. While [4] focuses on prospective gains (e.g., distributing malware in case of fake torrents or advertising a website in case of real torrents), in the next section we reverse the perspective and quantify the cost they incur.

5 Characterization of seedbox hosting service

We first report a detailed characterization of the seedbox-hosting ecosystem in Tab. 3, combining information coming from three sources. The first portion addresses the popularity of the hosting service in our dataset, reporting the percentage of peers $P\%$ and torrents $T\%$ employing each service, as well as the average amount of torrents seeded by peers as an indication of their relative level of activity. The second portion pertains to the popularity in the Web, reporting the global and local Alexa rank and the number of links pointing to the Website of the hosting service. The third portion reports the service SLA (i.e., storage space and bandwidth[3]) and cost (in USD) for low- vs high-end services, gathered by manually browsing the websites. Tab. 3 only reports the bulk of services accounting for 97.5% and 90% of the torrents and peers, and exclude a relatively long tail of unpopular hosting services. Notice that only a limited subset of hosting services that are popular in our trace appears to have been previously listed (e.g., in `http://seedboxgui.de/`). From the above dataset, we can extrapolate that the monthly operational expenditure (only considering the hosting service) of the 150 seedboxes observed during our study is about 33,000 USD – a rather tiny amount. While the average cost per seedbox is lower than 100 USD/mo, expenditure may be higher for providers using multiple boxes.

[3] Many hosting services only report Ethernet access (100Mbps/1Gbps), but the actual (unknown) uplink bandwidth will be shared among hosts

Table 4. Criteria for choice of seedbox hosting service

Pearson	**Alexa**			**Storage**		**Cost**	
ρ	Rank	Local	Indegree	min	max	min	max
T	-0.1	-0.1	0.0	0.0	-0.1	-0.2	-0.2
P	**-0.4**	-0.0	**0.3**	0.2	-0.2	0.0	**-0.4**
T/P	**0.8**	-0.2	-0.1	-0.1	**0.3**	-0.1	**0.5**

Tab. 4 reports a correlation-based analysis of criteria for the choice of seedbox hosting service. We consider what criteria affect this choice by computing the correlation coefficient $\rho(X, Y)$ between pairs of vectors in Tab. 3. As for X we consider either the number of torrents T_s, peers P_s and T_s/P_s using a given seedbox hosting service s. As for Y, we consider the popularity of the hosting service in the Web (measured by Alexa rank, local rank or indegree), or the service features (minimum vs maximum storage space; we exclude the bandwidth for reason exposed above) and cost (minimum vs maximum cost). Mild (0.3-0.5) to strong (above 0.5) correlations are highlighted in boldface in the table.

Though unsurprising, two behaviors emerge from Tab. 4, which are mainly related with the size of the seeding business. First, considering the general seeding professional (row P), we can see that seedbox choice is biased towards popular hosting services that are also popular on the Web. Notice that a high Alexa rank (same for local) implies a low popularity for the hosting service in the Web, while a high Alexa indegree correlates with high popularity. Hence, the Pearson correlation coefficient has opposite meaning for these indexes: i.e, a negative (positive) correlation for Alexa rank (indegree) implies that popular hosting services used in the BitTorrent ecosystem are also popular on the Web. Furthermore, though peer choice is not correlated with the minimum service cost, high maximum cost may however be a deterrent for the average seeding professional $\rho(P, maxUSD) = -0.4$. As a consequence of these two facts, cheaper brands (i.e., kimsufi.com) of hosting services that are popular on the Web (i.e., OVH) are largely popular on the BitTorrent ecosystem as well (i.e., even though the kimsufi.com Website is not popular according to Alexa, the cheaper kimsufi offer is available from the OVH Website, which is instead highly popular).

Second, it is not hard to imagine that professionals seeding hundreds of torrents will have more stringent technical constraints (e.g., in terms of bandwidth or storage capacity). This is precisely what can be observed from row T/P, where choice is correlated to the ratio of the number of torrents seeded by peers: notice indeed the strongly negatively correlated with website popularity $\rho(T/P, Rank) = 0.8, \rho(T/P, Indegree) = -0.1$ but rather correlated with high-end performance and cost $\rho(T/P, maxUSD) = 0.5$. Specifically, few peers having the largest $T/P = 340$ ratio select one of the most costly providers (i.e., nforce.nl). Yet, cost reductions still matter even for professionals. For example, the seedgroup with the second highest average $T/P = 106$ ratio uses kimsufi.com to seed about 3 times less peers than those on nforce.nl. As the maximum cost on kimsufi.com is about 4 times less than on nforce.nl, this relationship is well captured by Pearson's correlation $\rho(T/P, maxUSD) = 0.5$.

We did not take into account additional factors that may bias hosting choice, such as *legal aspects* (e.g., countries more lax in fighting piracy may be preferred [1]), or *physical location* (e.g., seeding EU content from US or China may be inefficient; yet, this is unlikely as many Tab. 3 services offer users the choice of data-center location).

6 Conclusion

We propose a lightweight detection method of content providers in BitTorrent that exploit correlation in time, TCP/IP space and content. Analysis on a large dataset shows that the heuristic reliably detects seeding professionals. Notably, we uncover emerging trends of (i) groups of seedboxes hosted by multiple providers, and (ii) a systematic use of large ranges of TCP ports, that were both undetected by previous methodologies. Finally, we report a preliminary study of the cost incurred by BitTorrent providers, quantifying their operational expenditure for seedbox services. Despite novel insights, this work also leaves some interesting points unanswered. First, detection algorithm could be improved (e.g., by additional sources of correlation such as tokens in the torrent name) and fine-tuned (e.g., Jaccard threshold, using other distance metrics). Second, seedbox groups could be characterized from multiple angles (e.g., countries, AS). Third, metadata could be enriched (e.g., checking whether a torrent has been removed, correlating IP addresses with Spamhaus database).

Acknowledgement

This work started as a student project of the INF570 course at Ecole Polytechnique, was performed at LINCS `http://www.lincs.fr`, and received funding from the EU under the FP7 Grant Agreement n. 318627 (Integrated Project "mPlane").

References

1. S. Alcock and R. Nelson, "Measuring the Impact of the Copyright Amendment Act on Residential DSL Users," in *ACM IMC*, 2012.
2. 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 *USENIX HotSec*, 2008.
3. M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson, "One hop reputations for peer to peer file sharing workloads," in *USENIX NSDI*, 2008.
4. R. Cuevas, M. Kryczka, A. Cuevas, S. Kaune, C. Guerrero, and R. Rejaie, "Is content publishing in BitTorrent altruistic or profit-driven?," in *ACM CoNEXT*, 2010.
5. S. Le Blond, A. Legout, F. Lefessant, W. Dabbous, and M. A. Kaafar, "Spying the world from your laptop: identifying and profiling content providers and big downloaders in BitTorrent," in *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2010.
6. C. Zhang, P. Dhungel, D. Wu, and K. W. Ross, "Unraveling the BitTorrent ecosystem," *IEEE Transactions on Parallel Distributed Systems*, vol. 22, pp. 1164–1177, July 2011.
7. J. Han, S. Kim, T. Chung, T. Kwon, H. Kim, and Y. Choi, "Bundling practice in BitTorrent: what, how, and why," in *ACM SIGMETRICS*, 2012.

8. G. Siganos, J. M. Pujol, and P. Rodriguez, “Monitoring the bittorrent monitors: A bird’s eye view,” in *Passive and Active Measurement (PAM)*, 2009.
9. X. Chen, Y. Jiang, and X. Chu, “Measurements, analysis and modeling of private trackers,” in *IEEE Peer-to-Peer (P2P)*, 2010.
10. K. Avrachenkov, P. Goncalves, A. Legout, and M. Sokol, “Classification of content and users in BitTorrent by semi-supervised learning methods,” in *IEEE IWCMC*, 2012.
11. S. Kim, J. Han, T. Chung, H.-c. Kim, T. T. Kwon, and Y. Choi, “Content publishing and downloading practice in bittorrent,” in *IFIP NETWORKING*, 2012.
12. X. Zhu and Z. Ghahramani, “Learning from labeled and unlabeled data with label propagation,” tech. rep., CMU-CALD-02-107, 2002.