DHT-based Traffic Localization in the Wild

Matteo Varvello, Moritz Steiner

Bell Labs, Alcatel-Lucent, USA
{*first.last*}@alcatel-lucent.com

***Abstract*—BitTorrent is both the dominant Peer-to-Peer (P2P) protocol for file-sharing and a nightmare for ISPs due to its network agnostic nature. Many solutions exist to localize BitTorrent traffic relying on cooperation between ISPs and the trackers. Recently, BitTorrent users have been abandoning the trackers in favor of Distributed Hash Tables (DHTs). Despite DHTs are complex heterogeneous systems, DHT-based traffic localization is also possible; however, it is unclear how it performs. The goal of this work is to measure DHT-based traffic localization in the wild. We run multiple experiments involving up to five commercial ISPs and a maximum duration of one month, collecting about 400 GB of BitTorrent traffic. Then, we perform an extensive analysis with the following goals: understand the impact of system parameters, verify accuracy of the measurements, estimate the localization benefits.**

I. Introduction

A. BitTorrent

BitTorrent is by far the most popular Peer-to-Peer (P2P) protocol, adopted by several file-sharing clients such as μTorrent, Transmission and BitComet. BitTorrent aims at maximizing the volume of data exchanged among peers without taking into account their geographic location. This causes expensive inter-ISPs traffic, and thus a monetary loss at the ISPs.

BitTorrent employs a *tracker* to discover peers and coordinate file exchanges. Peers retrieve the address of the tracker from a *torrent* file they download from the web. In the paper, we use the term torrent also to refer to a file or set of files that are exchanged. A peer contacts the tracker to retrieve a list of peers that participate to the *swarm*, group of peers that hold the file or a portion of it. The tracker answers with the *peer-list*, composed by 50 active peers.

With the objective of increased resiliency BitTorrent also uses distributed tracking, where a client discovers which peers hold a copy or a portion of a file querying a *Distributed Hash Table* (DHT). Each peer and torrent is assigned a unique identifier in the DHT, the *nodeID* and *infohash*; both are computed using hashing. Currently, BitTorrent use two large and incompatible DHTs called Azureus and Mainline.

Beside the tracker and the DHT, the *Peer-Exchange-Protocol* (PEX) is the third mechanism to discover peers that participate in a file exchange [9]. The PEX allows peers that download the same file to exchange their peer-sets pairwise.

B. Related Work

In the past, several interesting strategies have been proposed to localize BitTorrent traffic, *i.e.,* maintain traffic within an ISP when possible. Originally, researchers proposed to modify the peer selection at the client to favor local peers, as in [4], [7]. However, these designs only manage to select the few local peers, if any, from the small peer-set received. More effective designs, such as [2], [3], [10], propose to inform the trackers about the ISP of each peer. In this way, a tracker could reply to a peer request for a specific torrent with a list of peers located in the same ISP as the requesting peer.

Nowadays, BitTorrent DHTs are taking over the trackers as these are being shutdown by police due to copyright issues. For example, in [8] we measure that about 40% of the BitTorrent users from a large European ISP have already abandoned the trackers in favor of the DHTs. It follows that traffic localization mechanisms based on the central trackers might become soon ineffective. In [8], we also propose a technique to localize BitTorrent traffic that only relies on the DHT, with no need for the trackers. Compared to classic tracker-based localization, DHT-based localization is more challenging. In fact, DHTs are large P2P systems designed to fairly distribute responsibilities: it follows that tracking and controlling file swarms with the goal of localization is not trivial. In addition, DHTs are heterogeneous environments where several independent protocol implementations exist.

C. Contributions

The goal of this work is to measure DHT-based traffic localization in the wild. Our measurement methodology is as follows. We activate traffic localization for several ISPs and files by running the prototype for the Mainline DHT described in [8]. Meanwhile, we measure its performance by running several BitTorrent clients from up to five commercial ISPs for as long as four weeks. In total, we collect about 400 GB of BitTorrent traffic on which we perform two different analyses: (1) *sensitivity*, to both understand the impact of system parameters and verify the accuracy of our methodology, and (2) *localization benefits*, to quantify the volume of BitTorrent traffic that is kept local.

Our results show that while DHT-based traffic localization can keep local 90-100% of the traffic associated to a popular file, this fraction reduces to 60% for an unpopular one. In fact, as popularity increases the probability to find peers that concurrently request the same file from the same ISP increases as well. Also, at ISPs where the majority of the users have very good connectivity, the localization benefits are much higher than at slow ISPs. This happens for two reasons: (1) the BitTorrent protocol tends to favor fast peers, and (2) fast local peers reduce the chance that external peers contribute to a file download. In the experiments, we also discovered that BitComet has a non-standard DHT implementation that negatively detracts from the performance of DHT-based traffic localization.

II. Methodology

Our measurement methodology has two components: the *DHT side*, a DHT-based traffic localization, and the *client side*, a combination of BitTorrent clients and measurement tools to assess both DHT localization benefits and behavior. In the remainder of this section, we describe both components.

A. Client Side

Pcap - It collects statistics about a torrent download. The pcap tool works in three consecutive steps: (1) start of a BitTorrent client and collection of pcap traces using wireshark, (2) extraction of volume of traffic sent and received per peer in the swarm, and (3) post processing to derive the following statistics: volume of local traffic, download/upload speed per remote peer and number of peers contacted during a download. The pcap tool can be coupled with any legacy BitTorrent client.

Query - It also collects statistics about a file download. This is done by instrumenting a Transmission client to log for each peer it talks to the following statistics: upload/download rate, client type and client location (local or non-local). These statistics are dumped to a file every two seconds. This tool runs on Linux only as Transmission is a client for Linux.

Lookup analyzer - It analyzes a lookup operation in Mainline. First, it collects pcap traces filtered on the UDP port at which a BitTorrent client is running on. Then, it analyzes the pcap traces to extract the following statistics: (1) IP addresses of the peers that reply to a file's request, (2) time at which each reply message is received, and (3) set of "sources", both leechers and seeders, returned from each replying peer. The lookup analyzer can be coupled with any BitTorrent client implementation.

We run the tools from the following ISPs: *Comcast Cable* (US), *Verizon Internet Services* (US), *Free SAS* (France), *Telecom Italia* (Italy) and *Belgacom Skynet* (Belgium) (abbreviated Comcast, Verizon, Free, TItalia and Belgacom). At Comcast, we have access to a personal cable connection and both a Windows and Linux machine. At Verizon, we have access to a personal fiber connection and a Linux machine. At Free, TIalia and Belgacom, we have access to personal ADSL connections and Linux machines.

For our measurements, we consider a scenario where a user clicks on a "magnet link", a pointer to the infohash of a file that can be download from the Web; no tracker is thus involved. The rationale is to reproduce a scenario where either the trackers are unavailable or controlled by tracker-based localization, thus not detracting from the performance of the DHT-based localization mechanism. We also disable the PEX protocol in order to build a benchmark of the localization performance while providing high control on the experiments. Then, we briefly analyze the impact of PEX on the localization.

B. DHT Side

To the best of our knowledge, the solution proposed in [8] is the only available technique to localize BitTorrent traffic using DHTs. This DHT-based traffic localization mechanism works in two steps. First, it intercepts all messages from peers announcing in the DHT that they hold a file or a portion of it. Then, it intercepts all requests for this file and answer with local peer-sets. To intercept announces and requests for a file, the solution inserts several *sybils* [5] in the DHT: these are (logical) peers with nodeIDs close to the info_hash of the file to localize, that are controlled by a single (physical) peer. Sybils respond to the queries for this file with localized peer-sets. If only few local peers are available, external peers are used to complete the peer-set. This localization mechanism targets popular files only as they are the only ones that have potential for localization.

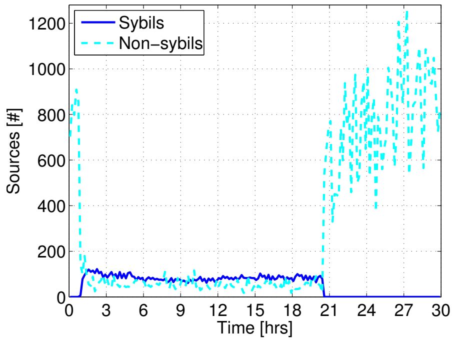


Fig. 1. Evolution over time of the sources received from non-sybil peers vs sources received from the sybils; μTorrent; Comcast; 30 hrs.

Unless otherwise stated, we run the "DHT side" (DHT-based traffic localization) from a data center in Chicago, USA. According to the evaluation goals, we activate the DHT side for a set of torrents with specific popularity at a given ISP. Since torrent popularity at ISP-level is not publicly available, we obtain the list of the 500 (globally) most popular torrents as indicated by PirateBay on May 5th, 2012. Then, we activate the localization for 100 torrents, randomly selected out of the 500 most popular torrents, along 24 hrs in order to quantify their popularity at ISP level, *i.e.,* number of requests from peers located at the same ISP.

Unless otherwise stated, for the experiments in Comcast we select the torrents whose average number of sources is respectively the 100, 80, 60 and 20th percentile of the torrent popularity distribution; we refer to these torrents as 1, 2, 3 and 4, respectively. For the experiments that we run in the remaining ISPs, we only select the most popular torrent. The torrents we select for Verizon, Free and TItalia have a very high average number of sources, ranging from 120 at Free to 240 at Verizon, whereas the most popular torrent at Belgacom only has about 35 local sources.

In the upcoming sections, we extensively analyze the performance of a DHT-based traffic localization. We start by a *sensitivity* analysis and then we dive into a quantification of the *localization benefits*.

III. Sensitivity Analysis

In this section, we study the sensitivity of the DHT localization mechanism from both the DHT and the client side. Our goal is to both understand the impact of system parameters and verify the accuracy of our measurement methodology.

A. DHT Side

The number of sybils per file, k, is the only system parameter on the DHT side. Since k increases the cost of the localization mechanism (memory, CPU and bandwidth), here we attempt to find the smallest value of k which does not impact the localization benefits. To do so, we localize torrent 1, 2, 3 and 4 along seven days using a different number of sybils k on each day: 1, 2, 4, 8, 16, 32 and 64 sybils, respectively. Meanwhile, we log the number of worldwide sources per torrent every 15 minutes and observe how they evolve over time. Although not shown due to space limitations, we observe that for all torrents the median number of sources remains constant as the number of sybils k equals 64 and 32: 2,800 and 100 sources for torrent 1 and 4, respectively. As k decreases the number of available sources decreases as well, reaching about 5% of the initial number as k equals 1. We thus conclude that when $k < 32$, peers in the DHT are not always able to lookup the sybils and thus the sybils cannot intercept their announces and requests for a file. It follows that other peers than the sybils receive these messages detracting from the performance of the localization mechanism. For this reason, in the following experiments we set $k = 32$.

In P2P networks, torrents can become suddenly very popular or unpopular. It is thus important that the DHT side quickly reacts to popularity changes. We measure the responsiveness of the DHT side as follows. On the client side, we couple the lookup analyzer and the pcap tool with μTorrent so to download torrent 1 every 5 minutes for 30 hrs; we run the tools and μTorrent on a Windows machine in Comcast. On the DHT side, we activate the localization for torrent 1 and Comcast after one hour from the beginning of the experiment, and we let it run for 19 hrs. Figure 1 shows the evolution over time of the number of sources for torrent 1 received by the client every five minutes. We divide the sources as the ones received from the sybils (blue solid line) and the ones received from non-sybil peers (light blue dashed line). During the first hour, the client receives between 700 and 900 sources from non-sybils peers; the number of sources received from the sybil is equal to zero as traffic localization is deactivated. At t=1hr, the sybils start serving local sources for torrent 1: in about 15 minutes, the number of sources received from the sybils grows to about 100-120 sources, a value which stays constant in the following 19 hrs. Meanwhile, the number of sources received from non-sybil peers rapidly decreases from 900 to a value comprised between 20 and 120 sources. Note that 15 minutes is the maximum time interval between two client's announce messages for a torrent, as indicated in the Mainline specifications [1]. At t=20hrs, we deactivate the traffic localization for torrent 1: as the sybils are unreachable, the number of sources received from the sybils drops to zero. Instead, the number of sources received from non-sybils grows from 40 to 600 in 5 minutes, and it goes back to about 800 sources in 15 minutes. In the following 10hrs, the number of sources received from non-sybils oscillates between 400 and 1,200, whereas the number of sources received from the sybils is always equal to zero.

This experiment shows that the localization mechanism is very responsive. In fact, it only requires 15 minutes to localize traffic for a torrent, which we believe it is quick enough to follow torrent popularity changes. This experiment also shows that the localization mechanism does not take full control of a swarm. In the paper, we largely study this phenomenon and explain its causes. Ironically, while this detracts from the volume of traffic that can be localized, it ensures overlay resilience, *i.e.,* that a client's download is never interrupted due to absence of local peers.

B. Client Side

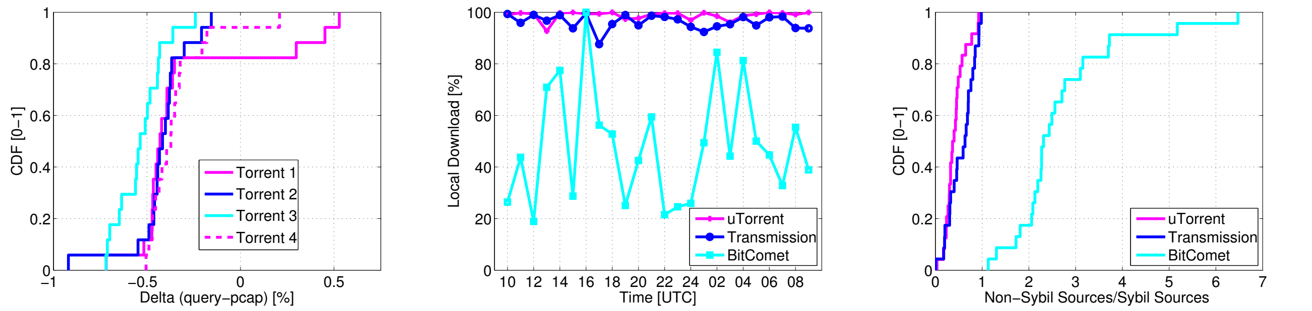
To start, we verify that both pcap and query tool accurately measure the volume of traffic that stays local. On the DHT side, we localize torrents 1, 2, 3, 4 for Comcast over one day. On the client side, we instrument both the pcap and query tool to download each torrent for 10 minutes every hour during one day. We ran this experiment at the machine located at Comcast between June 1st and 2nd, 2012. This experiment generated about 100 GBytes of P2P traffic.

Figure 2(a) shows the Cumulative Distribution Function (CDF) of the difference in local download traffic as measured with the query and the pcap tool. Each distribution is centered around 0, *i.e.,* the two tools report the same measures most of the time. The distributions stretch from a minimum of -1% to a maximum of +0.5%; this indicates that the values reported by each tool have a discrepancy of less than $\pm 2\%$. The pcap tool seems to underestimate the volume of local traffic; this happens because some interfering traffic (which was hard to filter) can be present at the moment of the pcap capture. For this reason, in the following experiments we prefer to use the query tool when possible.

Finally, we study whether the usage of a specific BitTorrent client impacts the accuracy of the measurements. On the DHT side, we localize torrent 1 for Comcast. On the client side, we couple the pcap tool with the three most popular BitTorrent clients, namely μTorrent, Transmission and BitComet, and run them at the Windows and Linux machines at Comcast. We use two virtual machines to concurrently run μTorrent and BitComet. We instrument the pcap tool to download torrent 1 every hour for 10 minutes during one day. In addition, we run an instance of the lookup analyzer together with each client. We ran this experiment at the machines located at Comcast between June 5th and 6th, 2012. This experiment generated about 80 GBytes of P2P traffic.

Figure 2(b) shows the evolution over time of the fraction of traffic that stays local as measured for each client. The traffic localization does not reach 100% all the time. On average, 95-97% of the traffic stays local with Transmission and μTorrent, whereas only 50% with BitComet. We expected to measure 100% of localization at each client as the client-to-tracker communication and the PEX protocol are disabled: each client should receive only peer-sets from the sybils and thus only interact with local clients. Also, we measure a different fraction of local traffic with each client. While the small difference between μTorrent and Transmission is most likely due to some in-deterministic client behaviors, it is clear that *DHT traffic localization is less effective for BitComet.*

To further understand this result, we analyze the data collected by the lookup analyzer. We find that additional



(a) Difference in local download traffic as measured with query and pcap tool; Transmission; torrent=[1,2,3,4]; seven days.

(b) Local Download Traffic; μTorrent, Transmission, BitComet; torrent 1; one day.

(c) CDF of sources received from non-sybil peers vs sources received from the sybils; μTorrent, Transmission, BitComet; one day.

Fig. 2. Sensitivity Analysis; Client Side; Comcast.

peers beside the sybils replied to file requests sent by the clients. Thus, the three clients receive additional sources other than the local sources provided by the sybils. This implies that non-sybil peers in the DHT have previously received announce messages for this torrent. We inspect the announce traffic collected at each client. Both μTorrent and Transmission effectively locate the sybils and announce to 3 and 8 sybils, respectively, whereas BitComet does not correctly announce to the sybils. This probably happens because BitComet does not properly implement the DHT announcement mechanism: it fails to lookup the closest peers to an infohash thus sending announce messages to other peers than the sybils. In [6], the authors observe the same behavior for BitSpirit and KTorrent, two less popular clients. *The fact that few BitTorrent client implementations might not announce to the sybils negatively impacts the DHT-based traffic localization also for clients which (correctly) announce to the sybils.*

However, this does not justify why the BitComet client sees much less traffic localization than μTorrent and Transmission. To explain this result, we compute the ratio of sources received by non-sybil peers (mostly non-local) and local sources received by the sybils. Figure 2(c) shows the CDF of this ratio computed for each client. While Transmission and μTorrent have a similar behavior, e.g., in 100% of the experiments the client receives more sources from the sybils than from non-sybil peers, BitComet departs from it. In 80% of the experiments, the BitComet client receives a number of sources from non-sybil peers which is at least twice as high as the number of sources received from the sybils; this explains the lower traffic localization. Intrigued by this behavior, we visually inspected the sources received by the BitComet client. We observe that the majority of the non-local sources (received from non-sybil peers) are BitComet clients as well. *This suggests the presence of a parallel P2P network formed by BitComet clients only: we suspect that BitComet tends to favor BitComet peers in its routing tables.*

IV. Localization Benefits

This section aims to answer a fundamental question: How effective is DHT-based traffic localization? We start with the analysis of a four-weeks experiment conducted in Comcast. Then, we continue with an analysis of a one-week experiment that involves five different ISPs.

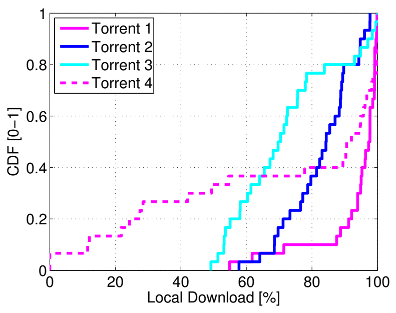
A. One Month, Single ISP

We ran the experiments along 28 days, from June 1st to June 28th, 2012. On the DHT side, we localize torrents 1, 2, 3, 4 for Comcast. On the client side, we instrument the query tool to download each torrent for ten minutes every day between 3 and 4 PM (EST). This experiment generated about 60 GBytes of P2P traffic.

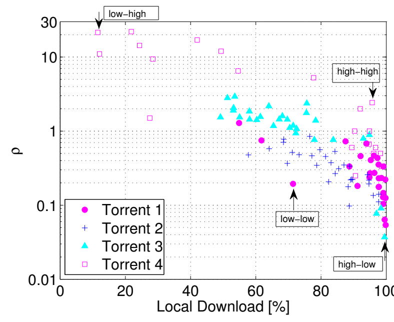
Figure 3(a) shows the CDF of the fraction of traffic that stays local per torrent. With the exception of torrent 4, as the torrent popularity decreases, the fraction of traffic that stays local decreases. For example, during 80% of the time more than 90% of the traffic stays local for torrent 1, whereas such a high traffic localization is achieved only in 20% of the measurements for torrent 3. For torrent 4, the localization benefits are either very low, e.g., between 10 and 50%, or very high, e.g., between 90 and 100%. Similarly, for 20% of the measurements torrent 3 achieves higher localization benefits than torrent 2, despite being systematically less popular. This happens because the number of locally available sources is not the only parameter that plays a role in the localization benefits. It is also very important to take into account the number of sources retrieved from non-sybil peers as well as the download speed of both local and non-local peers.

To better understand the latter result, we dissect the swarm of each download repetition. First, we derive the set of sources the client connects to and divide it into: (1) *non-local* sources received from non-sybil peers, and (2) *local* sources received from the sybils. Then, we compute the *non-local-to-local ratio*, ρ, defined as the ratio of the number of non-local sources versus the number of local ones.

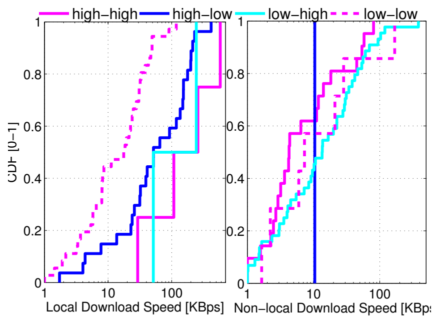
Figure 3(b) shows a scatter-plot of the fraction of local traffic (x-axis) and ρ (y-axis) per torrent over four weeks. Globally, as ρ increases, the fraction of traffic which stays local decreases as well. For example, when this ratio assumes values larger than one, *i.e.,* a client connects to more non-local than local sources, the fraction of local traffic is mostly lower than 70%. Similarly, when this ratio is very low, e.g., lower than 0.1, about 100% of the traffic stays local. However, in few cases the volume of local traffic can be very high also in presence of high ρ and vice-versa. For example, if we focus on torrent 4 in the upper right section of the Figure we see that 95% of its traffic stays local despite $\rho = 2$, *i.e.,* there are twice



(a) CDF of Local Download Traffic.



(b) Scatter-plot of local download and "non-local-to-local" ratio (ρ).



(c) CDF of local (left) and non-local (right) download speed; high-high, high-low, low-high, low-low.

Fig. 3. Localization benefits analysis; Comcast; torrent=[1,2,3,4]; four weeks.

more non-local than local sources, which is counter-intuitive.

To explain the latest observation, we analyze the download speed of local and non-local sources in four representative download repetitions (each indicated by an arrow in Figure 3(b)). We label these repetitions as follows.

High-high, high fraction of local download and high ρ. We consider a download of torrent 4 for which we measure 95% of localization and $\rho = 2$. *High-low*, high fraction of local download and low ρ. We consider a download of torrent 3 for which we measure 99% of localization and $\rho = 0.03$. *Low-high*, low fraction of local download and high ρ. We consider a download of torrent 4 for which we measure 10% of localization and $\rho = 20$. *Low-low*, low fraction of local download and low ρ. We consider a download of torrent 1 for which we measure 70% of localization and $\rho = 0.2$.

We derive the download speed for both local and non-local sources by dividing the amount of bytes downloaded from a source by the time this source sends data. Figure 3(c) shows the CDFs of the download speed for both local and non-local sources as computed for the high-high, high-low, low-high and low-low swarms. We make the following observations.

High-high – Non-local sources are mostly slow, *i.e.,* their download speed is always smaller than 100 KBps. Conversely, the few local clients are very fast: 50% of them allow the client to download at more than 100 KBps and up to 600 KBps. This explains why despite there are twice more non-local sources than local ones ($\rho = 2$), 99% of the traffic stays local.

High-low – There are 27 local sources whose download speeds vary in the range 2-400 KBps. Only a single non-local source is present and it only offers a download speed of 10 KBps. This explains why 99% of the traffic stays local.

Low-high – There are only two local sources with download speed of 50 and 250 KBps, respectively. The non-local peers are slower than the local ones, e.g., 80% have a download speed of less than 50 KBps, but one peer is very fast with about 400 KBps. The combination of high number of non-local sources and some high download speeds from non-local sources explain why only 10% of the traffic stays local.

Low-low – There are 36 local sources with a download speed in the range 1-120 KBps. Only seven non-local clients are present, but one of them is very fast with a download speed of about 170 KBps, thus contributing a lot to the non-local download traffic. This explains why only 70% of the traffic stays local despite $\rho = 0.2$.

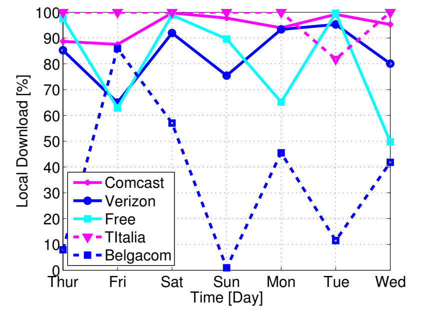
To summarize, *the volume of traffic which stays local is proportional to the fraction of local sources that DHT-based traffic localization can provide. In few cases, the download speed that each single peer can provide also plays a key role.*

B. One Week, Multiple ISPs

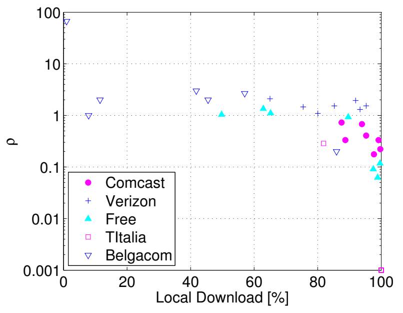
We are interested in measuring how the localization mechanism performs at multiple ISPs. On the DHT side, we localize the most popular torrents at Comcast, Verizon, Free, TItalia and Belgacom. On the client side, we instrument the query tool to download each torrent for ten minutes every day between 3 and 4 PM (machine local time). We ran this experiment on the machines we control at each of the above mentioned ISPs during seven days, from June 16th to June 22, 2012. This experiment generated about 16 GBytes of P2P traffic.

Figure 4(a) shows the evolution over time of the fraction of traffic that stays local at each ISP. Traffic localization works at best in TItalia, where 100% of the traffic stays local for 6 over 7 days, and at worst in Belgacom, where the fraction of local traffic is lower than 60% for 6 out of 7 days. As observed in Figure 3(a) for a very popular torrent, more than 90% of the traffic stays local at Comcast most of the time. At Verizon and Free, between 50 and 99% of the traffic stays local.

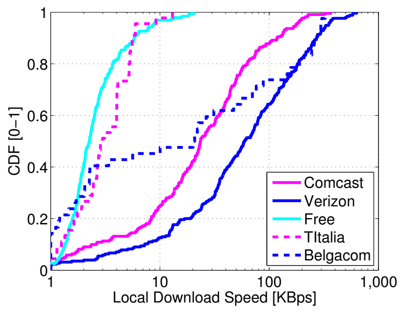
To further understand the localization benefits at each ISP, Figure 4(b) shows a scatter-plot of ρ, the "non-local-to-local ratio", and the fraction of local download per ISP. ρ is lowest at Free and highest at Belgacom: as already observed in Figure 3(b), this trend follows the torrent popularity, e.g., on average there are 100 and 10 local sources at Free and Belgacom, respectively. TItalia departs from this trend: despite there are on average fewer local sources than at Free, on average 30 versus 100 local sources, ρ equals zero in 6 out of 7 repetitions. Finally, Figure 4(b) shows that for a similar value of ρ the fraction of local traffic largely differs at different ISPs. For example, when ρ assumes values between 1 and 2, the local download traffic at Belgacom is always smaller than 60% whereas it stays between 60 and 90% at Verizon. Similarly, in 4 out of 7 measurements at Verizon and Comcast the traffic localization stays between 85 and 90% despite ρ values of $\approx$ 3 and $\approx$ 0.5, respectively. Thus, it appears that *the ISP one localize traffic for also plays an important role in the performance of DHT-based traffic localization.*



(a) Evolution over time of the local download traffic.



(b) Scatter-plot of local download and "non-local-to-local" ratio (ρ).



(c) CDF of local download speed.

Fig. 4. Multi-ISP analysis; Comcast, Verizon, Free, TItalia, Belgacom; one week.

To explain the latest observation, we plot the CDFs of the local download speed measured at each ISP (Figure 4(c)). This is computed by deriving the average download speed per local source and per day. Figure 4(c) shows that Verizon local peers are the fastest, with a median average download speed of 60 KBps and a maximum average speed of 700 KBps. This explains why in Figure 4(b), given the same value of ρ, more traffic tends to stay local at Verizon. TItalia and Free are the two slowest ISPs, with local download speed mostly in the order or few KBps. As above, this explains why despite Verizon and Free have comparable ρ, we measure higher traffic localization at Verizon. Finally, a very particular behavior is observable at Belgacom: 50% of the times the local download speed is lower than 10 KBps, whereas the rest of the time it rapidly grows up to hundreds of KBps. However due to a high ρ, the fraction of local traffic at Belgacom is the lowest.

C. *One Day, Single ISP*

We analyze the impact of the PEX protocol on the localization benefits. On the DHT side, we localize a torrent for Verizon with about 70 local sources. On the client side, we instrument the query tool to download the torrent twice every hour: for the first download, we deactivate the PEX whereas we activate it for the second one. We ran this experiment at our machine located at Verizon between July 1st and 2nd, 2012. This experiment generated about 24 GBytes of P2P traffic.

Figure 5 shows the evolution over time of the fraction of traffic that stays local when the PEX is disabled and enabled. *The PEX negatively impacts the DHT-based traffic localization*; however, the average traffic localization achieved decreases only by 10%, e.g., from 88% measured when PEX is disabled to 78% when the PEX is enabled.

V. Conclusion

BitTorrent, today's most popular Peer-to-Peer protocol for file-sharing, causes expensive inter-ISPs traffic due to its network agnostic nature. To reduce such traffic, in the past one had to control BitTorrent trackers. However, recently BitTorrent users have been abandoning the trackers in favor of Distributed Hash Tables (DHTs); it follows that traffic localization needs to rely on the DHT. DHT-based traffic localization is challenging as DHTs are complex and heterogeneous systems. In this paper, we assess the performance of DHT-based traffic localization for BitTorrent via a large scale measurement. We run multiple experiments involving up to five commercial ISPs and a maximum duration of one month, collecting overall 400 GB of BitTorrent traffic. We find that the volume of traffic which stays local is proportional to the fraction of local sources that DHT-based traffic localization can provide. Also, the localization benefits tend to be higher at fast ISPs, *i.e.,* ISPs where subscribers have very good connectivity. Finally, we find that BitTorrent clients have different implementations of the DHT protocol; some implementations, such as the BitComet one, detract from the overall localization benefits.

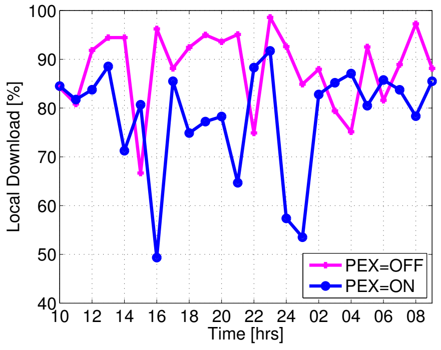


Fig. 5. Evolution over time of the local download traffic; Verizon; PEX=[ON,OFF]; one day.

References

[1] BitTorrent Specification. wiki.theory.org/BitTorrentSpecification.

[2] V. Aggarwal, A. Feldmann, and C. Scheideler. Can ISPs and P2P users cooperate for improved performance? *CCR*, 37(3):29–40, 2007.

[3] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang. Improving traffic locality in bittorrent via biased neighbor selection. In *ICDCS*, Lisbona, Portugal, July 2006.

[4] D. R. Choffnes and F. E. Bustamante. Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems. In *SIGCOMM*, Seattle, USA, August 2008.

[5] J. R. Douceur. The Sybil Attack. In *IPTPS'02*, Cambridge, MA, USA.

[6] I. S. Garcia. Exploring Mainline DHT: an experimental approach. Master thesis, KTH Royal Institute of Technology, November 2010.

[7] J. Ledlie, P. Gardner, and M. Seltzer. Network coordinates in the wild. In *NSDI*, Cambridge, MA, USA, April 2007.

[8] M. Varvello and M. Steiner. Traffic Localization for DHT-based BitTorrent networks. In *Networking'11*, Valencia, Spain, May 2011.

[9] D. Wu, P. Dhungel, X. Hei, C. Zhang, and K. W. Ross. Understanding peer exchange in bittorrent systems. In *P2P'10*, Delft, NE, 2010.

[10] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz. P4P: provider portal for applications. In *SIGCOMM'08*, Seattle,USA.