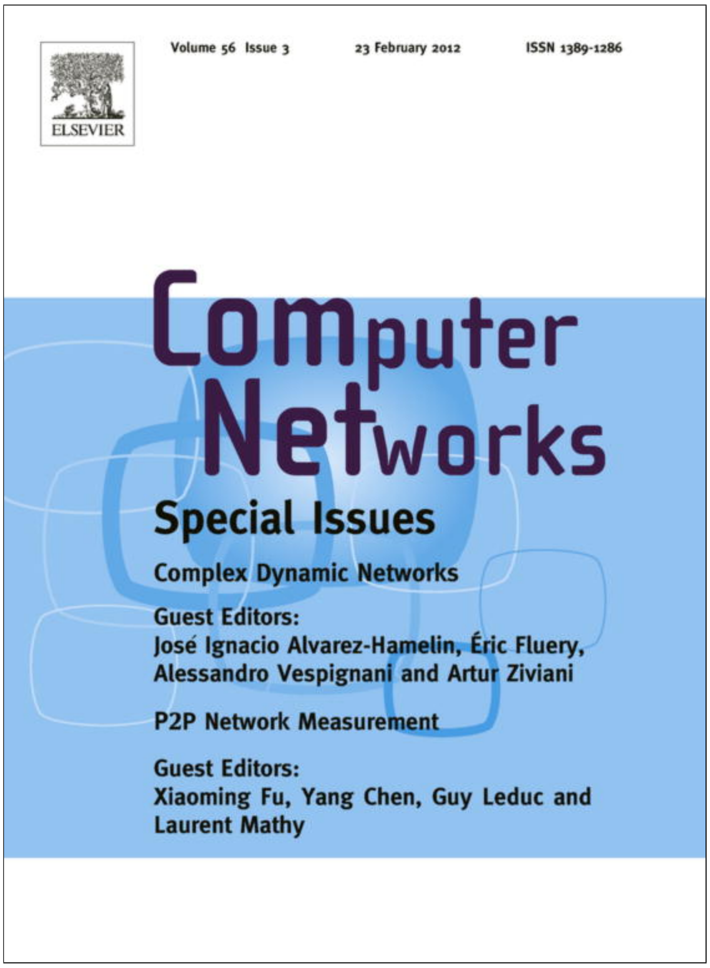

Volume 56 Issue 3
23 February 2012
ISSN 1389-1286
ELSEVIER
Computer Networks
Special Issues
Complex Dynamic Networks
Guest Editors:
José Ignacio Alvarez-Hamelin, Éric Fluery,
Alessandro Vespignani and Artur Ziviani
P2P Network Measurement
Guest Editors:
Xiaoming Fu, Yang Chen, Guy Leduc and
Laurent Mathy

Computer Networks 56 (2012) 1054–1065

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Understanding BitTorrent: A reality check from the ISP's perspective

Matteo Varvello *, Moritz Steiner, Koen Laevens

Bell Labs, Alcatel-Lucent, United States

ARTICLE INFO

Article history:
Received 25 February 2011
Received in revised form 8 September 2011
Accepted 25 October 2011
Available online 27 December 2011

Keywords:
Peer-to-Peer
Measurement
BitTorrent

ABSTRACT

During the last decade, Peer-to-Peer (P2P) applications have been one of the dominant components of Internet traffic. Understanding BitTorrent, by far the most popular P2P protocol for data distribution, is extremely valuable to shed some light on the nature of distributed systems. This paper surveys the existing measurement studies and sets out to verify the acquired corpus of knowledge about BitTorrent by analyzing the largest and most comprehensive data-set so far. We collected BitTorrent traffic at four major European ISPs during 2009 and 2010, a vantage point not yet exploited by previous measurement studies. Our analysis puts into perspective and corroborates several well-known findings, such as that: (1) 20% of the most popular torrents represent more than 95% of the BitTorrent activity, (2) only 1–3% of the BitTorrent traffic stays local, *i.e.*, within an ISP, (3) 4–44% of the BitTorrent traffic could be localized using appropriate locality-awareness techniques, and (4) about 20% of downloads get stalled due to scarcity of content pieces.

1. Introduction

Peer-to-Peer (P2P) is the most emerging Internet-based technology deployed in the last decade. From 2002 to 2007, P2P largely dominated the Internet traffic with a share up to about 60% [20]. Only recently, HTTP overtook P2P becoming responsible for the majority of Internet traffic [14]. This phenomenon is related to the big boom of http-based audio and video streaming services such as YouTube [21].

P2P was first introduced for file-sharing and then rapidly adopted by other services such as live-streaming, on-line gaming, and voice. File-sharing is by far the dominant P2P application, and *BitTorrent* [2,5,17] the most popular P2P protocol. The BitTorrent file-sharing network currently counts millions of users located world-wide that share hundreds of thousands of files on a daily basis [22]. We provide a brief background on BitTorrent in Section 2.

The research community has extensively measured BitTorrent [10,6,9,22] (Section 3). Researchers have instrumented BitTorrent clients to monitor user behaviors, crawled BitTorrent trackers to gather statistics about the number of users and files, and also monitored BitTorrent traffic within user access networks. These studies focused on different aspects of BitTorrent exploiting at best the traces collected. However, each of them has a limited point of view, e.g., tracker-centric or client-centric. Also, they either focus on a large set of users for a short time, or on a small set of users for long time.

This work bridges the gap between previous BitTorrent measurement studies by collecting and analyzing today's largest and most comprehensive BitTorrent data-set. Our motivation is to verify results of previous studies on this data-set collected from authoritative sources, multiple large ISPs.

The methodology we adopt is as follows. First, similarly to Zhang et al. [22] we crawl the most popular websites that index BitTorrent content (PirateBay, BitTorrent, Mini-Nova, IsoHunt, SuprNova and Vuze) in order to gather information about popular public *trackers*. Then, we intercept the traffic exchanged between BitTorrent clients and the popular trackers by setting up filtering rules at ISP border routers. Compared to the related work, our approach targets completeness by exploiting the unique advantage to collect BitTorrent traffic directly at several ISPs. It allows us to monitor the user behavior of a large number of peers,

* Corresponding author. Tel.: +1 7328887258.
E-mail address: matteo.varvello@alcatel-lucent.com (M. Varvello).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.10.031

e.g., their upload and download speed. We are not aware of previous efforts that have compared yet BitTorrent traffic from multiple ISPs. Our methodology and data sets are detailed in Section 4.

Along 2009 and 2010, we intercepted BitTorrent traffic at four major ISPs counting respectively 2.5 million, 1.3 million, 600,000 and 420,000 subscribers. Due to some technical hazards, we collected BitTorrent traces of different durations, between two and four weeks. The comprehensive results of our analysis can be found in Section 5.

Our main findings are as follows:

- *Few files are very popular* – Independently of the monitored ISP, 20% of the most popular files accounted for more than 95% of the BitTorrent traffic.
- *P2P traffic is ISP-unfriendly* – The majority of BitTorrent traffic was consumed by both peering and transit links; only a negligible fraction (1–3%) stayed local within an ISP.
- *There is potential for traffic localization* – 4–44% of the BitTorrent traffic could be localized leveraging a location-aware mechanism [19].
- *There is scarcity of content pieces* – About 20% of user downloads are stalled for more than 30 min and up to more than 10 h due to scarcity of content pieces.
- *High upload/download speed* – The average download speed per ISP varied from 250 to 750 Kbps, whereas the upload speed varied from 180 to 570 Kbps.
- *Users are impatient* – Half of the sessions lasted 10 min, only. Moreover, 96% of the sessions ended before the download was completed.

The rest of this paper is organized as follows; we start with an overview of BitTorrent in Section 2 and describe the related work in Section 3. Section 4 describes our measurement methodology and data collection. Then, Section 5 analyzes our results before we conclude in Section 6.

2. BitTorrent background

This Section presents some background on BitTorrent. We start by describing the BitTorrent vocabulary that is used throughout the paper. Then, we briefly overview the BitTorrent protocol for the sake of a clear understanding of the trace collection and analysis. A complete description of the BitTorrent protocol and main mechanisms can be found in [5,12].

2.1. Vocabulary

Chunk – It is the atomic component of a file. Files are split into equal sized chunks in order to allow parallel download and upload. The chunk size is defined by the file owner, e.g., 256 KB is a typical size. Chunks are randomly distributed among peers.

Seeder – It is a peer that holds a complete copy of a file and offers it for upload.

Leecher – It is a peer that has not yet completed the download of a file. A leecher offers for upload only the chunks it has already downloaded.

Swarm – It is the set of seeders and leechers that participate to a file exchange.

Tracker – It is the server that coordinates a swarm and keeps track of the active peers within a swarm. The tracker does not participate to the file transfer as it does not host the file.

Torrent – It is a.*torrent* metadata file that describes the file exchanged by peers participating in a swarm, e.g., names, sizes and checksums of all chunks of the file, and address of the tracker that coordinates the swarm. In the paper, we use the term torrent also to describe the file pointed to by a.torrent metafile.

Session – It is the uninterrupted time-frame a peer is active in a swarm.

2.2. Overview

BitTorrent focuses on efficient content dissemination and not on content lookup. Content lookup is performed on the Web: a user that aims to download a file gathers the file's torrent through any of the many websites that index torrents in the Internet. Content dissemination is then managed via the *tracker protocol* and the *peer wire protocol*. For a full protocol reference the reader is referred to [3].

The tracker protocol is based on HTTP and defines the interactions between a peer and a tracker. Initially, a peer contacts a tracker in order to retrieve a list of peers involved in the swarm. The tracker answers with the *peer-list*, a random subset of active peers, generally 50 peers. A peer interacts with a tracker regularly while it participates in a swarm in order to send information about the volume of bytes it has downloaded or uploaded. In the paper, we refer to these messages as *reports*. A tracker replies to a report by sending to the peer a new peer-list. The frequency of the reports is regulated by the tracker via the *min_interval* field contained in the tracker replies. Generally, the *min_interval* field is set to 15 min.

The peer wire protocol coordinates the exchange of content in a swarm. The communication between two peers is initiated with a handshake where peers exchange their peer identifier named *peerID*. The peerID uniquely identifies a peer during a BitTorrent session. Once the handshake is accomplished, the two peers exchange the *bitfield*, *i.e.*, a manifest of the chunks they hold. Finally, data transfer is performed using chunks as transfer units triggered by back-to-back requests.

Recently, BitTorrent introduced *decentralized tracking* where any peer can act as a tracker. Decentralized tracking leverages two independent mechanisms: a Distributed Hash Table (DHT) [7] and the peer exchange (PEX) protocol. A DHT is a structured P2P network used for content storage and retrieval. In the BitTorrent context, the DHT is used to store and locate information about which peers hold what files. The PEX is a communication protocol that allows peers participating to the same swarm to share their peer-sets.

3. Related work

The analysis of BitTorrent and its traffic characteristics have been a very fertile research area. We now summarize the pieces of work that are more closely related to our work.

Izal et al. [9] assess the performance of the BitTorrent protocol by measuring the life of a single and very popular torrent over five months (about 180,000 downloads). The authors analyze the tracker log for this popular torrent and instrument a BitTorrent client to collect statistics while participating to this torrent swarm. The authors show the benefits of the tit-for-tat mechanism as well as the great BitTorrent ability to sustain flash-crowd (about 50,000 clients in the first 5 days). We use a similar methodology in order to infer BitTorrent characteristics. However, we rather intercept the client-to-tracker traffic at large scale.

Zhang et al. [22] present results from a comprehensive crawl of the most popular public BitTorrent trackers. The authors crawl five very popular torrent discovery sites (Pirate Bay, Mininova, Torrent Reactor, BTmonster, and Torrent Portal) and identify 4.6 million torrents (although only 1.2 million are active) and 39,000 trackers (although only 728 are active). Successively, they repetitively interrogate the active trackers in order to collect a snapshot of the active peers per torrent at a given time. The analysis of the data collected generates several interesting findings. For example, the authors show that the BitTorrent network is rich in long-tail content (e.g., 82% of the files are hosted by no more than 10 peers), and that μTorrent and the PirateBay are respectively the most popular client and tracker. We leverage a technique similar to [22] in order to identify popular BitTorrent trackers. However, we then exploit this information differently: rather than retrieving the statistics hosted at these trackers, we intercept the traffic directed towards them.

Karagiannis et al. [10] focus on traffic localization in the BitTorrent network. They collect BitTorrent traffic at the edge of a user access network with 20,000 users. Their analysis shows that BitTorrent is extremely ISP unfriendly: 50–90% of locally available content is downloaded from external peers. Also, a localization algorithm enforcing communication among peers located at the same ISP could reduce expensive inter-ISPs traffic of about 40%. The main limitation of this study is the small scale of the data collection. For example, in their traces at most 8 files are concurrently downloaded by at least two users. Beside these measurements, the authors use the traces provided by Izal et al. [9] to draw conclusions valid at a global perspective. One of the main findings is that the localization potential is a function of the provider size (*i.e.*, the number of local users that share a file). Similarly to [10], we also collect BitTorrent traffic directly at the network layer and we do also compute the localization potential.

Cuevas et al. [6] refine the analysis performed in [10] over a larger dataset. They collect traces over one day spanning more than 3.5 million users spread over 9000 ASes. Their analysis shows that very little BitTorrent traffic stays local, and that 10–20% of the transit/unlocal traffic could be eliminated while preserving user download rates. Nevertheless, tolerating a small degradation in user download rates of up to 5% allows to eliminate about 35% of the unlocal traffic. Their analysis shows another very interesting result: BitTorrent traffic localization is more beneficial for fast ISPs than for slow ISPs. However, the amount of traffic that can be localized does not monotonically increase with the speed that an ISP offers to its users. The main limitations of the traces collected by Cuevas et al. are twofold: (1) short duration and (2) incomplete view of the active clients within an ISP.

Similarly to [6], Blond et al. [4] crawled over one day more than 200,000 torrents spanning 6 million unique peers spread among more than 9000 ASes. Based on this trace, the authors evaluate a policy to enforce traffic localization which can reduce inter-ISPs traffic by about 40%.

4. Methodology and data collection

Our methodology consists of intercepting the BitTorrent traffic exchanged between ISP subscribers and popular trackers. In the remainder of this Section, we describe how we identify and rank BitTorrent trackers according to their popularity, and how we capture BitTorrent traffic. Then, we describe the traces collected. Finally, we discuss the limitations of our measurement methodology.

4.1. Tracker discovery

Similarly to [22], we develop an application to discover torrents and trackers in the BitTorrent network. Our application, named *scout*, contains the following modules:

- The *Html Crawler* repetitively queries several popular torrent discovery websites and download all the available torrents.
- The *Torrent Parser* parses the torrents retrieved by the Html Crawler and creates a database entry for each torrent with the following attributes: torrent name, number of seeders, number of leechers, tracker URLs and IP addresses.
- The *Updater* periodically resolves tracker URLs, verifies their reachability, and updates their database entries. In case a tracker URL cannot be resolved in the latest run of update, we keep its latest IP address resolved.

We launch the scout towards the following websites: PirateBay, BitTorrent, MiniNova, IsoHunt, SuprNova and Vuze. We perform a partial crawl of these websites the week before the set up of the data collection at an ISP. At each crawl, the scout generates a data-set of about 300,000 torrents, from which we extract the URLs and IP addresses of 4000 trackers. Then, we rank the trackers according to the number of *recent*, less than one month old, and *popular*, more than five seeders, torrents. In the following, we use this tracker list to drive the BitTorrent traffic collection; this traffic might or not be associated to any of the 300,000 torrents we crawled from the Web according to their popularity evolution at the ISP where traffic is captured.

4.2. Data capturing

We capture the messages exchanged between BitTorrent clients and the popular trackers using filtering rules at ISP border routers. The filtering rules match the IP

Table 1
Trace summary; [ISP-A, ISP-B, ISP-C, ISP-D].

	Duration (days)	Size (TBytes)	Avg. traffic rate (Mbps)	Peak traffic rate (Mbps)	Monitored subscribers
ISP A	26	3	7	25	1,300,000
ISP B	18	4.1	19	45	420,000
ISP C	16	0.13	0.7	1.6	90,000
ISP D	14	1.8	8.6	25	2,520,000

addresses of the popular trackers as provided by the scout application. The verified assumption is that popular trackers reside at ISPs where we do not collect traces.

The client-to-tracker messages that we intercept contain both peer-list and reports (cf. Section 2.2) for each ISP subscriber that participates to the BitTorrent network. These messages are exchanged with a frequency of 15 min. In the paper, we assume that BitTorrent clients correctly report to the trackers and gracefully leave the system, *i.e.*, they inform the trackers they are connected to when they leave the system. For this reason, the resolution of our traces can be higher than 15 min in few cases (cf. Section 5).

We now describe the data collected. For anonymity reasons, we refer to the monitored ISPs simply as *ISP-A*, *ISP-B*, *ISP-C* and *ISP-D*, and we do not reveal the precise dates when the data collections were performed. Table 1 summarizes the main characteristics of the traces collected: trace duration and size, average and peak traffic rate, total number of monitored subscribers.

- *ISP-A*: We monitored about 1.3 million subscribers, *i.e.*, about 70% of the total subscribers, for 26 consecutive days. We set-up 740 filtering rules at the ISP border routers and captured 3 TBytes of BitTorrent traffic at an average rate of 7 Mbps, with peaks up to 25 Mbps. In total, 4005 million messages were captured, *i.e.*, an average of about two thousand messages per second. Exceptionally and thanks to the cooperation of ISP-A, we also obtained some statistics about the underlying traffic, e.g., HTTP and Usenet.
- *ISP-B*: We monitored about 420,000 subscribers, *i.e.*, 100% of the total subscribers, for 18 consecutive days. We set-up 3429 filtering rules at the ISP border routers and captured more than 4 TBytes of BitTorrent traffic at an average rate of 19 Mbps, with peaks up to 45 Mbps. In total, 1092 million messages were captured, *i.e.*, an average of about 700 messages per second.
- *ISP-C*: We monitored about 90,000 subscribers, *i.e.*, 3.75% of the total subscribers, for 16 consecutive days. We set-up 2000 filtering rules at the ISP border routers and captured 130 GBytes of BitTorrent traffic at an average rate of 0.7 Mbps, with peaks up to 1.6 Mbps. In total 60 million messages were captured, *i.e.*, an average of about 43 messages per second.
- *ISP-D*: This ISP counts approximately 520,000 fixed subscribers and 2 million mobile subscribers. We monitored 100% of both fixed and mobile subscribers for 14 consecutive days. We set-up 2349 filtering rules at the ISP border routers and captured about 2 TBytes of BitTorrent traffic at an average rate of almost 9 Mbps, with peaks up to 25 Mbps. In total 1750 million messages were captured, *i.e.*, an average of about one thousand messages per second.

4.3. Limitations

The collection of BitTorrent traffic at ISP scale is an extremely challenging task. In the following, we discuss the limitations of our measurement methodology by focusing on their root causes as well as on their impact on the data analysis.

The major limitation of our measurement methodology is that we only intercept the BitTorrent traffic exchanged between peers and trackers, and we do not collect the traffic directly exchanged between peers, *i.e.*, chunk requests, data transfers and peer exchanges. It follows that the statistics we discuss in the paper are derived from the client-to-tracker traffic and do not refer to the actual data transferred over the network. However, capturing the traffic directly exchanged between BitTorrent clients at ISP scale is unfeasible. In fact, this data collection would require the set-up of hundreds of thousand filtering rules at each ISP router (not only at the border routers). However, routers currently only support between few hundreds and few thousand filtering rules according to their make.

The previous limitation has also another consequence. Peers in BitTorrent learn about other peers not only from the tracker but also from the DHT and the direct peer exchanges (cf. Section 2.2). The usage of the DHT is becoming a common practice among BitTorrent users. For example, we showed in [18] that about 40% of BitTorrent users at a large ISP in Europe prefer the DHT to the trackers. Both DHT and direct peer exchanges are not captured by our methodology since we monitor the client-to-tracker traffic, only. The consequences of this limitation are twofold. First, even more BitTorrent traffic compared to our observations may be present within an ISP. Second, we might underestimate the peer-sets and their evolution over time.

Another consequence of the previous limitation is that we cannot precisely re-construct how a swarm evolves over time, *i.e.*, at which peers each chunk is replicated at a given time. In order to gather this knowledge, we would have to monitor the bitmaps exchanged between all peers that participate in a swarm. For the same reason as above, this is unfeasible with our methodology. This limitation affects the analysis of the traffic decomposition in local, peering and transit (cf. Section 5.1.2) as well as the analysis of the localization benefits (cf. Section 5.2.2).

A minor limitation of our study is the asynchronous data collection, *i.e.*, traces are collected at four different points in time during 2009 and 2010. Thus, when comparing BitTorrent characteristics among the ISPs we refer to different time-frames. This is a limitation due to the fact that torrent popularity as well as user behavior may vary over time. In addition, at the end of 2009 the largest available tracker, the Piratebay, shut off its service completely.

Finally, our methodology relies on the assumption that BitTorrent clients correctly report to the trackers. As shown in [16], several BitTorrent implementations falsely report to the tracker. While we cannot identify and thus

Table 2
BitTorrent traffic characteristics; [ISP-A, ISP-B, ISP-C, ISP-D].

	Sessions (M)	Swarms (K)	Downloads completed (K)	Download (TBytes)	Upload (TBytes)
ISP-A	40.5	787.1	1803.7	1694.0	1101
ISP-B	34.7	977.3	2235.6	1990.0	1525
ISP-C	1.7	94.4	109.9	102.0	55.6
ISP-D	16	275	452.6	493.4	465.6

filter misbehaving clients, we believe that such lying and abuse is not widespread in the BitTorrent community.

5. Results

In this Section, we analyze the BitTorrent traces collected. Our analysis centers around the following entities: *traffic*, *torrents*, *users* and *trackers*. For each entity, we first describe its global characteristics and then focus on a fine-grain analysis.

5.1. Traffic

Table 2 summarizes some global statistics about the BitTorrent traffic as observed in our traces: sessions, swarms, downloads completed, TBytes of data downloaded and uploaded. In total, our traces contain about 93 million sessions during which 4.6 millions files are downloaded which generate more than 4 PBytes of traffic on the downstream and 3 PBytes in the upstream. Among the different ISPs, the figures reported in Table 2 show some large variations. This is due to the fact that each trace has a different length and spans a different number of subscribers. However, even the smallest data-set (*i.e.*, the one collected at ISP-C) contains more than 100,000 completed downloads which generate about 100 TBytes of traffic on the downstream and more than 50 TBytes on the upstream.

5.1.1. Goodput analysis

We aim to quantify the volume of *useful bytes* moved by BitTorrent within each ISP. This means, we aim to compute the BitTorrent *goodput*, not taking into account the transmission protocol overhead (TCP) as well as the BitTorrent protocol overhead (signaling messages). To do so, we use the information about the bytes downloaded and uploaded by each peer as reported to the trackers with the reports. Fig. 1(a) and (b) shows the weekly pattern for respectively the download and upload goodput. Since the datasets do not overlap in time (cf. Section 4.3), for each ISP we pick a different week starting on Monday 00:00 CET and going until Sunday 24:00 CET.

Globally, Fig. 1 shows a daily cycle typical of Internet-based applications: low activity during the early morning, increase towards the end of the day, and then fall off during the night. Precisely, peak user activity is observed between 8pm and 10pm, whereas lowest activity is observed in the early morning around 6am. Despite at each ISP we measure different absolute numbers for the goodput, the average ratio between minimum and peak goodput is very similar, e.g., ranging from a minimum of 0.375 at ISP-D to a maximum of 0.44 at ISP-C. We also notice that users are more active during the day in the weekend compared to working days. The download goodput (Fig. 1(a)) exhibits a greater daily variation compared to the upload goodput (Fig. 1(b)). The difference between the lows and the highs for the download goodput is a factor of three, while it is only a factor of two for the upload goodput. Furthermore, the distinction between working and non-working days is greater for the download than it is for the upload goodput.

We now focus on each ISP. The minimum download and upload goodput are measured at ISP-C where we monitor the minimum number of subscribers, *i.e.*, 90,000 subscribers. Although not clearly visible in Fig. 1, the goodput measured at ISP-C stays between 200 Mbps and 1 Gbps for the download and between 200 Mbps and 400 Mbps for the upload. The highest download goodput is measured at ISP-B and it is between 5 and 16 Gbps, *i.e.*, about 50% more than ISP-A and 400% more than ISP-D. A similar result is observable for the upload goodput.

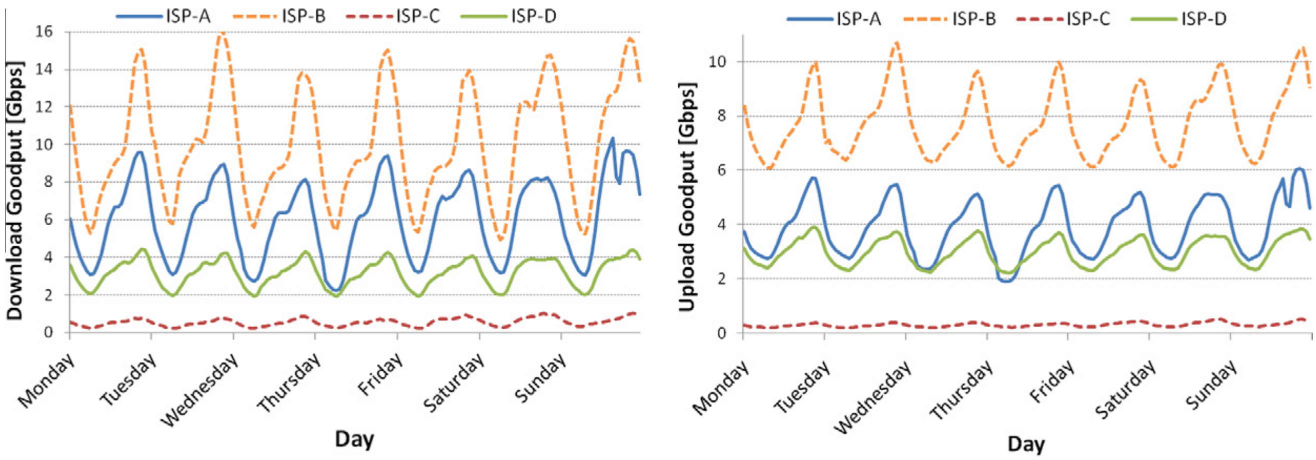
Fig. 1(a) and (b) shows two significant drops for both the download and upload goodput measured at ISP-A during Sunday afternoon (extreme right of the figures). In order to gather a better understanding of this phenomenon, Fig. 2 plots the evolution hour-by-hour of the number of messages exchanged between ISP-A subscribers and the four most popular trackers on that Sunday. Fig. 2 shows two interesting results. First, PirateBay[1] owns the four trackers that attract the majority of the traffic from ISP-A, e.g., up to 415,000 messages at peak hour. Second, the traffic directed towards these trackers dropped by almost 70% between 5pm and 7pm. Although not reported in Fig. 2, our data exhibit the same behavior for all other trackers. Thanks to the cooperation with ISP-A, we found out that this happened due to a small outage associated to a DNS update.

5.1.2. Traffic breakdown

The network agnostic nature of BitTorrent does not take into account from "where" a peer retrieves a file. We now analyze our traces in order to quantify how much BitTorrent traffic stays *local*, *i.e.*, within an ISP. To do so, we use the peer-list that a BitTorrent client receives from the tracker. For a peer *P*, we label each IP address contained in *P*'s peer-list as *local* when it is owned by the same ISP as *P*, *peering* when it is owned by an ISP that shares a peering link with *P*'s ISP, and *transit* when it is owned by an ISP connected to *P*'s ISP through a transit link.

Fig. 3(a) shows for each ISP the average number of local, peering and transit IP addresses returned by the trackers during the monitoring period. Note that the standard deviation for each of the computed averages is comprised between 0.2% and 1%. If we assume that a peer retrieves content uniformly from the peers in its peer-list, Fig. 3(a) clearly indicates that the BitTorrent traffic is extremely non-local: overall, less than 3% of the traffic stays local. Even worse, the majority of the traffic is consumed using transit links, from a minimum of 42% for ISP-C to a maximum of 91% for ISP-B. These results confirm the figures reported in [6].

[1] Currently, the PirateBay only functions as an indexing site and no longer as a tracker.



(a) Download goodput.

(b) Upload goodput.

Fig. 1. Global traffic analysis; [One week]; [ISP-A, ISP-B, ISP-C, ISP-D].

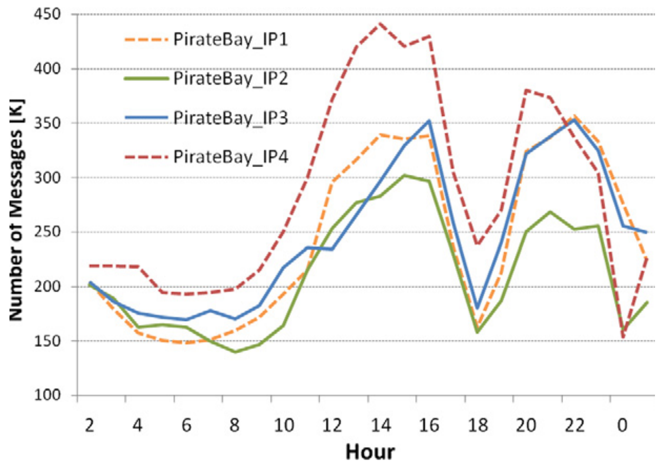
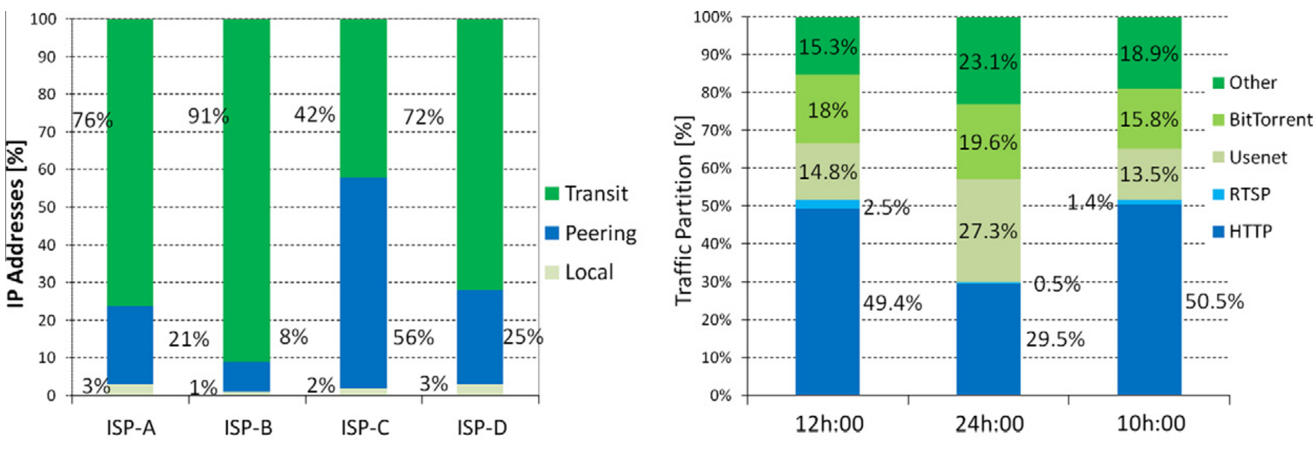


Fig. 2. Number of messages towards popular trackers; [Sunday]; [ISP-A].

We now aim to answer the following questions: *what is the impact of Peer-to-Peer on the total traffic observed within an ISP?* To do so, we derive an estimation of the total BitTorrent traffic starting from the goodput (cf. Section 5.1.1). Precisely, we apply the following corrections: (1) additional 10% of traffic due to the TCP overhead [8] and (2) additional 2% of traffic due to the signaling messages exchanged between peers [11]. Then, with the cooperation of ISP-A, we compare our estimation of BitTorrent traffic with the statistics about underlying traffic as released by ISP-A.

Fig. 3(b) shows three representative snapshots of the traffic decomposition at ISP-A obtained along two days at 12h00, 24h00 and 10h00, respectively. Three main traffic components emerge: HTTP, BitTorrent and Usenet. HTTP traffic is currently the dominating traffic with a share between 29.5% and 50.5%. BitTorrent and Usenet share a comparable fraction of the traffic, respectively 15.8–19.6% for BitTorrent and 13.5–27.3% for Usenet. Finally, a large fraction of the traffic (between 15% and 23%) cannot be identified. This traffic encompasses both applications that generate few data such as email, as well as more data intensive applications such as DHT-enabled BitTorrent clients. Note that the latter applications would account for



(a) Average local, peering and transit IPs ; [ISP-A; ISP-B; ISP-C; ISP-D].

(b) Protocol analysis ; [ISP-A]

Fig. 3. Traffic breakdown.

additional BitTorrent traffic; however, their detection is extremely challenging (cf. Section 4.3).

5.2. Torrents

Globally, our traces span about 450,000 torrents at ISP-A, 640,000 torrents at ISP-B, 74,000 torrents at ISP-C and 213,000 torrents at ISP-D. The number of *active* torrents at a given time, *i.e.*, torrents with at least one requesting peer, follows the daily cycle described in Fig. 1. The largest number of concurrent active torrents is observed at ISP-B, with a minimum of 120,000 torrents in the early morning and a maximum of 180,000 torrents between 8pm and 10pm. The minimum number of concurrent active torrents is measured at ISP-D, *i.e.*, between 5000 and 9000 active torrents.

In the remainder of this Section, we analyze *torrent popularity* and the *local download opportunities*.

5.2.1. Torrent popularity

We start with a simple analysis of the torrent popularity: we rank the torrents according to the volume of download traffic they generate. Thus, the torrents whose chunks are transferred most frequently are most popular, whereas the torrents whose chunks are transferred least frequently are unpopular. Accordingly, we compute the CDF of the torrent popularity at the four ISPs (Fig. 4). We find that the distribution of the torrent popularity does not significantly change among the four ISPs. For this reason, Fig. 4 only shows the torrent popularity curve derived from the traces collected at ISP-A. Fig. 4 shows that about 90% of the BitTorrent activity is concentrated on the top 10% most popular torrents (out of about 450,000 torrents monitored at ISP-A). For example, the top 50 most popular torrents account for 6.3% of the download activity.

We now refine the analysis of torrent popularity at each ISP in order to take into account the time component: we compute for each torrent how many peers request it over time. Thus, the more frequently a torrent is requested, the more it is popular and vice versa. We proceed as follows. We pick one week of traffic at the four ISPs and we divide it into fixed time intervals. Then, for each torrent and time interval we compute the number of peers involved in the swarm. We use the reports as indicators of peer (in) activity, e.g., if we do not see a report to the tracker within two successive time intervals we conclude that the peer abandoned the swarm. Accordingly, we set the time interval to 15 min as this value is used by most BitTorrent clients to transmit reports (cf. Section 2.2). As a result of our analysis, we obtain for each torrent the evolution of its popularity over time, *i.e.*, the number of peers interested in it at a given time. Since we cannot plot thousands of popularity curves for each ISP, in the following we simply discuss our findings. The first interesting observation (common to the four ISPs) is that most of the time the majority of the torrents are very unpopular. Precisely, 75% of the torrents attract no single peer from each ISP during half of the trace durations. Even more surprisingly, 99% of the torrents attract a maximum of two concurrent peers located at the same ISP during half of the trace durations. Less than 0.1% of the torrents have more than 50 local peers during half of the trace duration. This means that peers within the same ISP are highly desynchronized, and the chance that peers request the same torrent at the same time is very low, even for large ISPs. However, not all torrents are so unpopular. For example, the most popular torrents in ISP-A, ISP-B, ISP-C and ISP-D have respectively 529, 297, 19, 457 concurrent peers available during half of the trace durations.

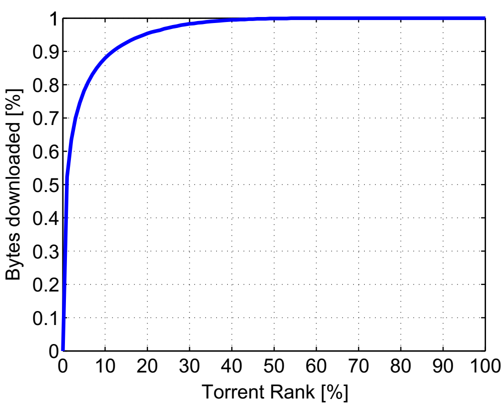


Fig. 4. Torrent popularity; [ISP-A].

5.2.2. Torrent localization

We now exploit the data about torrent popularity in order to answer a fundamental question for the research community as well as for the ISPs: *how much BitTorrent traffic can be effectively consumed within an ISP without requiring help from peers located at other ISPs?* Because we have no information about the difference in download speed that can be achieved from local and remote peers, we reformulate the question as follows: *how many local peers can be returned by a localization mechanism à la P4P [19] to a peer that wants to start a download*?

We proceed as follows. We assume that when a peer *P* downloads a file, it receives a list of 50 peers every 15 min from a localization enabled tracker. These peers represent the *total download opportunities* for *P*. Among these peers, the peers located at the same ISP as *P* represent the *local download opportunities* for *P*. The tracker prioritizes local peers, when available. In the following, we refer to the local download opportunities as the fraction of total download opportunities which are local. Note that if we make the strong assumption that a peer uniformly downloads from all peers in its peer-set, the local download opportunities match the fraction of transit traffic that can be saved and traded for intra-ISP traffic.

Fig. 5 shows the CDF of the local download opportunities computed for the four ISPs. The Figure shows a very similar behavior for ISP-A, ISP-B and ISP-C: 90% of the torrents have almost no local download opportunities, while the remaining 10% of the torrents have between 4 and 10% local download opportunities for the respective ISPs. Conversely, at ISP-D 96% of the torrents have almost no local download opportunities, but the remaining 4% of the torrents contribute to reach an overall local download opportunities of 44%! Further analysis of our traces shows that this happens because seeders located at ISP-D tend to

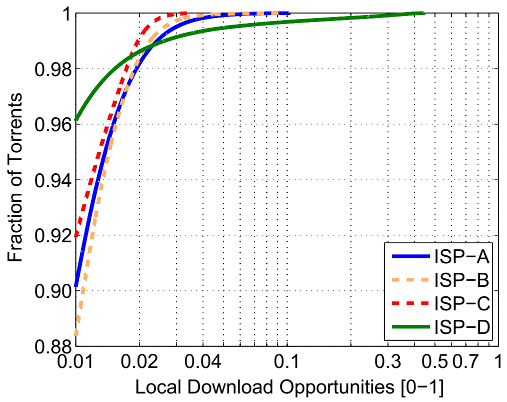


Fig. 5. CDF of the local download opportunities; [ISP-A, ISP-B, ISP-C, ISP-D].

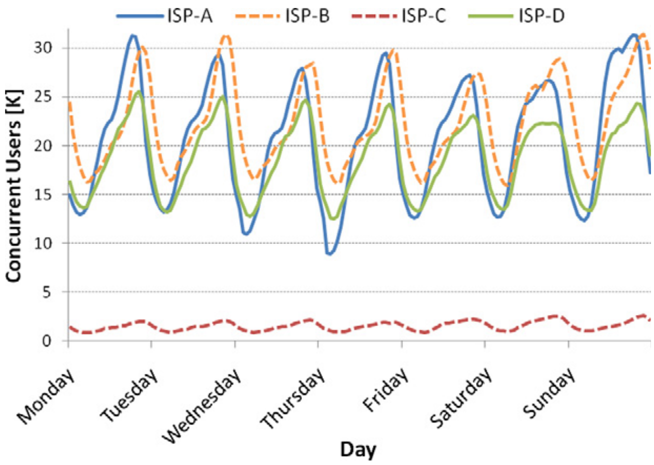


Fig. 6. Concurrent users; [ISP-A, ISP-B, ISP-C, ISP-D].

remain on-line for a long time (cf. Section 5.3.3). The take-home result of this analysis is that a localization mechanism should focus on the few locally popular torrents as the remaining torrents can only be localized to a very small extent.

5.3. Users

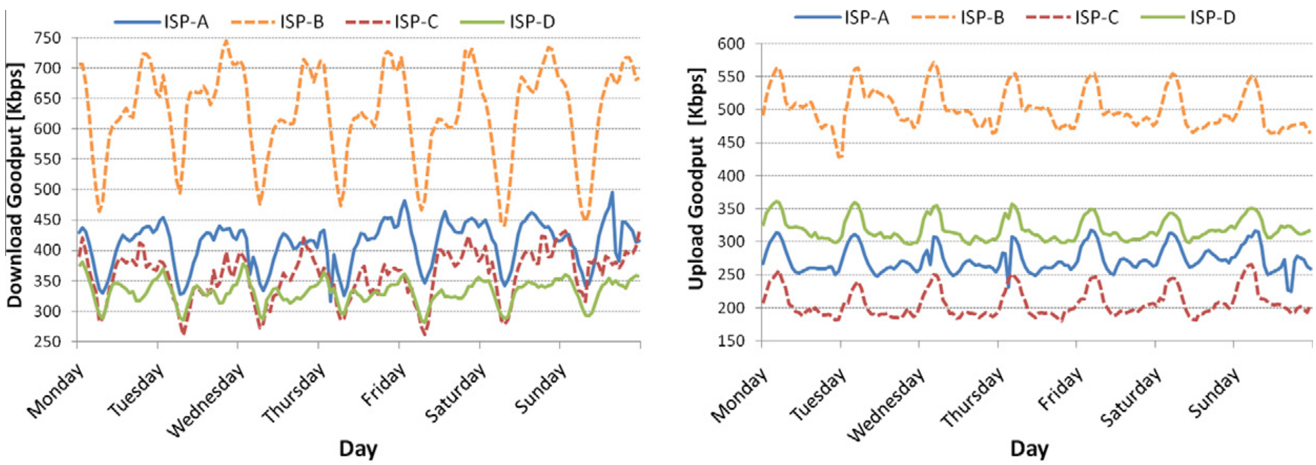
We identify a unique BitTorrent user via the peerID over a time period of 15 min, *i.e.*, the frequency of reports transmitted to the trackers (cf. Section 2.2). Accordingly, Fig. 6 shows the evolution over time of the number of concurrent BitTorrent users as observed at the four ISPs. ISP-C contributes with the minimum number of concurrent users, e.g., 1000 users at early morning and 3000 users during the evening. ISP-A, ISP-B and ISP-D show a much more similar trend, ranging from a minimum of 10,000 users up to more than 30,000 users.

In the remainder of this Section, we zoom into the analysis of BitTorrent user characteristics. First, we analyze user upload and download goodput and BitTorrent clients popularity. Then, we analyze in detail user session characteristics.

5.3.1. Goodput analysis

We exploit the peer reports in order to measure a user download and upload goodput, *i.e.*, the effective upload and download speed perceived by the user. We compare two successive user reports in a time interval t in order to obtain the volume of byte downloaded or uploaded by a user during t. Then, we compute the download and upload goodput simply by dividing these figures by t.

Fig. 7(a) and (b) shows the weekly pattern of the upload and download goodput computed at the four ISPs. The goodput is averaged among the active users hour-by-hour. The average download and upload goodput per user follow the same cycle as the global download and upload goodput (cf. Section 5.1.1). In fact, as the number of concurrent users increases, the available download and upload resources in the P2P network increase, too. This is a concrete example of the self-scalable property typical of P2P networks. The most visible result is that clients from ISP-B have an extremely large download and upload goodput, up to respectively 750 and 550 Kbps. Users at ISP-A, ISP-C and ISP-D share a very similar download goodput between 250 and 450 Kbps.



(a) Download goodput. (b) Upload goodput.

Fig. 7. Goodput analysis; [ISP-A; ISP-B; ISP-C; ISP-D]; [One week].

In [6], Cuevas et al. make some assumptions on the BitTorrent users upload rate in order to derive some insights on the traffic localization potential. In the absence of ISP-level information, they use data from the Ookla Speedtest service [15] to derive the distribution of BitTorrent users upload rate per country. Accordingly, they assume that the median upload speed per subscriber is lower than 610 Kbps for 80% of the countries. At the monitored ISPs, we measure a median upload rate of 268, 495, 199 and 313 Kbps, respectively, thus somehow validating the assumptions used in [6].

5.3.2. Client analysis

Multiple BitTorrent clients are freely available on the Internet, and we aim to estimate their popularity. We identify the client used by each peer via the peerID field (cf. Section 2.2). As already observed by Zhang et al. [22], we find that μTorrent is by far the most popular client, and it is used by half of the BitTorrent users independently of the ISP. Azureus [1] and the BitTorrent client [2] are respectively the second and third most popular clients, and they are used by about one third of the BitTorrent population. All other clients have a limited usage, *i.e.*, less or about 5% of the BitTorrent population, with the exception of BitComet in ISP-D which accounts for 12% of the users.

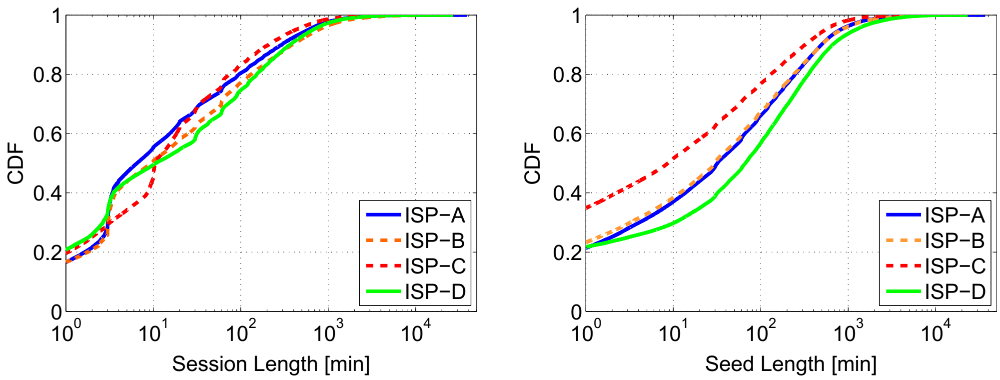
5.3.3. Session analysis

A session is the uninterrupted time-period a user participates in a swarm, *i.e.*, to the download/upload of a torrent (cf. Section 2.1). For example, if a user starts a download in the morning, shuts down her client, and completes the download in the evening, this results in two different sessions. Similarly, four BitTorrent clients that run at the same home network of a single ISP subscriber and download the same torrent generate four independent sessions. A peer that connects to multiple trackers for a given torrent is nevertheless participating in the respective swarm only once, and therefore this behavior results in one session only. Note that in our traces we find that 95% of the users do not participate in multiple torrents simultaneously, *i.e.*, only 5% of the users are active in multiple parallel sessions.

Fig. 8(a) plots the CDF of the user session length measured at the four ISPs. The four distributions exhibit a very similar trend indicating that users across the four ISPs share similar habits. For example, the median session length is 7 min for users at ISP-A, and 10 min for users at the other ISPs. 20% of the sessions last between one hour (ISP-C) and almost 2 h (ISP-D). Nevertheless, 1% of the sessions are extremely long, *i.e.*, ≈16 h, with a peak session length measured at ISP-A of about 28 days.

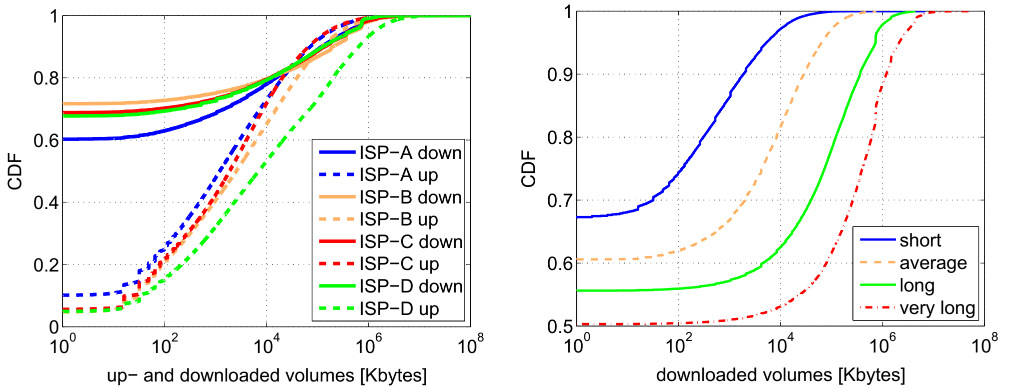
The majority of the sessions (93–96%) do not generate a new seeder, *i.e.*, they end without the file being completely downloaded. For the remaining sessions, we compute the *seeder length*, *i.e.*, the time a client stays on-line after the end of its download. Fig. 8(b) shows the CDF of the seeder length measured at the four ISPs. The seeder length trends measured at the four ISPs are less homogeneous than the session length trends (Fig. 8(a)). Users at different ISPs have different seeder lengths, e.g., ISP-C subscribers have short seeder lengths (a median seeder length of 9 min), whereas ISP-B and ISP-A subscribers have much longer seeder lengths (a median seeder length of respectively 30 and 32 min). The longest median seeder length are observed for the users of ISP-D with 67 min. The longer seeder times we observe for ISP-D lead to more active torrents compared to the other ISPs (cf. Section 5.2.1). The seeder time does also influence the localization potential (cf. Section 5.2.2). Finally, between 2% and 5% of the seeder length values are extremely long, *i.e.*, between 16 h and 28 days. This observation indicates that the 28 day long session measured at ISP-A refers to a very persistent seeder.

We now investigate how much data each leecher uploads and downloads per session. We focus on the leechers since seeders have a very predictable behavior: they only upload content as they already hold a copy of the file. Fig. 9(a) plots the CDF of the uploaded and downloaded bytes per leecher session at the four ISPs. Globally, the shapes of the distributions for both the uploaded and downloaded bytes are similar among the different ISPs. We first focus on the uploaded bytes. Accordingly, in 5–10% of the leecher sessions users do not upload any data. These sessions can be associated to *free-riders*, e.g., users who run a modified BitTorrent client [13]. However, some users are very generous and upload up to 1 GB in a single session. We now focus on the downloaded bytes. Similarly to the uploaded bytes, in 1% of the sessions users download large amounts



(a) CDF of the user session length. (b) CDF of the seeder session length.

Fig. 8. Session analysis; [ISP-A; ISP-B; ISP-C; ISP-D].



(a) CDF of the uploaded and downloaded KBytes per session ; [ISP-A; ISP-B; ISP-C; ISP-D].

(b) CDF of the downloaded KBytes per session ; ISP-A ; session=[short ; average ; long ; very long].

Fig. 9. Session analysis.

Table 3
Statistics per profile.

Profile	Total IPs (K)	Active IPs (% of profile)	Avg. download (MBytes)	Avg. up-to-down ratio
Very fast	26.5	28	723	0.74
Fast	407.3	17	521	0.73
Normal	600.1	14	412	0.6
Slow	897.7	6	242	0.47

of data, up to 0.9 GB at ISP-D and 1.4 GB at the remaining ISPs. However, 60–70% of the leecher sessions have no download. This means users start a torrent and do not see any progress during the time their client is running. In order to understand the latest observation, we analyze the evolution of the download volumes as a function of the user session length. Since all ISPs show similar behaviors (Fig. 9(a) and Fig. 8), we only focus on ISP-A.

We proceed as follows. We label user sessions as *short* when they last less than 5 min; *average*, when they last between 5 and 30 min; *long*, when they last between 30 min and 10 h, and *very long* when they last more than 10 h. Fig. 9(b) shows the CDF of the downloaded volume for short, average, long and very long sessions. In 67% of the short sessions, users do not download any content; this can be due to the short session length which does not allow the BitTorrent client to locate and obtain any chunk. However, the fraction of sessions that exhibit no download only reduce to 60%, 55% and 50% for average, long and very long sessions, respectively. Given average, long and very long sessions account for about 40% of the total sessions (Fig. 8(a)), it follows that about 20% of user downloads are stalled for more than 30 min and up to more than 10 h. We attribute this behavior to a scarcity of chunks, i.e., no chunk is available at the time that a user request comes.

5.3.4. Subscription profile and BitTorrent usage

Is there any correlation between a peer Internet access (upload and download bandwidth) and its behavior in the BitTorrent network? We received from ISP-A the information about the download bandwidth assigned to each IP address we monitored. Four different profiles of subscribers are possible: *very fast* (20 Mbps), *fast* (8–12 Mbps), *normal* (4–6 Mbps) and *slow* (0.4–1.5 Mbps).

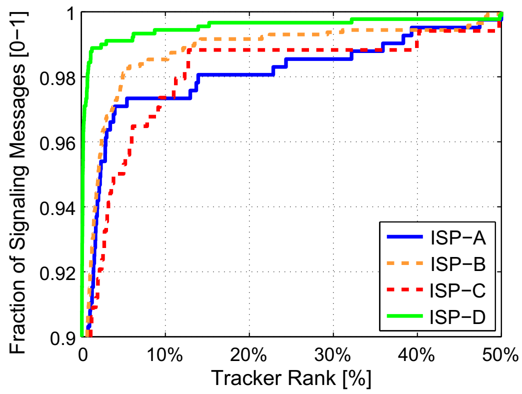


Fig. 10. Tracker popularity; [ISP-A, ISP-B, ISP-C, ISP-D].

Table 3 reports some general statistics regarding the four different subscriber profiles. The majority of the 1.9 million subscribers of ISP-A are profiled as normal and slow subscribers. Among these, 14% and 6% of the subscribers participate to the BitTorrent network, respectively (*i.e.*, connect at least once during the 26 days long measurement period). Comparatively, 17% of the fast subscribers and 28% of the very fast subscribers are active in the BitTorrent network. Thus, high-bandwidth subscribers (very fast and fast profiles) are more likely to participate in the BitTorrent network compared to the low-bandwidth subscribers (normal and slow profiles). Similarly, high-bandwidth subscribers tend to download more compared to the low-bandwidth subscribers, e.g., an average of 242 MBytes per user and per day for the slow peers versus 723 MBytes for the very fast peers. Finally, the average upload-to-download tends to decrease as the subscriber download bandwidth decreases. For example, slow subscribers download two times more than they upload (*i.e.*, the upload-to-download ratio equals 0.47), whereas fast subscribers only download about 35% more than they upload (*i.e.*, the upload-to-download ratio equals 0.73).

5.4. Trackers

At each ISP, we monitor a different subset of the 4,000 trackers discovered (cf. Section 4.1) due to the limited and different number of filtering rules available at each ISP border routers (cf. Section 4.3). However, we find that only a subset of the tracker IPs inserted as filtering rules are active, *i.e.*, exchange any message with the ISP subscribers. Precisely, 409 unique trackers are active at ISP-A, 1431 at ISP-B, 337 at ISP-C and 895 at ISP-D.

We aim to understand trackers' *popularity*. Similarly to Section 5.2, we rank the trackers according to the fraction of total traffic they exchange with BitTorrent peers during the trace collection period. This means that the trackers exchanging the most messages are the most popular and the trackers exchanging the least messages are the least popular. Fig. 10 shows the measured fraction of total traffic versus the tracker rank at the four ISPs. The four curves are all very skewed, indicating that few very popular trackers account for the most of the traffic. For example, 10% of the most popular trackers are responsible for almost the totality of the traffic, e.g., 97% at ISP-A, ISP-C and 99% at ISP-D.

Further analysis of our traces shows that different trackers are popular across the different ISPs. If we focus on the top 10 trackers, accounting for more than 50% of the traffic, no tracker is popular at all four ISPs, and just two trackers are popular at three ISPs. ISP-C and ISP-D have the largest number of popular trackers in common (five trackers).

6. Conclusion

BitTorrent is currently the most popular Peer-to-Peer protocol and it has been extensively analyzed by the research community. This paper surveys the results presented in the literature and confirms many findings using a new, more comprehensive data-set.

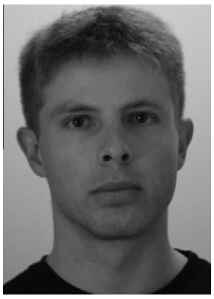
Our data-set consists of traffic exchanged between BitTorrent clients and the most popular trackers at ISP border routers. We set-up this data collection at four major ISPs in Europe during 2009 and 2010. This data collection is a very challenging task and we believe is a significant contribution of this paper. The second contribution of this paper is the extensive analysis of the traces collected which sheds some light on three major BitTorrent entities: *traffic*, *torrents*, *users* and *trackers*.

Our paper confirms that BitTorrent traffic is mostly unlocal, e.g, 95% of this traffic traverses expensive transit links between ISPs. However, only 1% of the torrents have potential for localization since very few torrents are popular within an ISP and users are highly desynchronized. Accordingly, a localization mechanism a la P4P can only partially return local peer-sets to requesting peers; on average, only 4–44% of the peers composing a peer-set could be local. We also highlight two potential problems in the BitTorrent ecosystem: (1) due to a scarcity of content pieces, 20% of user downloads can be inactive for more than 30 min and up to more than 10 h and (2) 5–10% of the users are free-riders, *i.e.*, they download content and never upload.

References

[1] Azureus/Vuze. <http://www.azureus.sourceforge.net/>.
[2] BitTorrent. <http://www.bittorrent.com/>.
[3] BitTorrent Specs. <http://wiki.theory.org/BitTorrentSpecification/>.
[4] S.L. Blond, A. Legout, W. Dabbous, Pushing BitTorrent locality to the limit, Elsevier Computer Networks 55 (3) (2011) 541–557.
[5] B. Cohen, Incentives Build Robustness in BitTorrent, Technical Report, 2003.
[6] R. Cuevas, N. Laoutaris, X. Yang, G. Siganos, P. Rodriguez, Deep diving into BitTorrent locality, in: INFOCOM, 2011.
[7] F. Dabek, B. Zhao, P. Druschel, I.S. Towards, A common api for structured peer-to-peer overlays, Lecture Notes in Computer Science 2735 (2003) 33–44.
[8] J.-H. Huang, C.-W. Chen, On performance measurements of TCP/IP and its device driver, in: LCN, Minneapolis, MN, USA, 1992.
[9] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A.A. Hamra, L. Garces-Erice, Dissecting BitTorrent: five months in a torrent's lifetime, in: PAM, Antibes Juan-les-Pins, France, April 2004.
[10] T. Karagiannis, P. Rodriguez, K. Papagiannaki, Should internet service providers fear peer-assisted content distribution? in: IMC, Berkeley, CA, USA, October 2005.
[11] A. Legout, G. Urvoy-Keller, P. Michiardi, Understanding BitTorrent: An Experimental Perspective, Technical Report INRIA-00000156, Inria, November 2005.
[12] A. Legout, G. Urvoy-Keller, P. Michiardi, Rarest First and Choke Algorithms are Enough, in: IMC, Rio de Janeiro, Brazil, October 2006.
[13] T. Locher, P. Moor, S. Schmid, R. Wattenhofer, Free riding in BitTorrent is cheap, in: HotNets, Irvine, CA, USA, November 2006.
[14] G. Maier, A. Feldmann, V. Paxson, M. Allman, On dominant characteristics of residential broadband internet traffic, in: IMC, Chicago, IL, USA, November 2009.
[15] Ookla's speedtest throughput measures.<https://confluence.slac.stanford.edu/display/IEPM/Ookla%27s+Speedtest+Throughput+Measures>.
[16] M. Piatek, T. Kohno, A. Krishnamurthy, Challenges and directions for monitoring p2p file sharing networks – or – why my printer received a dmca takedown notice, in: USENIX Security Symposium, 2008.
[17] D. Qiu, R. Srikant, Modeling and performance analysis of BitTorrent-like peer-to-peer networks, in: SIGCOMM, Portland, OR, USA, August 2004.
[18] M. Varvello, M. Steiner, Traffic localization for DHT-based BitTorrent networks, in: Networking, Valencia, Spain, May 2010.
[19] H. Xie, Y.R. Yang, A. Krishnamurthy, Y.G. Liu, A. Silberschatz, P4P: provider portal for applications, in: SIGCOMM, Seattle, WA, USA, August 2008.
[20] Yankee. <http://www.yankeegroup.com/>.
[21] YouTube. <http://www.youtube.com>.
[22] C. Zhang, P. Dhungel, D. Wu, K.W. Ross, Unraveling the BitTorrent Ecosystem, IEEE Transactions on Parallel and Distributed Systems 22 (7) (2011) 1164–1177.

Matteo Varvello received the Research Master's degree in network and distributed systems from EURECOM, Sophia Antipolis, France, in 2006, the M.Sc. (cum laude) degree in networking engineering from Polytechnic of Turin, Turin, Italy, in November 2006, and the Ph.D. degree in computer science from Telecom Paris, Paris France, in December 2009. He has been a Researcher with Alcatel-Lucent, Holmdel, NJ, since January 2010. His current research interests include peer-to-peer, content-centric networking, cloud computing, and energy efficiency.



Moritz Steiner received the M.S. degree (Diplom) in computer science from the University of Mannheim, Germany in 2005, and the Ph.D. degree in computer networks from Telecom ParisTech, France and the University of Mannheim in 2008. His doctoral thesis investigates how to build network virtual environments from unstructured peer-to-peer networks. It also introduced measurement techniques and presented extensive measurement results on a real-world, large-scale, structured peer-to-peer file sharing network, namely Kad. Currently he is a Member of Technical Staff at Bell-Labs located in Murray Hill, NJ. His research interests and project activities are in the areas of analysis and design of peer-to-peer networks and cloud computing.

Koen Laevens obtained a M.Sc. in Electrical Engineering and a Ph.D. in Applied Sciences from Ghent University (Belgium), in 1991 and 1999 respectively. During the years, he held several positions as researcher at Ghent University, Microsoft Research Cambridge, and Alcatel-Lucent Bell (formerly Alcatel Bell). His research focused on the broad area of performance evaluation and traffic engineering, all-optical network architectures and video delivery mechanisms in packet-switched networks. Currently, he is a co-founding project member inside the Bell Labs Ventures organization and still holds a part-time post-doc position at the Faculty of Engineering and Architecture of Ghent University.