The hidden locality in swarms

John S. Otto and Fabián E. Bustamante
Northwestern University
{jotto,fabianb}@eecs.northwestern.edu

***Abstract*—People use P2P systems such as BitTorrent to share an unprecedented variety and amount of content with others around the world. The random connection pattern used by BitTorrent has been shown to result in reduced performance for users and costly cross-ISP traffic. Although several client-side systems have been proposed to improve the locality of BitTorrent traffic, their effectiveness is limited by the availability of local peers.**

We show that sufficient locality is present in swarms – if one looks at the right time. We find that 50% of ISPs have at least five local peers online during the ISP's peak hour, typically in the evening, compared to only 20% of ISPs during the median hour. To better discover these local peers, we show how to increase the overall peer discovery rate by over two orders of magnitude using client-side techniques: leveraging additional trackers, requesting more peers per sample, and sampling more frequently. We propose an approach to predict future availability of local peers based on observed diurnal patterns. This approach enables peers to selectively apply these techniques to minimize undue load on trackers.

I. Introduction

Peer-to-peer (P2P) file sharing systems remain popular worldwide and generate a large fraction of total network traffic. BitTorrent [1], the most popular P2P file sharing system, has at least 165 M users spanning nearly every country in the world. Between them, these users share 15 M different files[1], contributing nearly a quarter of all fixed network traffic around the world [2]. While this traffic is primarily generated by end users, several businesses have adopted BitTorrent including Facebook,[2] Twitter,[3] and eBay.[4]

Clients (*peers*) downloading a file in BitTorrent comprise a *swarm*. Within a swarm, peers connect mostly at random, oblivious to the underlying network topology. This random pattern of connections complicates network management and increases the cross-ISP traffic cost of Internet, and potential impact the performance of end users [3], [4].

Several approaches have been proposed to improve the locality of BitTorrent traffic by modifying how peers find and connect to other peers [3], [5]–[7]. While previous studies have found locality in swarms [8], others have questioned the effectiveness of client-based approaches [9] due to the challenges of discovering local peers.

We overcome the challenges of local peer discovery by leveraging diurnal patterns and applying client-side techniques to improve overall peer discovery.

Through an analysis of swarm population dynamics, we show that locality is present in swarms – if one looks at the right time. For popular content swarms, 50% of ISPs seen in the swarm have at least five local peers online during the ISP's peak hour. During an ISP's peak hour, the relative fraction of local peers – and therefore the local peer discovery rate – is typically 50% higher than the daily average.

We evaluate client-side techniques that boost the peer discovery rate *by two orders of magnitude*, enabling peers to quickly discover online local peers. To achieve this, we leverage additional trackers beyond those listed in a .torrent file, request larger samples of peers from trackers, and increase the rate of requesting samples.

To balance the goals of local peer discovery and minimizing load on trackers, we propose an approach to identify the time of day at which the greatest number of local peers can be found. This enables peers to time-shift a download to maximize locality or to determine when additional probing is unlikely to yield more local peers. This information enables peers to strategically decide when – and when not – to do additional probing, thereby maximizing local peer discovery while preventing undue load on trackers.

The rest of this paper is organized as follows. In Sec. II we discuss BitTorrent's peer discovery mechanisms in greater detail. In Secs. III and IV we describe our methodology and identify patterns of available locality in swarms. We evaluate our techniques for increasing the rate of tracker-based peer discovery in Sec. V. After discussing related work in Sec. VI, we state our conclusions in Sec. VII.

II. Background

In BitTorrent, each piece of content in the system is called a "torrent" and is uniquely identified by its "info_hash" value. The content is split into pieces, which are disseminated among peers participating in that torrent.

BitTorrent has three ways for peers to learn about others in the swarm: queries to centralized "tracker" servers, Distributed Hash Table (DHT) lookups, and Peer EXchange protocols (PEX).

Centralized trackers maintain lists of peers active in a torrent and provide samples of the swarm population to peers on request. Trackers continue to be the predominant approach for peer discovery; Varvello and Steiner report that 59-66% of BitTorrent peers use them [6].

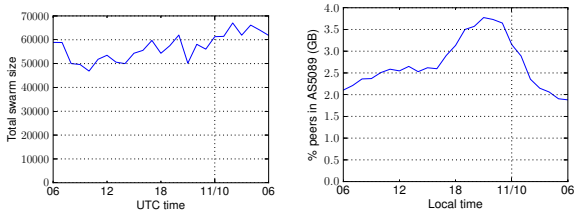
[1]According to meta-search engine torrentz.eu, the most popular BitTorrent meta-search engine as ranked by alexa.com.

[2]http://torrentfreak.com/facebook-uses-bittorrent-and-they-love-it-100625/

[3]http://www.datacenterknowledge.com/archives/2010/02/10/twitter-using-bittorrent-to-speed-servers/

[4]http://www.ebaytechblog.com/2012/01/31/bittorrent-for-package-distribution-in-the-enterprise/

978-1-4799-0521-8/13/$31.00 ©2013 IEEE



(a) Total swarm size (b) Swarm % AS5089 (GB)

Fig. 1. Time-of-day patterns in a globally-popular swarm over a 24-hour period. Total swarm size (left) does not show a diurnal pattern because the swarm includes peers from many time zones. For a single ISP (right), its percent of peers in the swarm follows a diurnal pattern peaking in the evening: this is when local peer discovery rate is the highest.

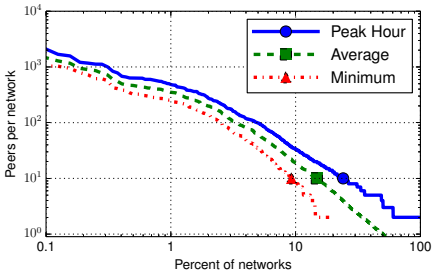


Fig. 2. Distributions of the minimum, average and maximum ("peak hour") number of peers seen in each ISP network. 50% of ISPs have at least 5 peers online during the peak hour. We find a 2-to-3-fold variation in available locality over the day (distance between minimum and peak hour curves).

Peer EXchange protocols (PEX) and Distributed Hash Table-based (DHT) peer discovery [10] were developed to reduce BitTorrent's reliance on trackers. Gossipping PEX protocols enable peers to exchange swarm population information directly. For DHT, peers build a decentralized overlay network to maintain swarm population information. However, these additional peer discovery mechanisms are not always available. In private torrent networks – BitTorrent darknets – the PEX and DHT peer discovery protocols are disabled; Zhang et al. found that these networks are used by 24 M users to share 4.4 M torrents – equivalent in size to the entire public English-speaking BitTorrent world [11]. Finally, firewalls or carrier-grade network address translation (CGN) may prevent users from utilizing DHT peer discovery.

Given the predominance of tracker use in BitTorrent, we focus on it for the remainder of this work.

Trackers maintain the set of peers online in each content swarm and upon request provide random samples of a swarm population so that a peer can request the content from other peers in the swarm. The tracker's interface to peers is a single request called an "announce"; the peer implicitly tells the tracker that it is interested in a piece of content and that the client is accepting connections from other peers on the given IP address and port. In return, the tracker responds with a set of other peers active in the swarm. We have found that the default sample size is typically 50 peers, though peers may request a larger sample. Tracker behavior varies with respect to maximum sample size and minimum interval between "announce" requests.

A key aspect of tracker functionality is that the peers given in the "announce" response are *randomly sampled* from the population. While there may be local peers in the swarm, the probability of finding them is low when the swarm is large or there are only a few local peers.

III. Swarm Sampling Methodology

To evaluate the availability and temporal dynamics of "local peers" in BitTorrent swarms, we need information about swarm composition at fine time granularity. We use repeated probes to each tracker to reconstruct the swarm population because we do not have direct access to all trackers' data structures. We study several of the most popular torrents, ranked in terms of swarm size on torrentz.eu and query all their listed trackers to obtain a complete view of the swarm.

We collect our dataset of sample swarm populations over a 17-day period, leveraging thousands of users around the world running the Dasu [12] plugin for the Vuze [13] BitTorrent client. Each client coordinates with a central server and queries trackers to obtain swarm samples. The clients obtaining the samples do not accept connections from remote peers regarding the queried torrent and no content is exchanged as a result of our experiments. Since querying the tracker is an implicit signal of activity in the swarm, the Dasu clients effectively join the swarm. We eliminate this potential bias by filtering out the IP addresses belonging to Dasu clients that we used to collect the dataset. At worst, we *under-report* available locality in the ISPs where our monitoring peers are located.

For each torrent, we aggregate swarm samples at one-hour granularity. Since this interval is shorter than the median BitTorrent session duration of approximately 4 hours [4], we assume that peers seen in a given hour are active in the system and that churn in the swarm population does not affect our results. We map each peer seen in our samples to its country based on the ISP advertising its IP address. In total, we see peers in 177 countries and 3076 networks.

In the following section we analyze our dataset collected using this methodology to estimate and predict local peer availability in swarms.

IV. Maximizing Local Peer Discovery

We study the temporal dynamics of swarm populations to measure available locality and how it varies throughout the day. The availability of local peers defines an upper bound on the potential benefits of biased neighbor approaches to improve the locality of BitTorrent traffic. Based on our analysis, we describe an approach to predict the best time to find locality.

Since swarms of globally-popular torrents are comprised of users around the world, total swarm size does not exhibit a clear diurnal pattern. Figure 1a plots the aggregate swarm size for a large swarm over a day. The relatively constant swarm size masks the underlying dynamics of the swarm.

When we consider a single ISP's percentage of peers in the swarm, we observe a clear diurnal pattern. Figure 1b plots the change in the relative fraction of a single ISP's peers in the swarm over the course of a day. We observe a similar diurnal pattern with a peak in the evening hours for many ISPs in the dataset.

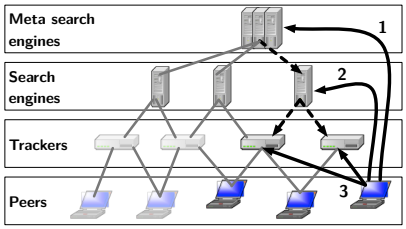


Fig. 3. Diagram of relationships between components and example interaction to download content. Users (1) select a torrent using a meta-search site, (2) download the torrent from a search site, and (3) the BitTorrent client contacts the trackers obtained in step 2. If not all trackers are received in step 2 (e.g. the left-most trackers), then the user will not be able to access the peers known only to those missing trackers (grayed out).

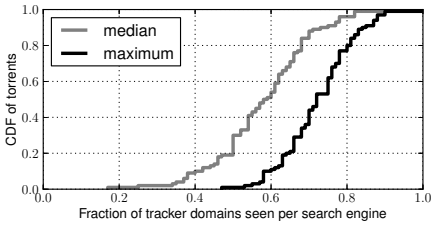


Fig. 4. Distribution of the median and maximum fraction of tracker domains included in .torrent files downloaded from different search engines. In the median case, users will only use 58% of the available trackers when downloading a torrent, missing over a third of the potential swarm population. No search site includes all trackers for a torrent.

A. Potential increase in local peer availability

We show that these diurnal patterns are general for ISPs in our dataset. In Fig. 2 we plot the minimum, average, and "peak hour" number of peers seen over the course of a day across ISPs. For instance, the largest network in the swarm (at the far left in the plot) has 1000 online peers at the minimum hour and 2000 peers during the peak hour.

The distance between the minimum and peak-hour curves shows how much the peer population changes based on time of day; for the top 1% of networks, we see that populations roughly double from the minimum to the peak-hour value. For smaller networks, the space between the curves is larger, indicating greater variation in their swarm size. For example, at the 90th percentile there is a 3-fold increase in online peers, from a minimum of < 10 to a peak of about 30 peers.

This 2-to-3-fold increase in the number of online peers directly translates to increased *local peer discovery* rates during the peak hour. Therefore, downloading during this peak hour increases the effectiveness of biased neighbor approaches that improve download performance and reduce cross-ISP traffic.

B. Predicting future local peer availability

We present an approach that enables peers to determine the best time to discover local peers. Our approach leverages the fact that the number of peers online in each network follows a diurnal pattern. Given a model for the diurnal trends of online users in each network, we can extrapolate *the future distribution* of a swarm's population from as little as a single sample from the swarm. To accomplish this, we build an empirical model of the diurnal trends for each network.

The best time to discover local peers is during the ISP's peak hour – when the relative fraction of peers is highest. We start with the existing swarm distribution (number of peers per ISP) and use the empirical model of each ISP's diurnal patterns to predict the number of peers online in each ISP for the next 24 hours. Then, for each hour, we estimate the relative fraction of peers in the target ISP by dividing the predicted value for the target ISP by the total predicted population size for that hour. Given this, we select the hour with the highest predicted fraction of peers.

This approach for predicting when local peers are online is useful for determining the probability of finding local peers, which informs whether the additional probing approaches we evaluate in the next section are likely to yield more local peers.

V. Maximizing tracker-based peer discovery

In this section, we evaluate two approaches to improving a peer's knowledge about the swarm beyond the default behavior: leveraging all available trackers and maximizing the peer discovery rate for each tracker. We begin by examining the relationships between components involved in tracker-based peer discovery to see how users can better utilize all available resources.

BitTorrent users interact with several components to download content from other peers, which are diagrammed in Fig. 3: trackers that coordinate peers, search sites that maintain indices of content, and meta-search engines that index the search sites themselves. Thin gray lines indicate relationships between these components. Solid lines show relationships established by the user and dashed lines indicate the possible relationships based on the user's choices of meta-search and search engines. Specifically, the user (1) chooses a torrent to download, (2) selects a search engine to download the .torrent file, and (3) contacts the trackers included in the .torrent file to discover peers in the swarm.

A key observation is that the user's choice of search engine in (2) determines the subset of known trackers. This constrains the client to the peers known to the subset of trackers (users that are not grayed out), ultimately limiting the client's ability to discover local peers.

A. Survey of torrent and tracker listings

To determine the impact of search engine choice on peer discovery, we map out the relationships between search sites and trackers for the 100 most popular torrents from torrentz.eu.

Regardless of the search site chosen, we find that users are always confined to a subset of trackers. For each torrent and search site, we compute the fraction of trackers listed out of the total trackers for that torrent. Figure 4 plots CDFs across torrents of the median and maximum fraction of trackers listed. In the typical case – for the median torrent and search site – less than 60% of all trackers are provided. Even in the best case ("maximum" curve), users would still be missing over 25% of trackers.

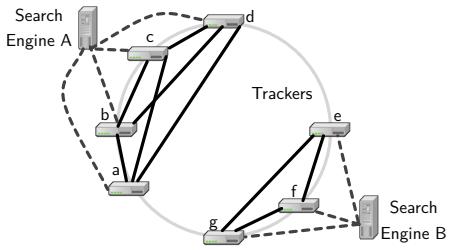


Fig. 5. Sample graph of trackers co-listed by search engines. Trackers a-d are listed on Search Engine A, while trackers e-g are listed on Search Engine B. If users download a torrent from A, then they will be part of the swarm on trackers a-d. Trackers a-d and trackers e-g form separate components; swarm populations of separate components do not overlap.

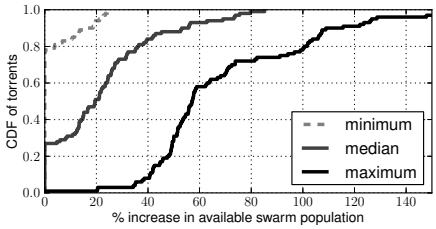


Fig. 6. % increase in known swarm population size when using all available trackers. For 75% of torrents, the median search site does not provide access to the full swarm population. In the worst case ("maximum" increase curve), utilizing all trackers can increase available swarm population by over 150%.

B. Improving peer discovery by using more trackers

We evaluate the extent to which knowledge of the swarm population is limited by the subset of trackers used. For each torrent, we use an undirected graph to model the relationships between search sites and the trackers that they reference. Each tracker is a vertex; when two trackers are listed by the same search engine, we add an edge to the graph between the trackers. Figure 5 shows an example of such a graph.

We use this representation to determine the increase in known swarm population when using all trackers – compared to the subset of trackers from one search site. We take a conservative approach to avoid double-counting peers present in multiple trackers' swarms by considering additional *components* in the graph we construct – which by definition have disjoint populations. In our example, the sets of trackers (a-d and e-g) are not listed together by any search engine, and therefore form two components. We under-estimate the population size of each component as the largest individual tracker population size in the component.

We determine the potential increase in swarm population as the percent increase from each search site's initial set of trackers to the total set of trackers. Figure 6 plots the minimum, median and maximum change in population across torrents. Starting with the trackers from a typical search site ("median" curve), 70% of torrents can expand the swarm population by using all trackers. The "maximum" curve gives an upper bound on this increase in population size, with a 57% increase in the median case and at most a 150% increase. Users can access a larger swarm population by utilizing all available trackers, increasing the probability of discovering local peers.

Since trackers impose rate limits on how frequently new peers can be requested, increasing the number of trackers that can be queried directly increases the sampling rate. Similar to our analysis of increase in swarm population, Figure 7 plots the minimum, median and maximum increases in the total number of trackers relative to the trackers listed on individual search sites. At a minimum, the number of trackers grows by 10%. The "maximum" curve shows the upper bound of benefits of this approach, with 95% of peers doubling the number of trackers and a median increase of 180%.

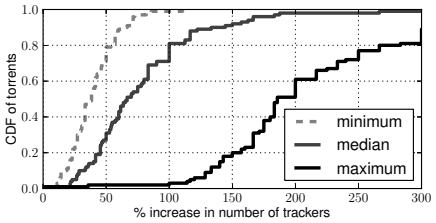


Fig. 7. % increase in number of trackers, relative to the number of trackers seen in .torrent files from a search engine site. Regardless of search site, there are always additional trackers available. For the median torrent, the number of trackers can be increased by 40% to 185%.

C. Pushing trackers to the limit

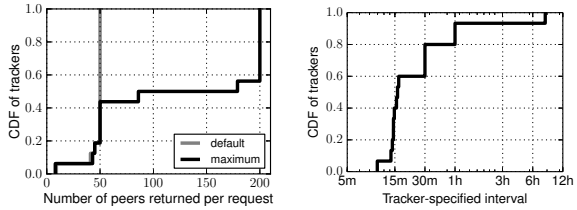
We evaluate the potential to increase the rate of peer discovery by probing trackers more aggressively. We test two approaches – requesting larger swarm samples size and making more frequent requests – using the trackers from our survey in Sec. V-A.

First, we attempt to increase the number of peers obtained from a request. Requests can optionally specify a sample size. To determine a tracker's default behavior, we request a sample of peers without including this parameter. Then, we determine the maximum sample size by iteratively increasing the requested sample size until the response stops growing. We ensure that swarm size is not a limiting factor by requesting samples from a very large torrent.

Figure 8a plots CDFs of the default and maximum sample sizes returned by each tracker. By default, 80% of trackers reply with at most 50 peers. Though 44% of trackers will not provide more than the default of 50 peers, the remaining 56% return up to 200 peers – a four-fold increase compared to the default. On average, this technique yields a 2x increase in peers per request.

Second, we determine whether trackers enforce their specified inter-request intervals, which defines the rate at which BitTorrent clients request additional peers. Figure 8b plots the distribution of these intervals seen across the trackers that we queried. For 60% of trackers, this value is about 15 minutes or less. To test this, we request samples from each tracker at increasingly shorter intervals until the inter-request interval is less than 10 seconds. *We were able to obtain swarm samples every few seconds from all trackers* – despite their specified intervals being on the order of minutes. Clients are therefore able to obtain multiple samples of the swarm population in a short time window. If we query trackers once per minute, this corresponds to an average increase by 70x in the rate at which we can obtain swarm samples.

Combining these approaches, clients can obtain swarm information from a single tracker *140x faster* compared to



(a) Sample size (b) Request interval

Fig. 8. Trackers' default parameters limit peer discovery rates. Many trackers will provide additional peers on request – up to 200, an average increase of 2x (left). Trackers typically ask clients to wait 15 minutes between requests, but we find that these rate limits are not enforced, enabling clients to more quickly obtain swarm information – up to 70x faster (right).

default behavior. Applying this approach to the full set of trackers available (see Fig. 7) further increases the peer discovery rate by 40% to 180%. Although these approaches do consume additional bandwidth and processing resources at the tracker, trackers can trivially control these additional costs by enforcing the rate limits that are built-in to the protocol. As we discussed in Sec. IV-A, we minimize the impact of these approaches by only employing them when we expect there are additional local peers in the swarm.

VI. Related Work

By default, BitTorrent peers make random connections to others in the swarm. Even when a local copy of data is available, 70-90% of the time peers download it from a non-local peer [14]. The resulting traffic can be costly to ISPs since they pay for inter-domain traffic to other networks.

Several systems [3], [5]–[7] have implemented biased neighbor selection with the goal of reducing inter-domain traffic and increasing peer download performance. Such systems influence the connections that each peer makes, nudging them towards preferring connections with nearby peers (e.g. within the same network) as opposed to randomly selected peers.

A key requirement of biased neighbor selection is the presence of locality in the swarm. Hoßfeld et al. conduct a detailed analysis of several BitTorrent datasets and report on temporal dynamics and the distribution of peers in ASes [8]. We go beyond the results in this work by quantifying the impact of diurnal patterns on local peer discovery and proposing an approach to predict future local peers online.

Another group of studies has analyzed the efficacy of biased peer selection systems [9], [15] and evaluated "What if?" scenarios that reveal large potential benefits with locality policies [14], [16], [17]. While Piatek et al. also find such potential, they note the difficulty in realizing these benefits because of the limited swarm knowledge at each peer [9]. In this work, we evaluate several techniques to overcome this limiting factor of incomplete swarm information and improve the efficacy of biased neighbor selection.

VII. Conclusions

Evaluating systems driven by end-user behavior is challenging due to the spatio-temporal characteristics and dynamics of their use. Results will vary depending on what one chooses to examine and when the measurement takes place. Any research on such systems should account for known behaviors and characterize newly observed dynamics to guide future work.

In this work, we identify dynamics in BitTorrent to better understand locality in swarms and guide our approach to improve local peer discovery. We show that client-based approaches to biased neighbor selection are generally challenged by the random sampling approach used by trackers and 2-3x diurnal variations in the number of online local peers. We documented that downloading during the peak hour maximizes the availability of local peers, with 50% of networks having at least 5 local peers online. To discover the locality hidden in the swarm, we evaluated several techniques that speed up the peer discovery process by over two orders of magnitude. These techniques are sufficient to provide complete knowledge of the swarm and allow peers to discover all locality in the swarm, thereby improving performance and reducing costly cross-ISP traffic.

Acknowledgments

We would like to thank the anonymous reviewers for their invaluable feedback and John Rula for his assistance with early drafts of the work. We are always grateful to Paul Gardner for his assistance with Vuze and the users of our software. This work was supported in part by the National Science Foundation through Awards CNS 0644062, CNS 0855253 and CNS 0917233.

References

[1] B. Cohen, "Incentives build robustness in BitTorrent," in *Proc. of the Workshop on Economics of Peer-to-Peer Systems (P2PEcon)*, 2003.
[2] Sandvine, "Global Internet phenomena report," 1H 2012, http://www.sandvine.com/news/global_broadband_trends.asp.
[3] 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*, 2008.
[4] J. S. Otto, M. A. Sánchez, D. R. Choffnes, F. E. Bustamante, and G. Siganos, "On blind mice and the elephant: Understanding the network impact of a large distributed system," in *Proc. of ACM SIGCOMM*, 2011.
[5] H. Xie, R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, "P4P: Provider portal for P2P applications," in *Proc. of ACM SIGCOMM*, 2008.
[6] M. Varvello and M. Steiner, "Traffic localization for DHT-based BitTorrent networks," in *Networking (2)*, 2011.
[7] 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 *Proc. of ICDCS*, 2006.
[8] T. Hoßfeld, D. Hock, S. Oechsner, F. Lehrieder, Z. Despotovic, W. Kellerer, and M. Michel, "Measurement of BitTorrent Swarms and their AS Topologies," University of Würzburg, Tech. Rep. 464, 2010.
[9] M. Piatek, H. V. Madhyastha, J. P. John, A. Krishnamurthy, and T. Anderson, "Pitfalls for ISP-friendly P2P design," in *Proc. of HotNets*, 2009.
[10] "BEP 5: DHT Protocol," http://www.bittorrent.org/beps/bep_0005.html.
[11] C. Zhang, P. Dhungel, D. Wu, Z. Liu, , and K. W. Ross, "BitTorrent darknets," in *Proc. of IEEE INFOCOM*, 2010.
[12] M. A. Sánchez, J. S. Otto, Z. S. Bischof, D. R. Choffnes, F. E. Bustamante, B. Krishnamurthy, and W. Willinger, "Dasu: Pushing experiments to the Internet's edge," in *Proc. of USENIX NSDI*, April 2013.
[13] Vuze, Inc., "Vuze," http://www.vuze.com.
[14] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should Internet service providers fear peer-assisted content distribution?" in *Proc. of IMC*, 2005.
[15] H. Wang, J. Liu, and K. Xu, "On the locality of BitTorrent-based video file swarming," in *Proc. of IPTPS*, 2009.
[16] R. Cuevas, N. Laoutaris, X. Yang, G. Siganos, and P. Rodriguez, "Deep diving into bittorrent locality," in *Proc. of IEEE INFOCOM*, 2011.
[17] S. Le-Blond, A. Legout, and W. Dabbous, "Pushing BitTorrent locality to the limit," *Computer Networks*, 2010.