BitTorrent Locality and Transit Traffic Reduction: *when, why and at what cost?*

Ruben Cuevas*, Nikolaos Laoutaris†, Xiaoyuan Yang†, Georgos Siganos† and Pablo Rodriguez†
*University Carlos III Madrid, Spain †Telefonica Research, Spain

***Abstract*—A substantial amount of work has recently gone into localizing BitTorrent traffic within an ISP in order to avoid excessive and often times unnecessary transit costs. Several architectures and systems have been proposed and the initial results from specific ISPs and a few torrents have been encouraging. In this work we attempt to deepen and scale our understanding of locality and its potential. Looking at specific ISPs, we consider tens of thousands of concurrent torrents, and thus capture ISP-wide implications that cannot be appreciated by looking at only a handful of torrents. Secondly, we go beyond individual case studies and present results for few thousands ISPs represented in our dataset of up to 40K torrents involving more than 3.9M concurrent peers and more than 20M in the course of a day spread in 11K ASes. Finally, we develop scalable methodologies that allow us to process this huge dataset and derive accurate traffic matrices of torrents. Using the previous methods we obtain the following main findings: (i) Although there are a large number of very small ISPs without enough resources for localizing traffic, by analyzing the 100 largest ISPs we show that Locality policies are expected to significantly reduce the transit traffic with respect to the default random overlay construction method in these ISPs; (ii) contrary to the popular belief, increasing the access speed of the clients of an ISP does not necessarily help to localize more traffic; (iii) by studying several real ISPs, we have shown that *soft* speed-aware locality policies guarantee win-win situations for ISPs and end users. Furthermore, the maximum transit traffic savings that an ISP can achieve without limiting the number of inter-ISP overlay links is bounded by "unlocalizable" torrents with few local clients. The application of restrictions in the number of inter-ISP links leads to a higher transit traffic reduction but the QoS of clients downloading "unlocalizable" torrents would be severely harmed.**

I. INTRODUCTION

Most design choices in P2P applications are dictated by end user performance and implementation simplicity. Bootstrapping is one such example: a new node joins a P2P overlay by connecting to a *Random* set of neighbors. This simple process provides fault tolerance and load balancing to end users and implementation simplicity to developers. Its downside, however, is that it is completely oblivious to the requirements and operating constraints of ISPs and thus it often leads to serious problems such as increasing the transit costs, worsening the congestion of unpaid peering links [23], and expediting the upgrade of DSLAMs. Therefore, several ISPs have allegedly started rate limiting or blocking P2P traffic [12]. In response, P2P applications have tried to conceal and evade discriminatory treatment by using dynamic ports and protocol encryption.

Much of this tension can be avoided by biasing the overlay construction of P2P towards *Locality*. It is known that geographic proximity often correlates with overlap of consumption patterns [19] and thus bootstrapping P2P users with other nearby ones can confine P2P traffic within ISPs instead of letting it spill to other domains over expensive transit links. This simple idea has received much attention lately because it is generic and thus can be applied to a variety of P2P applications independently of their internal logic (scheduling, routing, *etc.*). Systems like P4P [34] and ONO [7] have been proposed for localizing the traffic of the BitTorrent file sharing protocol [8].

Despite the interesting architectures and systems that have been proposed, we believe that we still stand on preliminary ground in terms of our understanding of this technology. The main ideas are straightforward, but their implications can be quite the opposite, for several reasons. First, different torrents can have quite diverse *demographics*: a blockbuster movie has peers around the world and thus can create much more transit traffic than a local TV show whose peers are mostly within the same country/ISP, especially if language gets in the way. Predicting the ISP-wide transit traffic due to P2P amounts to understanding the demographics of thousands of different torrents downloaded in parallel by all the customers. Things become even more complicated in the case of the BitTorrent protocol whose free-riding avoidance scheme makes peers exchange traffic predominately with other peers of similar speed [21]. Thus even if two ISPs have similar demographic composition, the fact that they offer different *access speeds* can have a quite pronounced impact on the amount of transit traffic that they see. The combined effect of demographics and access speeds makes it risky to generalize observations derived from a particular ISP and few individual torrents.

II. OUR CONTRIBUTIONS

Our work provides detailed case studies under representative ISP-wide workloads as well as holistic views across multiple (thousands of) ISPs. In all cases we demand that the input be as representative as possible (demographics and speed of different ISPs) and the methodology be scalable without sacrificing essential BitTorrent mechanisms like the unchoke algorithm, the rarest (*i.e.*, least replicated) first chunk selection policy, and the effect of seeders. We collected representative input data by scraping up to 100K torrents of which at least 40K had active clients from Mininova and Piratebay, the two most popular torrent hosting sites in the world according to the Alexa Ranking at the moment of our measurement study. We then queried the involved trackers and leverage the Peer Exchange (a gossiping protocol) to construct a map of BitTorrent demand demographics of up to 3.9M concurrent users and more than 21M total users over the course of a day, spread over 11K ISPs. For all those ISPs we obtained speeds from a commercial speed-test service [2] and from the iPlane project [25].

Our datasets are too big to conduct emulation or simulation studies. To process them we employ three scalable methodologies: (i) we use a probabilistic methodology for deriving speed-agnostic upper and lower bounds on the number of piece exchanges (*i.e.*, unchoke slots) that can be localized within

The research leading to these results has been partially funded by the European Union's FP7 Program under the projects eCOUSIN (318398) and TREND (257740), the Spanish Ministry of Economy and Competitiveness under the eeCONTENT project (TEC2011-29688-C02-02), and the Regional Government of Madrid under the MEDIANET Project (S2009/TIC-1468).

an ISP given its demand demographics. This technique allows us to scale our evaluation up to as many ISPs as we like; (ii) we evaluate the effect of access speed on the ability of a Random overlay construction policy to keep unchokes local within an ISP. For this purpose we define a metric named *Inherent Localizability*; (iii) finally, we define a more accurate deterministic methodology that estimates the resulting traffic matrix for a BitTorrent swarm taking into consideration the speeds of clients from the involved ISPs. This technique allows us to zoom in into particular ISPs and refine our estimation of transit traffic and end-user QoS.

In our analysis, we study the performance of several overlay construction mechanisms that include: *Random*, the default BitTorrent overlay in which a node selects its neighbors at random; *Locality Only If Faster, (LOIF)*, an end-user QoS preserving overlay that switches remote neighbors for locals only when the latter are faster; *Locality*, a simple policy that maximizes transit savings by switching as many remote neighbors as possible with local ones, independently of relative speed; *Strict*, is a strict version of Locality in which a node, after performing all the possible switches, just keeps one remote connection.

Summary of results: We shed light on several yet unanswered questions about BitTorrent traffic. Specifically:

(1) We use the demand demographics of the 100 largest ISPs from our dataset to derive speed agnostic upper and lower bounds on the number of chunk exchanges that can be kept local. In half of the ISPs, Locality keeps at least 42% and up to 72% of chunks internal, whereas Random can go from less than 1% up to 10%.

Next we focus on the three largest US and the three largest European ISPs in our dataset and derive their traffic matrices using both demographic and speed information. These detailed case studies reveal the following:

(2) LOIF preserves the QoS of users and reduces the transit traffic of fast ISPs by around 30% compared to Random. In slower ISPs the savings are around 10%.

(3) Locality achieves transit traffic reductions that peak at around 55% for the ISPs that we considered. The resulting penalty on user download rates is typically less than 6%.

(4) The barrier on transit traffic reduction is set by "unlocalizable" torrents, *i.e.*, torrents with one or very few nodes inside an ISP. In large ISPs, such torrents account for around 90% of transit traffic under Locality and are requested by few users of the ISP ($\sim$10%). In a sense, for these ISPs, the majority of users is subsidizing the transit costs incurred by the few users with a taste for unlocalizable torrents. Smaller ISPs host a larger number of clients in unlocalizable torrents.

(5) By limiting the number of allowed inter-ISP overlay links per client huge reductions of transit (>95%) are possible. The resulting median penalty is around 20% but users on "unlocalizable" torrents incur huge reduction of QoS (99%).

(6) Finally, we show that, contrary to popular belief, increasing the speed of access connections does not necessarily keep more traffic local as it might bring an ISP within unchoke distance from other fast ISPs which previously did not send it traffic.

Overall our results show that there is great potential from locality for both ISPs and users. There has been quite some speculation about this, so our effort was to substitute the speculation with concrete numbers about the benefits of this technology and thus assist ISPs with procurement decisions and implementors with further development of it.

A previous version of this paper was published at IEEE Infocom 2011 [10]. In this paper we extend our analysis to a larger number of ISPs and datasets. Moreover, we carefully analyze the impact that speed upgrades may have for an ISP transit traffic. Finally, we provide details and an accurate validation for the proposed method to estimate the traffic matrix of BitTorrent swarms.

The remainder of the article is structured as follows. In Sect. III we derive upper and lower bounds on the number of localized unchokes under Random and Locality overlays, independently of ISP speed distributions. In Sect. IV we present our measurement study of BitTorrent demographics. We also define a metric for explaining the performance of Random when factoring in real speed distributions across ISPs. In Sect. V we present a methodology for estimating BitTorrent traffic matrices and in Sect. VI we define the overlay construction policies that we use later in our study. Sect. VII characterizes the win-win situations and the tradeoffs between ISPs and users under different locality policies. In Sect. VIII we present a validation prototype for studying locality using live torrents and factoring in network bottlenecks. In Sect. IX we look at related work and we conclude in Sect. X.

III. WHY NOT A RANDOM OVERLAY?

Our goal in this section is to understand the cases in which a Random selection of neighbors localizes traffic well, and the ones in which it fails thereby creating the need for locality-biased neighbor selection. To do so we first need to understand the *stratification effect* [21] arising due to the unchoke algorithm [8] used by BitTorrent to combat free-riding. According to this algorithm, a node monitors the download rates from other peers and "unchokes" the k peers (typically 4–5) that have provided the highest rates over the previous 20 sec interval. These peers are allowed to fetch missing chunks from the local node over the next 10 sec interval. Therefore, as long as there are chunks to be exchanged between neighbors (*Local Rarest First* chunk selection works towards that [8]), peers tend to stratify and communicate predominantly with other peers of similar speed. In this section, we employ probabilistic techniques to carefully analyze the consequences of stratification on inter-domain traffic. We focus on a single ISP A and torrent T and analyze the conditions under which Random localizes sufficiently within A the traffic due to T.

A. Sparse mode – the easy case for Random

Let $V(T)$ denote the set of BitTorrent nodes participating in T, and $V(A,T) \subseteq V(T)$ the subset that belongs to ISP A. We say that ISP A is on *sparse mode* with respect to torrent T if the nodes outside A that participate in T have very dissimilar speeds with nodes that are within A. In this case, because of stratification, local nodes of A will talk exclusively to each other irrespectively of other remote nodes in their neighborhood. Then to confine all unchokes within A, each local node needs to know at least k other local neighbors. If W denotes the size of a neighborhood (40 upon bootstrap and growing later with incoming connections), then for Random to localize all traffic it has to be that a random draw of W out of the total $|V(T)|-1$ (-1 to exclude the node that is selecting) nodes yields at least k local ones. The probability of getting x "successes" (*i.e.*, local nodes) when drawing randomly W samples from a pool of $|V(T)|-1$ items, out of which

$|V(A,T)|-1$ are "successes", is given by the Hyper-Geometric distribution $\mathrm{HyperGeo}(x, |V(T)|-1, |V(A,T)|-1, W)$ [14]. Thus the expected number of localized unchokes is

$$\sum_{x=0}^{\min(|V(A,T)|-1,W)} \min(x,k) \cdot \mathrm{HyperGeo}(x, |V(T)|-1, |V(A,T)|-1, W) \tag{1}$$

Taking the mean value of the distribution we can write a condition for Random to localize well in sparse mode:

$$\frac{W \cdot (|V(A,T)|-1)}{|V(T)|-1} \geq k \tag{2}$$

B. Dense mode – things getting harder

ISP A is on *dense mode* with respect to T if the remote nodes participating in T have similar speeds to the nodes of A. In this case stratification does not automatically localize traffic inside A. From the standpoint of the unchoke algorithm, both local and remote nodes look equally good and thus the number of localized unchokes depends on their ratio in the neighborhood. Thus, although in sparse mode a random draw yielding $x \leq k$ local nodes would keep all x unchokes local, in dense mode it keeps only $k \cdot x/W$ of them local in expectation. To get the expected number of localized unchokes in dense mode we have to substitute $\min(x,k)$ with $k \cdot x/W$ in Eq. (1).

C. The promise of Locality

Let us now consider Locality, an omniscient overlay construction mechanism that knows all local nodes and thus constructs highly localized neighborhoods by providing each node with as many local neighbors as possible, padding with additional remote ones only if the locals are less than W. Then in sparse mode Locality localizes all unchokes as long as $|V(A,T)|-1 \geq k$, which is a much easier condition to satisfy than the one of Eq. (2), else it localizes only $|V(A,T)|-1$. In dense mode Locality localizes all unchokes as long as $|V(A,T)|-1 \geq W$.

IV. DEMOGRAPHICS OF BITTORRENT

We conducted a large measurement study of BitTorrent demand demographics. We begin with a presentation of our measurement methodology and then use the obtained demographics to derive upper and lower bounds on the number of localized regular unchokes under Random and Locality. At the end of the section we incorporate the effect of speed differences among ISPs and show that it is non trivial to predict what happens to the transit traffic of an ISP when it upgrades the speed of its residential accesses.

A. Measurement methodology

We use the BitTorrent crawler developed in [31] that obtains a snapshot of the IP addresses of all the clients participating in a set of torrents that are provided as input[1] . In Table I we present the different sets of torrents used in our study. Our crawler first scrapes a torrent indexing site to obtain `.torrent` meta information files. From them it obtains the addresses of the corresponding trackers. It queries repeatedly the trackers and uses Peer Exchange (the gossip protocol implemented in the latest versions of BitTorrent) to obtain all the IP addresses of clients participating in each torrent. The gathered IP addresses are mapped to ISPs and countries using the MaxMind database [1]. The crawler also obtains the number of seeders and leechers in each torrent. Crawling an individual torrent takes less than 2 minutes. Thus we get a pretty accurate "snapshot" of each torrent, *i.e.*, we are sure that the obtained IPs are indeed present at the same time. The time difference between the first and last crawled torrent was up to 90 minutes for the largest dataset (`mn40K`). However, we tracked individual torrent populations and found them to be quite stable across a few hours. Thus our dataset is similar to what we would get if we used a very large number of machines to crawl more torrents in parallel.

Set name	Source	Torrents	# IPs	# ISPs
`mn40K`	Mininova	latest 40K	3.9M	10.4K
`mn3K`	Mininova	latest 3K	17.4M	10.5K
`pb600`	Piratebay	600 most popular	21.9M	11.1K

TABLE I

TORRENT SETS COLLECTED IN THE PERIOD AUG-OCT 2009. FOR `mn40K` WE COLLECTED THREE VERSIONS, WITH ONE WEEK IN BETWEEN THEM. FOR `mn3K` AND `pb600` WE REPEATED THE CRAWL EVERY HOUR FOR ONE DAY. THE #IPS AND #ISPS FOR `mn40K` ARE PER SNAPSHOT, WHEREAS FOR `mn3K` AND `pb600` ARE DAILY TOTALS.

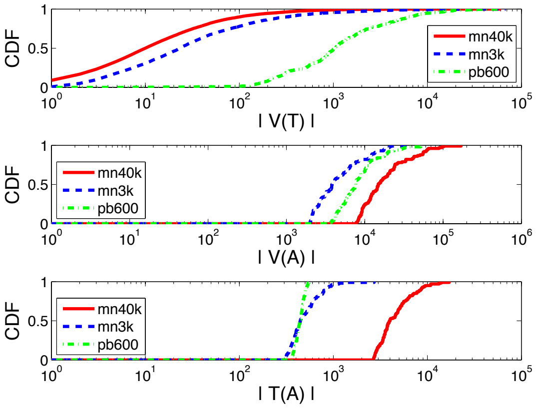


Fig. 1. Summary statistics for the measured BitTorrent demographics. Cdfs for: $|V(T)|$, the number of clients in a torrent, $|V(A)|$, the total number of clients in an ISP across all its torrents, and $|T(A)|$, the number of distinct torrents requested by the clients of an ISP.

B. High level characterization of the dataset

We use the following definitions. We let $\mathcal{T}$ denote the set of torrents appearing in our measurements and $\mathcal{A}$ the set of ISPs that have clients in any of the torrents of $\mathcal{T}$. We let $T(A)$ denote the set of torrents that have at least one active client in A, and $V(A) = \bigcup_{T \in T(A)} V(A,T)$ the set of all clients of A participating in any of the torrents $T(A)$. In Fig. 1 we summarize the measured demographics. Some points worth noting: (i) The largest torrent has approximately 60K clients in all three datasets. Looking at the large set, `mn40K`, we see that most torrents are small as has already been shown [28], [26]. `mn3K` has relatively bigger torrents because it is a subset of most recent torrents of `mn40K`, and recency correlates with size. `pb600` holds by definition only big torrents. (ii) Looking at the number of peers and torrents per ISP we see that `mn40K` has bigger values which is expected as it is a much bigger dataset than the other two and thus contains more and bigger ISPs (notice that in Table I the numbers for `mn40K` are per snapshot, whereas for the other two are aggregates over a day, *i.e.*, totals from 24 snapshots).

---

[1] Note that this crawler follows the guidelines defined in [35].

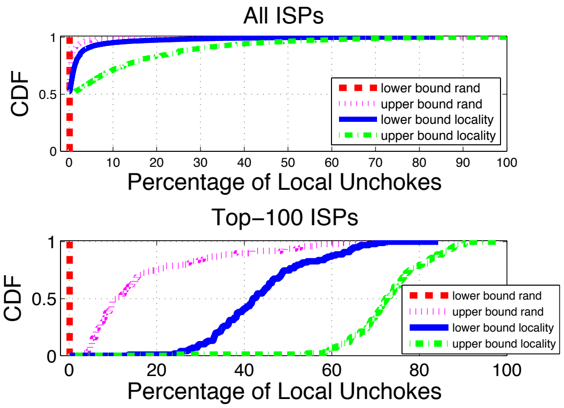


Fig. 2. CDF of the upper and lower bound on the number of localized unchokes under Random and Locality for `mn40K` in number of clients. Top: All ISPs. Bottom: Top-100 ISPs.

C. Speed agnostic bounds for the measured demand demographics

In Sect. III we defined the notions of sparseness and denseness for one ISP and a single torrent and noted that sparseness helps to localize traffic whereas denseness makes it harder. Therefore, by assuming that all the torrents $T(A)$ downloaded in A are concurrently in sparse mode we can get an upper bound on the expected number of unchokes that can be localized by an overlay construction policy for the given demand demographics and any speed distribution among different ISPs. Similarly, by assuming that all torrents are in dense mode we get a lower bound. In Fig. 2 we plot the upper and lower bound on localized unchokes for Random and Locality for the 10.3K (top) and top-100 (bottom) ISPs in number of clients in the `mn40K` dataset. These bounds were computed using formula (1) and its corresponding version for dense mode for single torrents and iterating over all $T \in T(A)$ from our demographics dataset adding each contribution with weight $|V(A,T)|/\sum_{T' \in T(A)} |V(A,T')|$ to capture the relative importance of T for A. Note that we use W = 40 for all the experiments because it is the default value used by most BitTorrent clients. We observe that around half of the 10.4K ISPs are unable to localize any traffic under either Random or Locality even for the most favorable case represented by the sparse mode. These are ISPs with a low number of BitTorrent clients that download torrents where there are no other local peers. We refer to these torrents as "unlocalizable" torrents. The low benefit that Locality techniques may bring to small ISPs (*i.e.*, those having a small population of BitTorrent clients) has been also pointed out by other works in the literature [17], [28], [33]. If we consider now the top-100 ISPs, that represent around 1% of the ISPs but account for more than 68% of all IPs in the dataset, the lower bound for Random is 0 for 95% of these ISPs and reaches a maximum value of 0.03%. This happens because for the huge majority of torrents, an ISP has only a small minority of the total nodes in the torrent. In dense mode, Random needs to get most of these few locals with a random draw which is an event of very small probability. On the other hand, this small minority of nodes performs much better in sparse mode yielding an upper bound for Random that is at least 10.94% in half of the top-100 ISPs. Locality has strikingly better performance. Its lower bound is at least 42.35% and its upper bound 72.51% in half of the top-100 ISPs. The huge improvement comes from the fact that Locality requires the mere existence of few local nodes in order to keep most unchokes inside an ISP. Note that the improvement factor is greater in the difficult case[2] (the lower bound goes from 0 to above 42% in half of the cases) while it is also quite big in the easy case (improvement factor of at least 6.63 in half of the cases). We have recomputed these bounds for the top-100 ISPs based on the `pb600` dataset that is biased towards large torrents. The results are presented in Appendix E-A2.

Note that these bounds paint, to the best of our knowledge, the most extensive picture reported up to now in terms of covered ISPs and torrents of the potential of locality given the constraints set by real demand demographics.

D. Factoring the effect of speed

The notions of sparseness and denseness have been useful in deriving speed-agnostic performance bounds based on the demand demographics and the overlay construction policy. To refine our analysis and answer more detailed questions we turn our attention now to the effect of speed. We do so through what we call *Inherent Localizability*. Let $A(T)$ denote the set of ISPs that have clients in torrent T. Let also $U(A)$ denote the uplink speed of nodes in ISP A. We focus on the uplink speeds because they are typically the bottleneck in highly asymmetric residential broadband accesses [11]. For now it suffices to assume that speeds differ only between ISPs. We define the *Inherent Localizability* $I_q(A,T)$ of torrent T in ISP A as follows:

$$I_q(A,T) = \frac{|V(A,T)|}{\sum_{A' \in A(T)} |V(A',T)| \cdot \mathcal{I}(A,A',q)},$$

where, $\mathcal{I}(A,A',q) = 1$ iff $U(A) \cdot (1-q) \leq U(A') \leq U(A) \cdot (1+q)$, and 0 otherwise. The parameter $q \in [0,1]$, captures the maximum speed difference that still allows a local node of A and a remote node of A' to unchoke each other. In reality q can be arbitrarily large because very fast nodes can unchoke much slower ones in the absence of other fast nodes. However, this simple metric suffices to understand interesting effects produced by the combination of speed and demographics. The inherent localizability $I_q(A)$ of ISP A across all its torrents is simply the weighted sum by $|V(A,T)|/|V(A)|$ of its $I_q(A,T)$'s for all torrents it participates in. $I_q(A)$ captures the density of A's nodes in torrents that it shares with other ISPs that have similar speed. Due to stratification, unchokes will take place among those nodes. For Random, $I_q(A)$ determines completely its ability to localize unchokes. $I_q(A)$ also impacts on Locality. However, Locality's overall performance depends on the absolute number of local peers.

E. Does being faster help in localizing better?

In this section we use inherent localizibility to study the effect of access speed on the ability of Random to keep unchokes internally in an ISP. ISPs have a natural interest in this question because on the one hand they want to upgrade their residential broadband connections to fiber but on the other hand, they wonder how this will impact their transit and peering traffic. Next we present a case study showing that it is difficult to come up with such predictions without using detailed demographic/speed information and corresponding methodologies to capture their combined effect.

---

[2] A detailed explanation for this can be found in Appendix E-A1.

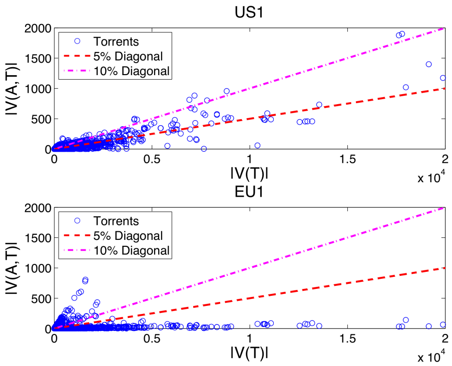


Fig. 3. Nodes in ISP A, $|V(A,T)|$, vs. total torrent size, $|V(T)|$, for US1 (top) and EU1 (bottom).

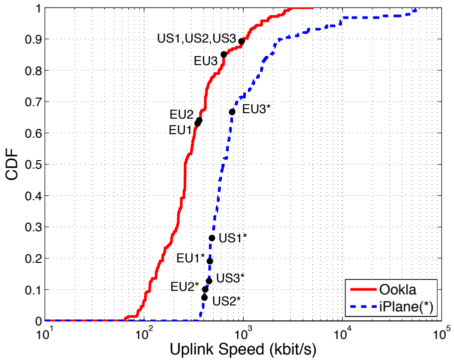


Fig. 4. CDF of uplink speeds per country. EU1–EU3, US1–US3 are ISPs studied in Sect. VII.

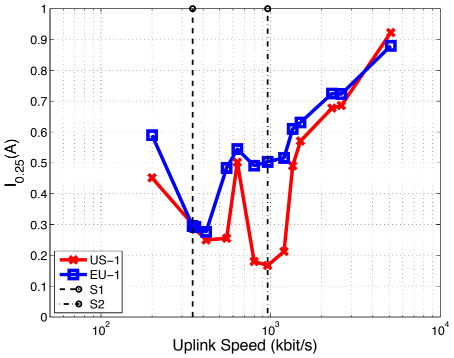


Fig. 5. The inherent localizability of US1 and EU1 for different speeds based on all their torrents (S1 = 347 kbit/s and S2 = 960 kbit/s).

1) A European and an American ISP: Consider the following two ISPs from our dataset `mn40K`: US1, with the largest population of nodes in America (according to our different datasets) and median upload speed 960 kbit/s, and EU1, with the largest population of nodes in Europe and median upload speed 347 kbit/s. In Fig. 3 we plot $|V(A,T)|$ vs. $|V(T)|$ for all $T \in T(A)$ for the two ISPs. A quick glance at the figure reveals that the two ISPs are radically different in terms of demand demographics. Because of the proliferation of English and its English content, US1 is participating in globally popular torrents. In the figure, the US1 torrents that are globally large (high $|V(T)|$) have a lot of clients inside US1. Also, torrents that are popular in US1 are also globally popular. In EU1 the picture is very different. The largest torrents inside EU1 are not among the largest global ones, whereas only very few globally popular torrents are also popular inside EU1. This has to do with the fact that EU1 is in a large non-English speaking European country that produces and consumes a lot of local, or locally adapted content.

2) The impact of demographics and speed on inherent localizability : We will now compute the inherent localizability of EU1 and US1. To do this we need the speeds $U(A)$ for all A that participate in common torrents with the two ISPs. We have obtained these speeds from the Ookla Speedtest service [2]. This dataset provides the median upload and download access speeds for 215 different countries. To this end, it leverages measurements of over 19 million IP client addresses around the world. For half of the countries the speed values are obtained from more than 4.5k samples whereas just 16.7% of the countries present less than 350 samples. In Fig. 4 we plot the cdf of median country speed based on the above dataset. It is interesting to observe that almost 80% of the countries have similar speeds that are below 610 kbit/s where the few remaining ones are sparsely spread in the range from 610 kbit/s to 5.11 Mbit/s. We also plot the corresponding cdf from iPlane [25] which we use in Appendix E-C3 for validation.

Using the above demographics and speeds we plot in Fig. 5 the localizability of ISP $A \in$\{EU1,US1\} for different $U(A)$, *i.e.*, we plot how the localizability of the two ISPs would change if we changed their speeds while keeping the speeds of all other ISPs fixed. We have assumed $q = 0.25$. Results are similar for most $q < 0.5$ whereas for larger ones speed starts becoming marginalized because high q's imply that any node can unchoke any other one. There are two points to keep from this figure. First, the localizability of EU1 is generally higher than that of US1 for the same speed. This means that if the two ISPs had similar speed, then the demographic profile of EU1 depicted earlier in Fig. 3 would lead to a higher inherent localizability as this ISP holds a larger proportion of the content requested by its users. Thus Random would perform better in EU1 than in US1. A second point to notice is that $I_{0.25}(A)$ is changing non-monotonically with $U(A)$. This happens because the set of remote ISPs and consequently the number of remote clients that can be unchoked by clients of A due to similar speed (within the margins allowed by a given q) changes as we vary the speed of A. If the torrents were spread uniformly across all the ISPs, and ISPs had similar size, then due to the sparsification of ISPs on the high speed region (Fig. 4), $I_{0.25}(A)$ would increase monotonically with $U(A)$. The real demographics and sizes of ISPs, though, lead to the depicted non-monotonic behavior that exhibits only a general trend towards higher intrinsic localizibility with higher local speed. This has important consequences on the expected amount of transit traffic under different speeds. For example, by going from speed S1 = 347 kbit/s to S2 = 960 kbit/s, the inherent localizability of EU1 increases from around 0.3 to around 0.5 and as a consequence its transit traffic under Random would decrease as more unchokes would stay inside the ISP. The opposite however happens for US1. Increasing the speed from S1 to S2 reduces the inherent localizability from 0.3 to 0.2, effectively increasing the number of unchokes going to remote nodes and thus the transit traffic as well. Further details on the combined effect of demand demographics and speeds on different type of torrents can be found in Appendix E-B.

In conclusion, the interplay between speed and demographics is complicated, an ISP can use our methodology to actually obtain an informed prediction of the impact of planned changes to its residential broadband offerings on its transit traffic.

V. BITTORRENT TRAFFIC MATRICES

Our analysis up to now has revealed important insights about the parameters that affect the performance of Random and Locality. However, in order to achieve our final goal (*i.e.*, to estimate the aggregate amount of traffic routed to an ISP transit link due to the torrents of our demographic datasets) we require a more sophisticated methodology able to accurately predict traffic matrices for thousands of torrents including clients from an ISP and model the behaviour of seeders and optimistic unchokes from leechers. In Sect. VII we present a methodology that accurately models these aspects. In this section we introduce the basic elements of this model.

We start with fast numeric methods (based on b-matching theory) that capture the unchoking behavior in steady-state, *i.e.*, when the Least Replicated First (LRF) chunk selection algorithm [8] has equalized the replication degree of different chunks at the various neighborhoods. From that point in time on, we can factor out chunk availability and estimate the established unchokes based only on the uplink speed of nodes. We extend this numeric method to capture also the initial flash-crowd phase of a torrent. The resulting model is much slower in terms of execution time and provides rather limited additional fidelity since the flash crowd phase is known to be relatively short compared to the steady-state phase of sufficiently large downloads (the size of a movie or a software package) [18], [20], [22]. For this reason we stick to our original faster model. Even without chunk availability information, this constitutes a substantial improvement over the current state of the art in P2P matrix computation [6] which is based on a gravity model driven by total number of peers per ISP (no demographics, speed, seeder/leecher information). Details of our numeric methods can be found in Appendix B.

Notice that although experimentation with real clients would provide higher accuracy in predicting the QoS of individual clients, it would not be able to scale to the number of torrents and clients needed for studying the impact of realistic torrent demographics at the ISP level (aggregate traffic in the order of several Gbit/s). Our scalable numeric methodology targets exactly that while preserving key BitTorrent properties like leecher unchoking (regular and optimistic) and seeding. We validate the accuracy of our methods against real BitTorrent clients in controlled emulation environments (in Appendix B-C) and in the wild with live torrents (Sect. VIII).

A. Modeling Leechers

Estimating the traffic flow among leechers is an involved task due to the unchoke algorithm [8]. This reciprocity based matching algorithm of nodes with similar speeds has many of the elements of a b-matching problem [5], [15]. In Appendix B-A we show how to cast the problem of estimating the traffic matrix from a torrent T downloaded by nodes in $V(T)$ as a b-matching problem in $V(T)$. We also pointed to work describing how to get a fast solution (a stable matching M) for the b-matching. M gives us the pairs of nodes that unchoke each other in steady-state. Using the stable matching M and the uplink speeds of nodes, we can compute the expected rate at which a node v uploads to its neighbor u:

$$upload(v,u) = \begin{cases} \frac{U(v)}{k+1}, & \text{if } (v,u) \in M \\ \frac{U(v)}{k+1} \cdot \frac{1}{|N(v,T)|-k}, & \text{otherwise} \end{cases}$$

The first case amounts to neighbors u that are allocated one of v's k regular unchokes in steady-state. The second case amounts to the remaining neighbors that receive only optimistic unchokes and thus share the single slot that is allocated optimistically.[3]

B. Modeling Seeders

Let $N(s,T)$ be the neighborhood of a seeder node s of torrent T. Existing seeders typically split their uplink capacity $U(s)$ among their neighbors following one of two possible policies. In the *Uniform* policy, all neighbors $u \in N(s,T)$ get an equal share $upload(s,u) = U(s)/|V(s,T)|$. In the *Proportional* policy, neighbor $u \in N(s,T)$ gets an allotment in proportion to its speed, *i.e.*, $upload(s,u) = U(s)U(u)|/\sum_{u' \in N(s,T)} U(u')$.

In Sect. VII we will use the seeder bandwidth allocation policies and the upload rates from the b-matching described in this section to compute the amount of BitTorrent traffic crossing inter-ISP links. Before that, however, we introduce the overlay construction policies that we will study.

VI. LOCALITY-BIASED OVERLAYS

Up to now our discussion has been based on an very basic locality biasing overlay construction algorithm, *Locality*, that provides a node v of A participating in T with $\min(W, |V(A,T)|-1)$ local nodes and pads up to W with randomly chosen remote nodes. In the evaluation presented in Sect. VII we will consider the following additional overlay construction policies:

– *Local Only If Faster (LOIF)*: In this case switches of remote for local nodes occur only if the local ones are faster. LOIF is an end-user QoS preserving policy.

– *Strict*: As in *Locality* all switches of remotes for locals are performed. Of the remaining remotes only one is retained and the rest are discarded from the neighborhood.

Note that *Locality*, *LOIF* and *Strict* are members of an extended family of overlay construction algorithms defined in Appendix C.

VII. IMPACT OF LOCALITY ON ISPS & USERS

The bounds presented in Sect. III provide a broad view of the impact of locality on the transit traffic of thousands of ISPs. They do not, however, provide any information regarding the impact of locality on end user download rates. Earlier work [34], [7] has demonstrated some "win-win" cases in which ISPs benefit by reducing their transit traffic, while at the same time their users get faster download rates. In general, however, talking mostly to local nodes can harm a user's download rate by, *e.g.*, depriving it from faster remote seeders and leechers (the latter can provide optimistic unchokes). Whether this happens depends on the interplay between demographics and speed. In this section we employ the traffic matrix computation methodology of Sect. V to present detailed case studies of the impact of different overlay construction mechanisms from Sect. VI on ISPs and their users. *We are primarily interested in discovering the boundaries of the win-win region from locality for both ISPs and users as well as the reasons behind them.*

A. Experimental methodology

Next we present the common parts of our methodology that appear in all experiments. Experiment-specific parts appear in the corresponding sections.

1) Input to the experiments: **Demographics:** We used the BitTorrent demand demographics measurements presented in Sect. IV. Our default dataset will be `mn40K`. We use `pb600` for validation in Appendix E.

Speed distributions: We assign to an ISP the median uplink speed of its country[4] [2]. We also use client speed distributions within an ISP from iPlane [25] in Appendix E. These

---

[3] It might be the case that in a stable solution node v is matched to less than k others (e.g., because it is of low preference to its neighbors). In such cases we add the unallocated unchoke bandwidth to the optimistic unchoke bandwidth that is evenly spread to choked neighbors.

[4] Note that this is a limitation of the Ookla dataset, not a choice of the authors.

(a) Transit traffic reduction

ISP	LOIF	Locality	Strict
US1	**32.00%**	**55.63%**	**97.47%**
US2	28.47%	48.40%	97.25%
US3	26.04%	41.45%	97.02%
EU1	**10.50%**	**39.12%**	**96.41%**
EU2	11.34%	44.89%	95.95%
EU3	16.18%	35.57%	96.98%

(b) Degradation of median QoS

ISP	LOIF	Locality	Strict
US1	**-6.71%**	**-1.32%**	**2.88%**
US2	-5.22%	-0.83%	4.43%
US3	-5.74%	-1.27%	4.96%
EU1	**-1.47%**	**3.33%**	**18.59%**
EU2	-0.55%	6.35%	11.72%
EU3	-3.21%	2.28%	14.67%

TABLE II
RESULTS FOR ISPS EU1-EU3, US1-US3, UNDER `mn40k` AND OOKLA SPEEDS.

speeds represent last mile bottlenecks. We consider network bottlenecks and other limitations of our speed datasets later in Sect. VIII using an experimental prototype and live torrents.

Seeder/leecher ratios: In dataset `pb600` we know exactly if a peer is a seeder or a leecher but in `mn40K` and `mn3K` we do not have this information. To solve this problem, we obtained from the correspondent tracker the number of seeders and leechers for each torrent. Then we made a client in our dataset a seeder with probability equal to the seeder/leecher ratio of its torrent. Thus although we do not have the exact identities of seeders, we do match the real seeder/leecher ratios. We validated this technique against the dataset `pb600` and got a minor variation compared to the ground truth.

2) Traffic matrix computation: In our experiments we are interested in quantifying the effects of locality biased overlay construction on a "home" ISP A assuming that other ISPs operate under Random. In effect we are interested on access ISPs connecting residential users to the Internet. Such ISPs care primarily to reduce their transit traffic. Transit ISPs on the other hand have the exact opposite objective. Additionally, at this early stage very few (if any) ISPs have adopted P4P-like strategies. For the above two reasons we leave the study of the interplay between multiple ISPs with different incentives with respect to locality to future work. Returning to traffic matrix computation, we perform it as follows.

(1) Using our measured demand demographics we identify the set of clients $V(T)$ for each torrent $T \in T(A)$ downloaded by clients in our home ISP A. We construct Random, LOIF, Locality, and Strict overlay graphs among the nodes in $V(T)$ as described in Sect. VI. We select the nodes to be seeders as described in Sect. VII-A1 and assume that they perform proportional seeding[5] .

(2) We feed each overlay graph resulting from the combination of the demographics of a torrent T and an overlay construction algorithm, together with an uplink speed distribution to the BitTorrent traffic matrix computation methodology detailed in Sect. V and Appendix B. The outcome is a traffic matrix indicating the transmission rate between any two nodes $v, u \in V(T)$.

(3) We adopt a simplified version of routing according to which all traffic between clients of our home ISP and an ISP of the same country goes over unpaid peering links, whereas traffic between clients of our home ISP and another ISP in a different country goes over a paid transit link. This simplified routing is actually on the conservative side, as it reduces the amount of traffic going to the transit link and thus also the potential gains from applying locality.

Repeating steps (1)–(3) for all torrents in $T(A)$ we obtain the aggregate amount of traffic going to the transit link of A due to the torrents appearing in our dataset.

3) Performance metrics: We study two performance metrics. The first one, *transit traffic reduction compared to random* is of interest to the home ISP. It is defined as follows: ((aggregate transit under Random)-(aggregate transit under Locality(δ, μ))) / (aggregate transit under Random). The second one, *user QoS reduction* is of interest to the clients of the home ISP. It is defined as follows: (q_x(download rate under Random)-q_x(download rate under Locality(δ, μ))) / q_x(download rate under Random), where q_x denotes the x-percentile of download rate computed over all nodes of home ISP. If not otherwise stated we will use the median ($x = 0.5$). Note, that file unavailability due to lack of pieces is also a metric of interest to BitTorrent clients. However different standard techniques of the BitTorrent protocol (e.g. optimistic unchoke) take care of this issue that based on experiments conducted in live swarms seems to not be a limitation of Locality techniques [34] [7]. Therefore, in this paper we focus just on the *user QoS reduction* to evaluate the impact of Locality solutions on the QoS perceived by users.

B. Comparing overlays

In Table II(a) we present the transit traffic reduction under various locality policies with respect to Random for the 6 largest ISPs[6] (3 from Europe and 3 from US) across our different datasets using uplink speeds from [2]. In Table II(b) we present the corresponding impact on user QoS. We will focus mainly on ISPs EU1 and US1, introduced earlier in Sect. IV-E.

1) Without bounding the number of inter-ISP links: We begin with "mild" locality policies that do not enforce constraints on the number of remote neighbors. The mildest of all, LOIF, replaces remote with local nodes in the neighborhood only if the locals are faster. In the case of US1 this yields a transit traffic reduction of 32% compared to Random. The corresponding value for EU1 is 10.5%. US1 is faster than EU1 and performs more switches of remotes for locals under LOIF and thus gets a higher reduction of transit traffic. Looking at Table II(b) we see that US1 pays no penalty in terms of QoS reduction for the end users from LOIF. Actually, the median value gets a slight speedup indicated by negative values (see Appendix E-C1 for other percentiles). The situation for EU1 is similar. The preservation of at least the same user QoS is an inherent characteristic of LOIF which by default leads to a win-win situation for both ISPs and users. The transit savings of LOIF can however be small, as in the case of EU1.

We can reduce the transit traffic further by imposing a less strict switching rule. Locality switches any remote client with a local one independently of speed. This increases the savings for US1 to 55.63% compared to Random whereas the corresponding number for EU1 rises to 39.12%. This is the highest transit reduction that can be expected without limiting the number of inter-ISP overlay links. This additional transit traffic reduction does not impose any QoS penalty on the customers of US1. EU1 customers on the other hand pay a small reduction of QoS of 3.33% because they lose some faster remote neighbors (EU1 is not among the fastest ISPs according to the country speeds depicted in Fig. 4). Under Locality win-win is not guaranteed but rather it depends on speed and demographics. For US1 Locality is again a clear win-win whereas for EU1 it is almost win-win.

---

[5] We use proportional seeding because it is the most extended technique.

[6] We have obtained similar results for other large ISPs.

(a) Transit traffic reduction

ISP	LOIF	Locality	Strict
M1	7.97%	18.09%	96.82%
M2	12.00%	21.79%	97.23%
S1	9.17%	13.16%	97.22%
S2	14.01%	26.02%	98.20%

(b) Degradation of median QoS

ISP	LOIF	Locality	Strict
M1	-2.14%	0.92%	37.55%
M2	-1.37%	2.99%	31.11%
S1	-0.05%	-1.09%	56.45%
S2	-2.56%	2.84%	73.85%

TABLE III
RESULTS FOR ISPS M1,M2 AND S1,S2 ISPS UNDER `mn40K` AND OOKLA SPEEDS.

2) Unlocalizable torrents: In the aforementioned results the transit traffic reduction has its upper bound at around 55%. This happens because the demographics of both US1 and EU1 include a long tail of torrents with very few local nodes. These torrents are "unlocalizable" in the sense that all overlay links for them will have to cross the transit link if the corresponding clients are to be given the standard number of neighbors according to BitTorrent's bootstrapping process (40-50 depending on version). The unlocalizable torrents put rigid limits on the transit reduction that can be achieved without enforcing constraints on the number of allowed inter-ISP overlay links. Interesting, although the unlocalizable torrents create most of the transit traffic, they are requested by a rather small percentage of the nodes of an ISP. In US1 90% of transit traffic under Locality is due to only 10% of the nodes. In EU1 90% of transit traffic is due to 13.44% of nodes.

3) Bounding the number of inter-ISP overlay links: If we want further transit traffic reductions then we need to control the unlocalizable torrents by enforcing strict constraints on the number of inter-ISP overlay links. In the last column of Table II(a) we depict the performance of Strict that permits up to 1 inter-ISP overlay link per torrent for a given client. Indeed in this case the transit traffic reduction is huge (around 96%-97% for both networks). The median user QoS drops by 18.59% in EU1. The situation is much better for US1 where the median speed drops by around 3%. However, nodes that are downloading unlocalizable torrents pay a heavy penalty of almost 99%.

C. Comparing ISPs

Inspecting Table II(a) we see that American ISPs in general achieve higher transit traffic reduction than European ones, across all locality biased overlay construction policies. We attribute this to the fact that Random performs very poorly in those ISPs because their content is more scattered around the world (they have smaller Inherent Localizability, Sect. IV-E). When comparing among American or among European ISPs, the observed differences correlate mostly with the size of the ISP. The reason is that in ISPs with approximately the same Inherent Localizability (*e.g.*, the 3 American ISPs), Random performs approximately the same, and thus any difference in transit reduction comes from the performance of Locality or LOIF. The latter depends on the absolute size of the ISP as a larger ISP can gather more easily enough local peers to reach the minimum number required by the bootstrapping process.

To validate our previous observations and enrich our discussion, Table III shows the transit traffic reduction and the degradation of median QoS for two ISPs of medium size (M1 and M2) and two small ISPs[7] (S1 and S2). We consider medium size and small ISPs as those having 1 and 2 order of magnitude less peers than the largest studied ISPs, respectively. First, the results confirm that independently of the size of the ISP, LOIF leads by default to a win-win situation. Locality shows an almost win-win situation, however under this policy the transit traffic reduction with respect to Random is significantly lower for these small ISPs than for the largest American and European ISPs. This is caused by the presence of more peers in unlocalizable torrents. In addition, as we have seen, Strict policy dramatically affects the QoS for peers in unlocalizable torrents. Therefore, smaller ISPs (hosting a larger number of peers downloading unlocalizable torrents) present higher median QoS degradation under Strict policy as shown by Table III(b).

VIII. VALIDATION ON LIVE TORRENTS

In our study up to now, we have only considered last mile bottlenecks imposed by access link speeds but no network bottlenecks due to congestion or ISP traffic engineering, including throttling [12]. In addition, although we evaluate the accuracy of b-matching in a controlled emulation environment (Appendix B-C), we can obtain further insights by testing our results in the wild where we can observe additional effects from delays and losses that are lacking from an emulation environment that captures only bandwidths. To address such issues we integrated LOIF, Locality, and Strict(μ) into the mainline Bittorrent client[8] (version 5.2.2). Next we describe briefly some implementation issues and then move to present the results of connecting to live torrents with our modified client.

A. Prototype implementation

Integrating these policies into the existing BitTorrent clients requires addressing some new requirements. First, we need to know for every Bittorrent client its ISP and country. For this, we use the MaxMind geolocation database [1]. Next, we need to discover the list of local clients in order to be able to substitute remote ones with local ones. For this, we use the PEX messages to discover all the participants of the swarm, and then use the MaxMind database to classify them. Finally, the last requirement which is specific to LOIF, is to estimate the speed of the local and remote clients. For this, we monitor the rate at which the clients send us HAVE messages, which indicates how fast they download. Finally, notice that the above method works only for inferring the speed of leechers. For seeders, we can only infer the speed of those seeders that unchoke us. Thus in LOIF we do not switch neighbors for which we do not have speed information.

Next, we briefly describe our implementation of LOIF. Many implementation decisions are influenced by how the python mainline Bittorrent client is designed and implemented. Every 40 seconds we perform the following steps:

(1) We classify the active neighbor peers in 3 lists: L_{ISP} which contains all peers that belong to the same ISP as the client, $L_{Peering}$ which contains all peers that are in ISPs with peering relationships and L_{Remote} which contains other peers.

(2) For every peer $R_i \in L_{Remote}$, we close the connection if there exists a peer $L_j \in L_{ISP}$ with higher estimated download speed. If such a peer does not exist then we check if there

---

[7] Again, we have repeated the experiment for several medium size and small ISPs obtaining similar results.

[8] We chose the mainline client because it was one of the most popular (open source) BitTorrent clients after μTorrent and Vuze. Moreover, μTorrent is not open source and hacking Vuze is rather complex.

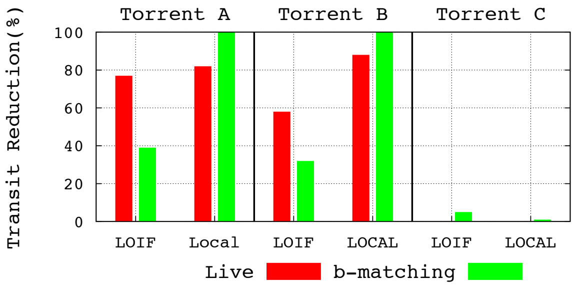


Fig. 6. Comparision between LOIF and Local

	Local	Remote	Percentage of Seed
Torrent A	521	46	63.2%
Torrent B	351	211	12.5%
Torrent C	3	666	66.4%

TABLE IV
LIVE TORRENT CHARACTERISTICS

exists a peer $C_j \in L_{Peering}$ with higher estimated download speed and in this case again we close the connection.[9]
(3) For each connection that was closed in the last step, the algorithm opens a new one, giving preference, first to those IPs that belong to the same ISP, then to those IPs belonging to peering ISPs and, finally, to those IPs belonging to other ISPs.

An important detail in our implementation is to always have a minimum number of neighbors (at least 40). This holds for LOIF and Locality, but not for Strict. For Strict(μ), we close connections and do not open new ones, if we have more than μ remote nodes.

B. *Experimental methodology*

We ran our modified Bittorrent client from an ADSL connection of ISP EU1. In all the torrents we first warmed up by downloading 30MB to avoid BitTorrent's startup phase. In each run, we re-initialize back to the same 30MB. Next, we download 50MB with each of the following policies: Random, LOIF, Locality, and Strict. We repeated each test 5 times, and reported averages over all runs. During each experiment we logged the list of IPs and the number of neighbors and used them later as input to our traffic matrix estimation technique of Sect. V. This way, we can compare the estimated transit savings with the real one on live torrents.

C. *High, medium, and low localizability torrents*

We used our prototype to validate some of our previous results. Although we cannot scale to the number of torrents discussed in Sect. VII, we tested torrents at characteristic points of the demographic spectrum. In particular, we tested a very popular torrent inside EU1 (Torrent A), an intermediately popular one (Torrent B), and an unpopular one (Torrent C). In Table IV we summarize the characteristics of the 3 torrents. In Fig. 6 we present the transit traffic savings as predicted by our traffic matrix estimation method and as measured on the live torrent under LOIF and Locality. We do not present results under Strict as they were always in perfect agreement.

Overall we see that the results under Locality are pretty consistent – estimation and measurement are within 10-20% of each other. In terms of absolute values things are as expected: in cases A and B there are enough local nodes to eliminate almost all transit traffic whereas in C there is 0 saving as there do not exist any local nodes to switch to. The difference between the 100% savings predicted by b-matching in A and B and the ones measured in practice has to do with implementation restrictions. As mentioned earlier, we update the overlay every 40 sec (which is equal to 4 unchoke intervals). During that time new incoming remote connections are accepted and can lead to unchokes that create transit traffic and thus eat away from the 100% saving expected upon overlay update instants when all remote connections are switched with local ones.

Under LOIF, the deviation between estimation and measurement is substantial: the measured transit saving is twice as big as the estimated one. To interpret this, we looked at the number of switches of remote nodes for local ones that LOIF performed in practice and realized that they were much more than we would predict. This in effect means that the real LOIF found the remote nodes to be slower than what expected from our speed dataset from Ookla [2]. We attribute this to several reasons. On the one hand, Ookla provides access speeds, however the actual bandwidth that a client dedicates to download a torrent is likely to be lower than its access speed. For instance, the user can be using other applications at the same time, she can be downloading more than one torrent in parallel thus splitting the bandwidth among them or simply she can limit the maximum bandwidth dedicated to Bittorrent. On the other hand, network bottlenecks or throttling at the inter-ISP links, in the origin ISP or the ISPs that host the remote nodes, reduce the actual bandwidth of clients in BitTorrent swarms. Although certainly interesting, identifying exactly why the remote nodes appear slower than expected is beyond the scope of the current work. See [12] for more.

IX. RELATED WORK

A. *Early work on locality-biasing*

One of the early works on locality-biased overlay construction was Karagiannis et al. [19]. Using traces from a campus network as well as a six-month-long logfile from a popular torrent, they showed that there is substantial overlap in the torrents downloaded by co-located clients. Another early work from Bindal et al. [4], studied the effect of limiting the number of inter-AS connections using simulations with synthetic demand. Aggarwal et al. [3] studied the effects of locality biasing on the Gnutella overlay. Apart from studying a different P2P system, they differ from our work by focusing on the overlay graph theoretic properties whereas we care about the traffic matrix.

B. *Recent systems for locality-biasing*

Following up on positive results on the potential of locality-biasing, a number of actual systems like P4P [34] and ONO [7] have appeared recently for the BitTorrent P2P protocol. The previous works focus on architectural and systems questions regarding "how" to implement locality-biasing, and in particular whether the goal can be achieved through unilateral client-only solutions, or bilateral cooperation is essential for making locality work for both ISPs and users. In terms of reported results, [34] presents a variety of use cases for P4P over different networks and P2P applications like Pando and Liveswarms. In all cases however, results are based on one

---

[9] The assumption is that nodes in the same country communicate over peering links. In our implementation we do not infer ISP relationships but we can do so with iPlane Nano [24].

or a few swarms and thus do not capture the aggregate effects created by tens of thousands of concurrent swarms with radically different demographics. The results reported in [7] on the other hand, are indeed from multiple torrents and networks, but they only report on the final outcome from using the ONO system without explaining how the demographics of the torrents and the speeds of the ISPs affect these outcomes. The main driving force behind our work is to explain "when" locality works and "why" when it does so and thus help in interpreting the results from systems like P4P and ONO or others to come in the future. Locality biasing has also been applied to P2P streaming systems [29].

C. BitTorrent measurements

A substantial amount of work has gone into BitTorrent measurements [18], [16], [30], [26]. These works go beyond locality to characterize things like the arrival pattern of new nodes, the seeding duration, the seeder/leecher ratios, etc. Our work apart from performing large scale measurements develops scalable methodologies that permit distilling non-trivial conclusions regarding the interplay of demographics, speed, and overlay construction. Relevant to our work is the recent work of Piatek et al. [28]. It discusses the potential for win-win outcomes for ISPs and users but puts most of its emphasis on implementation issues and the consequences of strategically behaving ISPs. Our work, on the other hand, is of performance evaluation nature and aims at pushing the envelope in terms of both the scalability and the fidelity of our evaluation methodology. Our dataset is large; we compute transit reduction from our 40K torrents whereas they use only 1000 torrents out of their 20K dataset. In terms of methodology, we capture the effect of stratification from choke/unchoke whereas [28] assumes cooperative clients and does not model the effect of speed. The only measurement work on BitTorrent traffic matrix estimation that we are aware of is due to Chang et al. [6]. It is based on a gravity model driven by the total number of peers per ISP. Thus it does not consider demographics, speed, seeder/leacher information nor it can be used for estimating the traffic from arbitrary non-random overlay construction policies.

In conclusion, the combination of the dataset that best approximate a real snapshot of a large number of BitTorrent swarms and an accurate and scalable methodology for the prediction of BitTorrent traffic matrices allows us to estimate tight bounds for the Transit Traffic reduction produced by different Locality policies.

X. CONCLUSIONS

In this paper we collected extensive measurements of real BitTorrent demand demographics and developed scalable methodologies for computing their resulting traffic matrix. Based on this we quantified the impacts of different locality-biasing overlay construction algorithms on ISPs and end-users. By studying real ISPs, we have shown that a large fraction of (very small) ISPs do not have enough resources to localize traffic. However, locality yields win-win situations for medium size and large ISPs. The win-win profile is bounded by "unlocalizable" torrents that have few local neighbors. Handling the unlocalizable torrents requires limiting the number of allowed inter-ISP overlay connections. This has a small impact on the average user but a dire one on the users of unlocalizable torrents.

REFERENCES

[1] MaxMind- GeoIP. info at http://www.maxmind.com/app/ip-location.

[2] Ookla's speedtest throughput measures. https://confluence.slac.stanford.edu/display/IEPM/Ookla' s+Speedtest+Throughput+Measures.

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

[4] 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'06*.

[5] K. Cechlárová and T. Fleiner. On a generalization of the stable roommates problem. *ACM Trans. Algorithms*, 1(1):143–156, 2005.

[6] H. Chang, S. Jamin, Z. M. Mao, and W. Willinger. An empirical approach to modeling inter-AS traffic matrices. In *ACM IMC'05*.

[7] D. R. Choffnes and F. E. Bustamante. Taming the torrent: a practical approach to reducing cross-isp traffic in peer-to-peer systems. In *Proc. of ACM SIGCOMM '08*.

[8] B. Cohen. Incentives build robustness in BitTorrent. In *First Workshop on Economics of Peer-to-Peer Systems*, 2003.

[9] R. Cuevas, M. Kryczka, A. Cuevas, S. Kaune, C. Guerrero, and R. Rejaie. Is content publishing in bittorrent altruistic or profit-driven? In *Proceedings of the 6th International COnference*, page 11. ACM, 2010.

[10] R. Cuevas, N. Laoutaris, X. Yang, G. Siganos, and P. Rodriguez. Deep diving into BitTorrent locality. In *IEEE INFOCOM'11*.

[11] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In *ACM IMC '07*.

[12] M. Dischinger, A. Mislove, A. Haeberlen, and K. P. Gummadi. Detecting BitTorrent blocking. In *ACM IMC'08*.

[13] R. Farahbakhsh, A. Cuevas, R. Cuevas, R. Gonzalez, and N. Crespi. Understanding the evolution of multimedia content in the internet through bittorrent glasses. *IEEE Network (Accepted for Publication)*, 2013.

[14] W. Feller. *An Introduction to Probability Theory and Its Applications*. Wiley, New York, 1968.

[15] A. Gai, F. Mathieu, F. de Montgolfier, and J. Reynier. Stratification in p2p networks: Application to BitTorrent. In *ICDCS'07*.

[16] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, analysis, and modeling of BitTorrent-like systems. In *ACM IMC'05*.

[17] T. Hoífeld, F. Lehrieder, D. Hock, S. Oechsner, Z. Despotovic, W. Kellerer, and M. Michel. Characterization of BitTorrent swarms and their distribution in the internet. *Elsevier COMNET*, 55(5), 2011.

[18] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. A. Felber, A. Al Hamra, and L. Garces-Erice. Dissecting BitTorrent: Five months in a torrent's lifetime. In *PAM'04*.

[19] T. Karagiannis, P. Rodriguez, and K. Papagiannaki. Should internet service providers fear peer-assisted content distribution? In *ACM IMC'05*.

[20] N. Laoutaris, D. Carra, and P. Michiardi. Uplink allocation beyond choke/unchoke or how to divide and conquer best. In *ACM CoNEXT'08*.

[21] A. Legout, N. Liogkas, E. Kohler, and L. Zhang. Clustering and sharing incentives in BitTorrent systems. In *ACM SIGMETRICS'07*.

[22] A. Legout, G. Urvoy-Keller, and P. Michiardi. Rarest first and choke algorithms are enough. In *ACM IMC'06*.

[23] T. Leighton. Improving performance on the Internet. *Commun. ACM*, 2009.

[24] H-V. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane Nano: Path prediction for Peer-to-Peer applications. In *NSDI'09*.

[25] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane: An information plane for distributed services. In *OSDI*, 2006.

[26] D. Menasche, A. Rocha, B. Li, D. Towsley, and A. Venkataramani. Content availability in swarming systems: Models, measurements and bundling implications. In *ACM CoNEXT'09*.

[27] J. S. Otto, M. A. Sánchez, D. R. Choffnes, F. E. Bustamante, and G. Siganos. On blind mice and the elephant. In *ACM SIGCOMM*, 2011.

[28] M. Piatek, H. V. Madhyastha, J. P. John, A. Krishnamurthy, and T. Anderson. Pitfalls for ISP-friendly P2P design. In *ACM HotNets'09*.

[29] F. Picconi and L. Massoulié. ISP friend or foe? making P2P live streaming ISP-aware. In *IEEE ICDCS'09*.

[30] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, and H.J. Sips. The BitTorrent P2P file-sharing system: Measurements and analysis. In *IPTPS'05*.

[31] G. Siganos, X. Yang, and P. Rodriguez. Apollo: Remotely monitoring the BitTorrent world. Technical report, Telefonica Research, 2009.

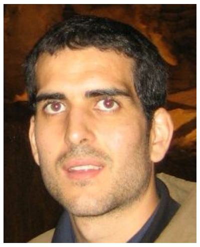
[32] J. J. M. Tan. A necessary and sufficient condition for the existence of a complete stable matching. *J. Algorithms*, 12(1):154–178, 1991.

[33] H. Wang and J. Liu. Exploring peer-to-peer locality in multiple torrent environment. *IEEE TPDS*, 23(7):1216–1226, 2012.

[34] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz. P4P: Provider portal for applications. In *ACM SIGCOMM'08*.

[35] B. Zhang, A. Iosup, J. Pouwelse, D. Epema, and H. Sips. Sampling bias in BitTorrent measurements. *Euro-Par 2010-Parallel Processing*, 2010.

[36] C. Zhang, P. Dhungel, D. Wu, and K. W. Ross. Unraveling the bittorrent ecosystem. *IEEE TPDS*, 22(7).



Ruben Cuevas obtained his MSc and PhD in Telematics Engineering at University Carlos III of Madrid (Spain) in 2007 and 2010, respectively. He is currently Assistant Professor at the Telematics Engineering Department at University Carlos III of Madrid. He has been research intern at Telefonica Research Lab and Courtesy Assistant Professor at University of Oregon in 2008 and 2012, respectively. He has been involved in several national and international research projects in the area of content distribution and green networking. Moreover, he is co-author of more than 30 papers in prestigious international journals and conferences such as IEEE/ACM Transactions on Networking, ACM CoNEXT, IEEE Infocom or IEEE P2P. His main research interests include Content Distribution, P2P Networks, Online Social Networks and Internet Measurements.

Nikolaos Laoutaris is a senior researcher at the Internet research group of Telefonica Research in Barcelona. Prior to joining the Barcelona lab he was a postdoc fellow at Harvard University and a Marie Curie postdoc fellow at Boston University. He got his PhD in computer science from the University of Athens in 2004. His general research interests are on system, algorithmic, and performance evaluation aspects of computer networks and distributed systems. Current projects include: Efficient inter-datacenter bulk transfers, energy-efficient distributed system design, content distribution for long tail content, transparent scaling of social networks, pricing of broadband services and ISP interconnection economics.

Xiaoyuan Yang is a Technology Expert in Telefonica Research, Barcelona. He is currently leading a passionate team to build a new advanced web acceleration solution for both dynamic and static web content. He is also the person that promoted Telefonica Global CDN that is currently deployed in more than 20 countries. His main research interests are related to high-performance computing and distributed systems. Before joining Telefonica Research, he was a teaching assistant in University Complutense of Madrid (UCM) where he worked on high-performance DNA data-mining problems. He got his PhD in computer science from the University Autonoma of Barcelona (UAB).

Georgos Siganos got his Ph.D. at the University of California, Riverside. He currently is with Telefonica Research, Barcelona. His research interests include peer-to-peer systems, distributed systems and social networks.

Pablo Rodriguez is the Internet Scientific Director at Telefonica. He also leads the Internet research group at Telefonica Research (Telefonica I+D) in Barcelona. Prior to joining Telefonica in 2007, he was a Researcher at Microsoft Research, at the Systems and Networking group. While at Microsoft, he developed the Avalanche P2P system and worked in content distribution, wireless systems, network coding, and large-scale complex networks. He was also involved in measurement studies of very popular services, such as the Windows Update system or FolderShare. During his research career he also worked at Bell-Labs, NJ where he developed systems to improve the performance of Wireless applications. Prior to Bell-Labs, he worked as a software architect for various startups in the Silicon Valley including Inktomi (acquired by Yahoo!) and Tahoe Networks (now part of Nokia). He received his Ph.D. from the Swiss Federal Institute of Technology (EPFL, Lausanne) while at Institut Eurcom, (Sophia Antipolis, France). During his Ph.D. he also worked at AT&T Labs (Shannon Laboratory, Florham Park, NJ). Pablo obtained postgraduate studies at EPFL and King's College, London respectively, and an a B.S./M.S. in Telecommunication Engineering at the Universidad Pública de Navarra, Spain.

Symbol	Description
W	number of neighbors for a peer
T	refers to a specific torrent
A	refers to a specific ISP
$\mathcal{T}$	the set of all torrents in a dataset
$\mathcal{A}$	the set of all ISPs with at lest one client in a dataset
$N(v,T)$	number of node v neighbors in torrent T
V(T)	the set of BitTorrent nodes participating in torrent T
V(A,T)	the set of BitTorrent nodes from A participating in torrent T
T(A)	the set of torrents requested by clients of ISP A
A(T)	the set of ISPs that have at least 1 client in torrent T
$c(v)$	number of chunks already downloaded by node v out of the total C chunks that make up a complete file
$U(v)$	node v upload rate
ν	maximum number of allowed remote (inter-ISP) neighbors under a given Locality policy
δ	defines the required ratio between the upload bandwidth of a local and a remote in order to perform a switch under a given Locality policy

TABLE V
Summary of most important symbols used in the paper

Appendix A
Summary of Symbols

Table V presents a summary of the most important symbols used throughout the paper and the supplemental material along with a short description for each of them.

Appendix B
Modelling BitTorrent Traffic Matrices

A. Modeling regular unchokes with a b-matching

The input to a b-matching problem consists of a set of nodes V, and functions $n : V \rightarrow 2^V$, $b : V \rightarrow \mathbb{Z}^+$, and $p : V^2 \rightarrow \mathbb{R}^+$ defined as follows: $n(v)$ defines the set of nodes to which v can be matched with (matching is symmetric here, and thus $u \in n(v)$ iff $v \in n(u)$); $b(v)$ defines the number of parallel matchings that v is allowed to establish; $p(v,u)$ is a measure of the preference that v has for becoming stably matched to u. A solution to a b-matching is a set M of matchings (edges) between pairs of nodes in V, such that for each matched pair $(v,u) \in M$, the matching and capacity constraints n, b are satisfied and further, there exists no "blocking pair" $(v',u') \in M$, *i.e.*, no pair that satisfies: $p(v,v') > p(v,u)$ and $p(v',v) > p(v',u')$.

It is easy to see that there exists a direct mapping from BitTorrent to b-matching [15]. Looking at a particular node v and torrent T: the neighborhood $N(v,T)$ can be mapped to the allowed matchings $n(v)$; the number of parallel unchokes k (default value for k being 4) at each 10 sec interval corresponds to $b(v)$, the number of matchings allowed for v; the uplink capacity $U(v)$ of a BitTorrent client v can be used as a measure of the preference $p(u,v)$ that each node $u \neq v$ would have for being matched with v in the context of a b-matching. b-matchings in which the preference for a node is the same independently of who is considering, i.e., for given u, $p(v,u) = p(v',u)$, $\forall v, v'$, are said to have a global preference function. Tan [32] has shown that the existence of a stable solution for the b-matching problem relates to the non-existence of circles in the preference function p, which is a condition that is certainly satisfied under a global preference function like $U(v)$. Therefore, for the aforementioned mapping from BitTorrent to b-matching, one can use a simple $O(|V(T)| \cdot k)$ greedy algorithm to find the unique stable matching that exists in this case [15].[10]

B. Completion level aware b-matching

In this subsection we extend the basic matching algorithm presented above to allow it to also capture the completion level of a node, i.e., the percentage of a file of total size C that it holds.

1) Edge filtering: Let $c(v)$ denote the number of chunks already downloaded by node v out of the total C chunks that make up a complete file. For a pair of neighbors (v,u) with $c(v) \geq c(u)$ let $I(v \rightarrow u)$, $c(v) - c(u) \leq I(v \rightarrow u) \leq c(v)$ denote the number of chunks of v that "are of interest" to u, *i.e.*, chunks that v has downloaded but u has not. It is easy to see that $I(u \rightarrow v) = c(u) - c(v) + I(v \rightarrow u)$, $0 \leq I(u \rightarrow v) \leq c(u)$. If we assume that the chunks held at some point in time by a node are a random subset of the entire set of chunks, which is reasonable granted LRF [8], then it follows that:

$$\begin{aligned} p_{vu}(x) &= P\{I(v \rightarrow u) = x, I(u \rightarrow v) = c(u) - c(v) + x\} \\ &= HyperGeo(c(u) - x, c(v), C, c(u)) \end{aligned} \tag{3}$$

where $HyperGeo(d,p,s,ss)$ denotes a hyper geometric pmf [14] giving the probability of drawing d "successes" with a sample of size ss from a pool of p items, of which s are "successes". Then the expected amount of interest in the two directions is:

$$\begin{aligned} E\{I(v \rightarrow u)\} &= \sum_{x=c(v)-c(u)}^{c(v)} x \cdot p_{vu}(x) \\ E\{I(u \rightarrow v)\} &= \sum_{x=c(v)-c(u)}^{c(v)} (c(u) - c(v) + x) \cdot p_{vu}(x) \end{aligned} \tag{4}$$

For pair (v,u) we define its *filtering probability* to be:

$$\phi(v,u) = \min\left(\frac{E\{I(v \rightarrow u)\}}{T \cdot U(v) \cdot (\sigma \cdot k)^{-1}}, \frac{E\{I(u \rightarrow v)\}}{T \cdot U(u) \cdot (\sigma \cdot k)^{-1}}, 1\right) \tag{5}$$

where σ is the size of a chunk and T is the duration of an unchoke interval. Given an instance of a b-matching problem $\langle V, n, b, p\rangle$ we filter it to obtain a new one $\langle V, n', b, p\rangle$ in which we keep an edge (v,u), meaning that $v \in n(u)$, $u \in n(v)$ and $v \in n'(u)$, $u \in n'(v)$, with probability $\phi(v,u)$, whereas we drop it with probability $1 - \phi(v,u)$.

2) Time-evolving completion ratios: Let $c_t(v)$ be the completion ratio of node v at time t and let M_t be the stable matching obtained from solving the b-matching $\langle V, n', b, p\rangle$ in which n' is obtained from n after applying the filtering procedure described above with completion ratios $\{c_t(v) : v \in V\}$.

---

[10] Uniqueness is guaranteed under the assumption that there are not ties in speeds. We made sure that his is the case by adding to our speed datasets a very small random noise.

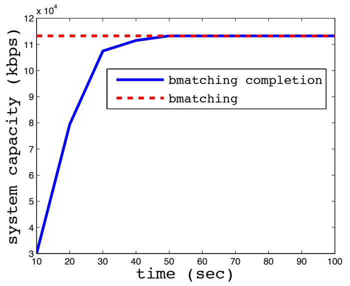


Fig. 7. Aggregate system capacity from baseline b-matching and b-matching with completion levels. Parameters: $|V| = 40$, uplink rates randomly distributed with mean 2 Mbit/s, $C = 10000$, $c_0(v) < 1\%$, $\forall v$, unchoke duration=10 sec, chunk size=32 kBytes. The more complex version converges within a minute to the steady-state value predicted by the baseline b-matching. Download completion requires around 30 $mins$.

Then the completion ratios of nodes can be updated at the end of the unchoke interval as follows:

$$c_{t+T}(v) = c_t(v) + \sum_{u:(v,u)\in M_t} \min\left(E\{I(u \to v)\}, \frac{T \cdot U(u)}{\sigma \cdot k}\right) \tag{6}$$

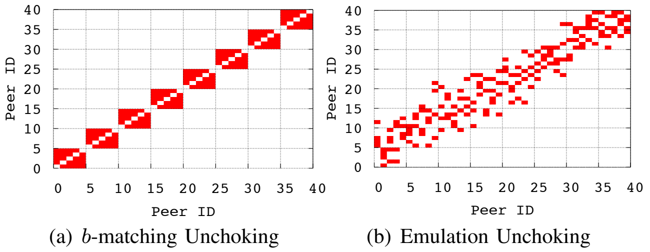
Thus with the above we have a method for mapping the effects of completion levels on the state of a torrent and consequently on the resulting matchings.

C. *Validation of modeling*

In this section we validate the accuracy of modeling the unchoke algorithm using a b-matching. We look at the two typical phases of a torrent's lifetime [30].

1) Startup phase: During the initial phase of a new torrent leechers hold few chunks and thus whether two nodes unchoke each other depends, beyond their speeds, on the set of chunks they hold. The b-matching modeling of unchoke described in Appendix B-A assumes that steady-state has been reached and thus chunk (in)availability does not affect the resulting matchings. In Appendix B-B we have extended this basic matching algorithm to allow it to also capture the *completion level* $c(v)$ of a node. We have used this completion-level aware b-matching in conjunction with small initial completion levels $c(v) < 1\%$ for all leechers to estimate the effect chunk availability has on the aggregate capacity of a torrent. BitTorrent's LRF chunk selection strategy is used for expediting the convergence to steady state. We verified this by comparing our two implementations of b-matching. In Fig. 7 we give an indicative example to show that for everything but very small files, the startup phase is much shorter than steady state. For this reason we can ignore it at small cost in terms of accuracy and focus on the baseline b-matching that is more scalable to large datasets than the more complicated one involving completion levels.

2) Steady state: Next we validate the extent at which the steady state matchings predicted by b-matching resemble the unchoking behavior of an actual mainline client of BT (v.3.4.2). More specifically, we set-up a carefully built torrent with 40 clients[11] , the slowest of which, was given an uplink capacity of 80 kbit/s, whereas successive ones were made increasingly faster using a step of 24 kbit/s. We chose such a small increment to recreate a rather difficult environment for stratification [21] to arise. For each client, we measured the frequency at which the client unchokes each other client, and then we put a mark on Fig. 8(b) for the k clients that it unchokes more frequently (client ids on the figure are assigned according to upload bandwidth in increasing order). Comparing with Fig. 8(a) that depicts the same information from the execution of b-matching under the same uplink capacities, it is easy to see that b-matching provides a reasonable prediction of the real unchokes that take place and therefore can be used as a scalable tool for processing huge numbers of small and large torrents[12] that would otherwise be impossible to simulate concurrently. We got similar accuracy using many other torrent sizes and uplink distributions, including empirical ones from measurement.



(a) b-matching Unchoking (b) Emulation Unchoking

Fig. 8. Unchoking Patterns

APPENDIX C
A FAMILY OF LOCALITY-BIASED OVERLAYS

In this appendix we present an extended family of overlay construction algorithms that we refer to as Locality(δ, μ). Its operation is as follows. It starts with a neighborhood $N(v,T)$ of $\max(W, |V(T)|-1)$ randomly selected neighbors which are then filtered based on speed comparisons against the set of local nodes $V(A,T)\backslash\{v\}$. These comparisons are modulated by the parameters δ, μ as follows. Parameter μ controls the maximum number of allowed remote (inter-ISP) neighbors in $N(v,T)$. If the number of remote nodes in $N(v,T)$ is greater than μ then a remote node u is substituted by a local w that is not already in the neighborhood until the number of remotes reaches μ. If there are no more local nodes for performing switches then u is taken out of the neighborhood. If the number of remotes in $N(v,T)$ is already below μ, then u is substituted by a not already taken local node w only if $1 - \frac{U(w)}{U(u)} < \delta$.

The overlay construction algorithms defined in Section VI are members of this family with the values of δ and μ detailed in Table VI.

Overlay	δ	μ
LOIF	0	$\min(W, \lvert V(T)\rvert - 1)$
Locality	1	$\min(W, \lvert V(T)\rvert - 1)$
Strict	1	1

TABLE VI
δ AND μ VALUES FOR LOIF, LOCALITY AND STRICT OVERLAY CONSTRUCTION ALGORITHMS.

---

[11] Note that around 80% of the torrents in our `mn40K` dataset have ≤ 40 peers.

[12] As a reference, in this paper we use this methodology to process more than 40k different torrents across our different datasets.

Number of Samples	Percentage of countries
0 - 355	16.7%
356 - 1302	16.7%
1303 - 4595	16.7%
4596 - 18866	16.7%
18867 - 119689	16.7%
119690 - 5706752	16.7%

TABLE VII
DISTRIBUTION OF NUMBER OF SAMPLES (INDIVIDUAL USERS) PER COUNTRY FOR OOAKLA SPEED DATASET

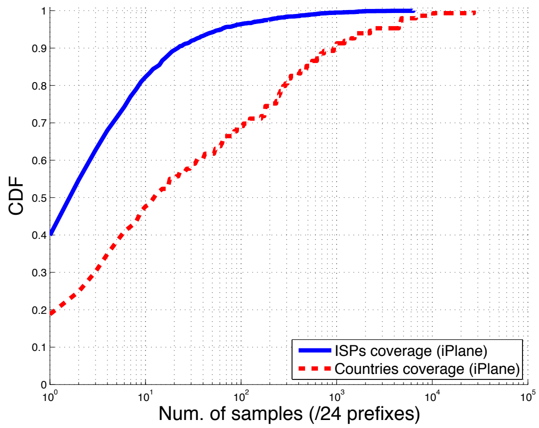


Fig. 9. Distribution of number of samples (/24 prefixes) per ISP and Country for iPlane dataset

APPENDIX D
FURTHER CONSIDERATIONS AND LIMITATIONS OF DATASETS

In this appendix we present a detailed discussion on some limitations of the used datasets. We separately discuss our speed and demographic datasets.

A. *Speed Datasets*

1) Representativeness of speeds datasets: We first characterize the representativeness of our two speed datasets, Ookla [2] and iPlane [25]:

- **Ookla** provides (as described in Section IV) the median upload and download speed for users located in 215 different countries obtained from a speed test service. Table VII summarizes the coverage of the dataset for different countries as reported in [2]. Just 16.7% of the countries present less than 350 samples.
- **iPlane** provides the access speed for around 87k /24 IP prefixes. Specifically, it offers the median access speed for each one of these prefixes. Unfortunately, the number of samples used to calculate that median value is not reported. iPlane speed dataset covers just 3248 ISPs (around 30%) and 149 countries from our `mn40k` dataset. Fig. 9 presents the distribution of the number /24 IP prefixes covered by iPlane for each country and ISP within our dataset. We observe that iPlane provides information for $\leq$ 10 /24 prefixes for 50% and 80% of countries and ISPs within `mn40k`, respectively.

Although iPlane offers a significantly higher granularity than Oakla, it provides a rather low coverage for some countries and ISPs that may lead to a lack of the required statistical representativeness in those cases. Hence, in this paper we have used Ookla as our main dataset and have leaved iPlane for validation purposes.

2) Effect of speed limitations imposed by users: It is well-known that some users enforce constrains in the speed of their BitTorrent software, for instance to avoid performance degradation of other applications. However, characterizing or modeling this phenomenon is a complex task that to the best of the authors' knowledge has not been addressed so far. Therefore, it is uncertain the fraction of users that use these techniques and the speed limitations that they impose.

In any case, the analysis conducted in the paper allows to discuss the expected results if user-enforced rate limiting would be an extended practice. In that case, the actual speed distribution for different ISPs would follow a step function in which the access speed of most users would be concentrated in few common steps (256KB, 512KB, 1MB, etc). This would produce a densification of the speed spectrum in those common speeds. Our analyses of the bounds in the number of local unchokes and the Inherent Localizability in Sec. IV suggest that this densification would lead Random and Locality to perform worse than what we observed in the results presented in the paper. However, the degradation in the performance of the Random overlay construction policy would be higher than for Locality polices and then, the relative gain obtained by Locality overlay construction algorithms would be higher than the one reported in the paper.

B. *Demographic Datasets*

The demographic datasets used along the paper were collected in 2009 from the two most popular portals at the time of the measurement study, namely Mininova and The Pirate Bay. In this subsection we discuss the implications that using a relatively old dataset may have in the obtained results.

1) Portals' Popularity: One of the portals that we use in our study, The Pirate Bay, has kept (and even increased) its popularity since our data collection campaign and is currently the most popular BitTorrent portal. However, Mininova removed all the copyrighted material after a court sentence at the end of 2009 and its popularity rapidly decreased. Hence, the fact that the main portal used in our analysis is not popular anymore might impact the obtained results.

Portals are just entities where torrent files are indexed and typically there is an important overlapping between content indexed across major portals [36]. This overlapping is especially high for popular and mid-popular torrents which host a major fraction of users. This suggests that, as far as the portal used to collect torrent files is popular, the set of torrents that we would use as input to our experiments is going to be similar and then the impact of the specific used portal is limited.

2) BitTorrent's demographics: The demographics of BitTorrent swarms may have evolved since our data collection campaign impacting the results presented in the paper. For instance, BitTorrent popularity may have significantly decreased making locality techniques less necessary or the popularity of BitTorrent across different ISPs may have changed leading to a substantial variation in the potential transit traffic savings for those ISPs. Next we present an elaborated discussion on the evolution of the demographics of BitTorrent based on real data collected between 2010 and 2012 in different works [9], [27], [13].

Overall BitTorrent popularity: A recent study conducted by J. Otto et al.[27] shows that the overall BitTorrent traffic has slightly increased in the last years. Furthermore, to the

(a) Countries

mn40k	2010	2011	2012
1	1	1	1
2	4	3	2
3	7	4	4
4	2	11	10
5	9	13	12
6	14	9	7
7	3	10	11
8	6	6	18
9	5	2	3
10	8	8	9
11	12	5	5
12	16	24	23
13	19	15	15
14	13	20	21
15	18	21	24
16	10	7	6
17	22	28	28
18	28	19	19
19	20	22	22
20	39	18	26
21	17	27	20
22	41	14	8
23	15	17	16
24	11	12	14
25	24	29	33

(b) ISPs

mn40k	2010	2011	2012
1	3	1	1
2	6	3	3
3	2	17	14
4	5	4	5
5	18	10	10
6	20	9	9
7	15	6	6
8	563	149	78
9	7	23	25
10	26	34	26
11	16	7	7
12	1	13	16
13	4	2	2
14	46	39	34
15	44	11	13
16	566	1756	2090
17	32	16	12
18	29	38	35
19	23	27	61
20	105	120	129
21	45	22	19
22	33	36	38
23	41	52	47
24	11	14	20
26	50	41	41

TABLE VIII
RANK OF THE TOP 25 ISPS AND COUNTRIES WITH THE LARGEST NUMBER OF USERS IN `mn40k` IN OUR DATASETS FROM MAY 2010, NOV. 2011 AND JAN 2012

best of the authors knowledge no ISP has implemented yet Locality techniques, thus, nowadays BitTorrent swarms are random overlays. In summary, because BitTorrent traffic is still very significant and swarms are formed following a Random overlay construction policy, Locality techniques are necessary and the results presented in this paper still useful.

Evolution of BitTorrent Popularity across Countries and ISPs: In separate works [13], [9] we have collected datasets that include snapshots of tens of thousands BitTorrent swarms. These datasets were collected in May 2010, Nov. 2011 and Jan. 2012. Although those datasets are different in nature from the ones presented in this paper (*i.e.*, in those studies our objective was collecting data from thousand of BitTorrent swarms over a period of few weeks rather than a snapshot over a couple of hours) they are still valid to estimate the demographics of BitTorrent across countries and ISPs.

Table VIII shows the rank of the Top 25 countries and ISPs in number of users from `mn40k` in our more recent datasets from May 2010, Nov. 2011 and Jan. 2012, respectively[13]. First, we observe that 22 countries remain in the Top 25 across all the datasets, although we observe some variations in the position of individual countries. As expected, at the ISP level we observe a higher variability: 13 ISPs stay in the Top 25 across all datasets whereas other 9 lose some popularity but remain in the Top 50. Just 3 ISPs suffer from a major change in their demographics that leads to a significant drop in their ranks. These results suggest that although there have been modifications in the popularity of ISPs and countries that may affect their ability to localize traffic, these variations are (in general) moderate and thus the impact on the presented results is expected to be equally moderate. Note that results are likely to vary significantly for those few ISPs that present an important change in their popularity.

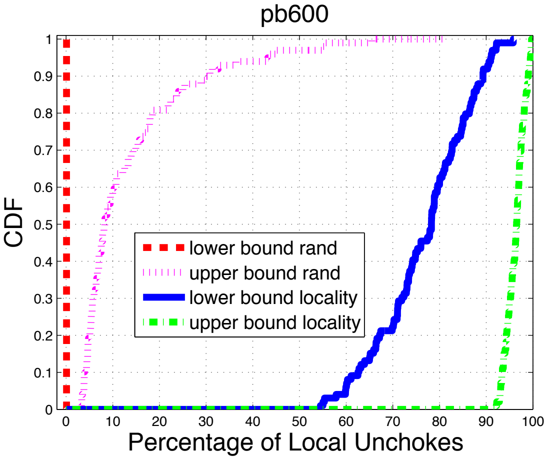


Fig. 10. Speed agnostic bounds for Top-100 ISPs in our `pb600` dataset.

In summary, the results presented in the paper are likely to be a good approximation in the current Internet. However, we acknowledge that some deviation, proportional to the variation in the popularity of BitTorrent in different countries and ISPs, may exist.

APPENDIX E
ADDITIONAL PERFORMANCE ANALYSIS RESULTS

In this Appendix we present additional results obtained with the different methodologies described in the paper, namely speed agnostic bounds (Section IV-C), Inherent Localizability (Section IV-E) and BitTorrent traffic matrices accurate estimation (Section VII).

A. Additional Results for Speed Agnostic Bounds

1) Locality gains in Dense vs Sparse mode: Overall Random localizes sufficiently in sparse mode as long as it can get a small number of local nodes in each neighborhood. In dense mode things become more challenging as it no longer suffices to guarantee a small threshold of locals but instead Random has to have a strong majority of locals in each neighborhood. In both modes, Locality has to satisfy easier conditions to localize the same number of unchokes. Further, we can actually prove that the improvement factor of Locality over Random in terms of the number of localized unchokes is higher in dense mode than in sparse mode. We consider only the case with $|V(A,T)|-1 \geq k$ and $|V(T)|-1 \geq W$ (the other ones can be worked out similarly). Based on the previous analysis we get that the expected improvement factor in sparse mode is:

$$\frac{k}{W \cdot \frac{|V(A,T)|-1}{|V(T)|-1}} \tag{7}$$

In dense mode for $|V(A,T)|-1 \geq W$ the improvement factor is:

$$\frac{k}{k \cdot \frac{|V(A,T)|-1}{|V(T)|-1}}$$

which is greater than Eq. (7) because $W > k$. For $|V(A,T)|-1 < W$ the improvement factor is:

$$\frac{k \cdot \frac{|V(A,T)|-1}{W}}{k \cdot \frac{|V(A,T)|-1}{|V(T)|-1}} = \frac{|V(T)|-1}{W}$$

which can be checked to be greater than Eq. (7) for the case with $|V(A,T)|-1 \geq k$.

---

[13] Note that similar results have been obtained for the Top 50 countries and Top 100 ISPs. We limit the discussion to the Top 25 for the shake of clarity. Moreover, we anonymize the ISPs identity and just refer to their rankings.

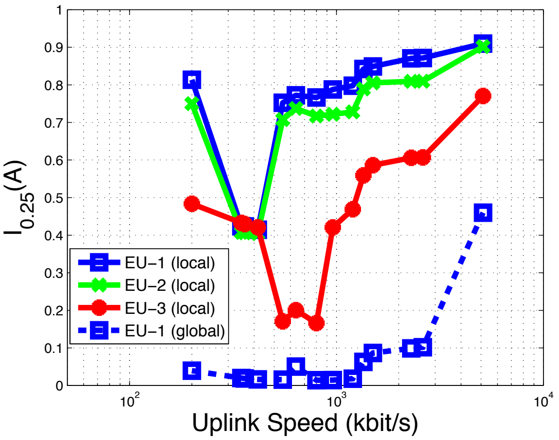


Fig. 11. The inherent localizability of 3 European ISPs based on their 10 most popular local torrents and the 10 most popular torrents across the entire dataset.

ISP	Prct 5	Prct 25	Prct 50	Prct 75	Prct 95
US1 (LOIF)	-3.98%	-7.93%	-6.71%	-5.61%	-5.87%
EU1 (LOIF)	1.53%	-2.01%	-1.47%	-2.51%	-2.50
US1 (Locality)	-2.45%	-3.38%	-1.32%	1.20%	3.46%
EU1 (Locality)	3.83%	6.36%	3.33%	2.72%	6.69%
US1 (Strict)	24.87%	-1.74%	2.88%	6.81%	16.81%
EU1 (Strict)	76.73%	16.07%	18.59%	18.00%	29.04%

TABLE IX
QoS DEGRADATION UNDER `mn40K` AND OOKLA SPEEDS.

ISP	LOIF	Locality	Strict
US1	**34.03%**	**77.86%**	**99.10%**
US2	30.56%	69.20%	98.73%
US3	37.11%	78.70%	99.27%
EU1	**15.25%**	**72.80%**	**99.35%**
EU2	21.22%	72.26%	99.18%
EU3	26.57%	71.92%	99.05%

TABLE X
TRANSIT TRAFFIC REDUCTION UNDER `pb600` AND OOKLA SPEEDS

2) Speed agnostic bounds for big torrents: In Fig. 10 we recompute the upper and lower bounds for localized unchokes for Random and Locality for the top-100 ISPs based on the `pb600` dataset. In this case, the upper bound of Random is lower (6.46%) because nodes from the same ISP become an even smaller minority in very large torrents. On the other hand, Locality benefits in terms of both upper and lower bounds. This happens because the bounds for Locality, unlike Random, depend on the absolute rather than the relative number of local nodes, which increases with larger torrents.

B. Analyzing the effect of speed and demographics in local and global torrents using the Inherent Localizability

It seems very difficult to devise simple rules of thumb for predicting how $I_q(A)$ will change with $U(A)$ without using detailed demographic and speed information as we did in Section IV-E. The complication owes to the torrent mix of each ISP, which includes both *global* and *local* torrents. Global torrents are those very popular torrents consumed by users around the world. Global torrents are omnipresent in the entire speed range, but as the country speed cdf sparsifies at higher ranges (Fig. 4), fewer of them will be encountered as remote neighbors when an ISP upgrades to such speeds. This leads to more internal unchokes of global torrents, effectively making the inherent localizibility of global torrents a monotonic function of speed.

Local torrents exist at specific ISPs and speed ranges and thus their behavior during speed upgrades is more difficult to predict. For example, an ISP at a French speaking African country will see its unchokes to and from remote ISPs increasing if it upgrades its residential accesses to speeds that bring it near the offered speeds in France, Belgium, and Canada. If it upgrades to even faster speeds though, its remote unchokes will fall because its local users would rather unchoke each other than peers in other countries. In Fig. 5 the localizability of US1 fell at around 1Mbit/s because it entered the region of many other US ISPs and thus started exchanging unchokes with them for content that although in English, is local to US (local TV, music, *etc.*). In Fig. 11 we compute the inherent localizability of the 10 most popular local torrents in 3 European countries and the corresponding 10 most popular across the entire dataset[14] . The global torrents change monotonically whereas local ones do not.

C. Additional results on Transit Traffic Reduction and Users QoS Degradation

1) Additional details on degradation of QoS under `mn40k` and Ookla speeds: In Table II(b) we reported the degradation of QoS in terms of median download rate under the different policies for the six largest ISPs. In this Appendix, we focus on the two largest ones, US1 and EU1, and we compare the QoS degradation in terms of additional percentiles (5%, 25%, 50%, 75%, 95%). Looking at Table IX we see that the reduction in terms of median is not much different than the reduction in terms of all other, but the two extreme, percentiles. The biggest difference is observed under Strict and is due to the heavy penalty paid by users downloading unlocalizable torrents, as explained in Sect. VII-B2. Notice that all observed differences are due to speed differences between ISPs. Differences between access speed within the same ISP do not play a role here because under Ookla all clients of an ISP are assigned their country speed.

2) Stability of results across demographic datasets: In Table X we recompute the transit traffic savings compared to Random based on the dataset `pb600` of the 600 most popular torrents of PirateBay and Ookla speeds. In this case the savings are even higher. There are two reasons for this. First, Locality and LOIF have more internal nodes to use. Second, Random is impacted negatively because despite the increase in absolute number of locals, their relative percentage compared to the entire population shrinks.

3) Stability of results across speed datasets: In this Appendix we analyze the transit traffic reduction and the degradation of the median QoS for our 6 major ISPs under `mn40K` and iPlane speeds [25] . Table XI shows the obtained results.

First, we observe that the overall trends are aligned with those obtained with Ookla speeds (See Table II): (i) the more strict the locality policy is the higher the transit traffic reduction is at the cost of a higher median QoS degradation; (ii) LOIF

---

[14] The results for EU2 and EU3 for the 10 most popular torrents across the entire datasets are similar to EU1, thus, for the shake of clarity, the figure only presents the results for EU1.

(a) Transit Traffic Reduction under

ISP	LOIF	Locality	Strict
US1	**16.14%**	**52.12%**	**96.63%**
US2	8.77%	46.73%	95.68%
US3	9.18%	39.55%	94.66%
EU1	**3.94%**	**43.89%**	**94.92%**
EU2	5.68%	50.89%	94.69%
EU3	12.68%	41.63%	95.62%

(b) Degradation of median QoS

ISP	LOIF	Locality	Strict
US1	**-0.23%**	**30.29%**	**50.13%**
US2	-0.15%	38.69%	59.18%
US3	-0.13%	42.34%	64.39%
EU1	**-5.45%**	**23.61%**	**45.25%**
EU2	-0.02%	29.08%	51.22%
EU3	-4.94%	16.56%	29.41%

TABLE XI
RESULTS FOR ISPS EU1-EU3, US1-US3, FOR `mn40k` AND IPLANE SPEEDS.

is by definition a win-win policy; (iii) the Transit Traffic reduction for Locality and Strict is perfectly aligned for both speed datasets.

However, there are also some notable differences. On the one hand, LOIF offers higher transit traffic reduction under Ookla speeds. On the other and, Locality and Strict present higher median QoS degradation under iPlane speeds. The reason for these discrepancies is the following: In Fig. 4 we computed the CDF of the median speed for the different ISPs within our `mn40k` dataset under Ookla and iPlane. Furthermore, we annotated the median speeds associated to the 6 major ISPs under study. We see that all these ISPs are located in a lower percentile of the distribution for the iPlane curve. Then, for LOIF the clients within these ISPs would perform a lower number of switches because under iPlane remote nodes are typically faster than local nodes. This leads to reduce the observed transit traffic saving, but does not affect to the users QoS because they do not lose fast peers. In the case of Locality and Strict, switches between remote and local peers are performed regardless of their speed, then transit traffic reduction is similar regardless of the used speed dataset. However, these switches are more harmful for users QoS under iPlane speeds where remote nodes are typically faster than local ones.

4) Stability of results across time: Next we evaluate the effect of time on our obtained transit savings. We look at hourly and weekly time scales.

a) Hours: Torrent membership is dynamic and thus if examined at different hours of the day a torrent will have different participants. In this section we evaluate the impact of dynamic membership on our results. We do so using `pb600`. As we can crawl this dataset in less than an hour, we performed 24 crawls of it across the different hours of a day. In Fig. 12 we plot the resulting transit savings for ISP US1 at different hours. One can see that dynamic membership has a very small impact on the calculated transit savings. As `pb600` is biased towards large torrents, we wanted to evaluate also a less biased dataset and for this purpose we used `mn3K`. Again, Fig. 12 shows that our computations are quite stable.

The depicted stability should not come as a surprise. First, transit traffic reduction is a relative metric so it is less sensitive to diurnal variations of torrent population than, *e.g.*, the absolute amount of transit traffic. Second, the aggregation effect from multiple independent torrents naturally smooths out any variability at specific hours.

b) Weeks: Next we recompute the transit savings for all 6 ISPs on different days. We do so by using 3 snapshots of `mn40K` taken from 3 consecutive weeks. In each case the dataset contains the 40K *latest* Mininova torrents and thus apart from membership differences on the same torrent, the snapshots also differ in terms of included torrents: the second one, *e.g.*, contains the new torrents of the latest week, which are not included in the previous one. The transit savings differ by less than 7% in all cases.

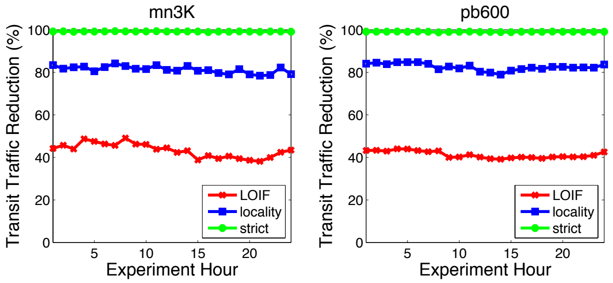


Fig. 12. Transit traffic reduction for US1 during different times of a day under Ookla speeds.

In summary, the experiments conducted in this subsection lead to the following conclusions: First, our methodology is robust because it provide meaningful results for all the considered demographic and speed datasets. Second, the obtained results are stable, and thus valid, across time. Finally, the use of different speed datasets leads to different quantitative results but the main observed trends remain the same.