Understanding Peer Distribution in the Global Internet

Jiangchuan Liu and Haiyang Wang, Simon Fraser University
Ke Xu, Tsinghua University

Abstract

The fast-growing traffic of peer-to-peer applications, most notably BitTorrent, is putting unprecedented pressure to Internet service providers. Understanding the peer distribution over the global Internet thus becomes critical toward building new generation of ISP-friendly peer-to-peer systems. There are unfortunately significant scalability and representability challenges in measuring and understanding real-world peer distribution. In this article we demonstrate a novel hybrid measurement methodology that uses the PlanetLab as a distributed probing platform to interact with BitTorrent trackers and peers in the global Internet. Our design achieves fast real-time scanning of genuine online peers; yet we carefully avoid the potential copyright infringement and traffic overhead for PlanetLab. From three months' data of over 9 million peers, we identify fundamental issues in conventional traffic locality designs, and also shed new light on re-engineering trackers and reusing historically downloaded data to make BitTorrent a better storage system.

Peer-to-peer (P2P) communications have gained tremendous popularity in the past decade. The most successful P2P file sharing application, BitTorrent, has enjoyed phenomenal growth since its deployment in 2001, and now contributes to almost 35 percent of Internet data exchanges. Its exceptional scalability and robustness come from the enormous computation, storage, and communication resources collectively available at participating peers.

Unfortunately, the ever increasing traffic among peers has also put unprecedented pressure on Internet service providers (ISPs), and the current BitTorrent design does not address the challenges from inter-autonomous system (AS) traffic well. Besides simple throttling of P2P flows, recent proposals have suggested topology-aware or access-locality-aware overlay construction to minimize neighbor distances and thus reduce long-haul traffic [1]. The peer distribution obviously plays an important role in the successful design and deployment of a new generation of ISP-friendly mechanisms. However, so far the distribution of BitTorrent peers has seldom been examined in the global Internet. As such, the potential benefit and even the applicability of the proposals in the real world remain unclear.

In this article we show that although BitTorrent is an open-source system, there are indeed significant challenges in measuring and understanding real-world peer distribution, particularly regarding scalability and representability concerns. We then demonstrate a novel hybrid measurement methodology that uses PlanetLab as a distributed probing platform to interact with BitTorrent trackers and peers in the global Internet. Our design achieves fast real-time scanning of genuine online peers, but we carefully avoid the potential copyright infringement and traffic overhead of PlanetLab. From three months' data of 9,111,245 peers, we show that BitTorrent peers exhibit strong geographical locality that could be explored. However, if we focus only on individual torrents, very few torrents are able to form large enough local clusters of peers. Even for the most popular ASs, the amount is lower than 5 percent, which makes state-of-the-art locality mechanisms for standalone torrents quite inefficient. Our measurement further shows the relation between peers and trackers. We reveal the pervasive penetration of multi-tracker configurations that potentially improves the availability of BitTorrent.

Our measurement identifies fundamental issues in conventional traffic locality designs, but also sheds new light. In particular, we suggest that the torrents can be treated as a storage system that effectively explores historically downloaded data. Meanwhile, the trackers should be re-engineered strategically, not only achieving the best coverage, but also minimizing the cost and thus ensuring deployability.

The rest of this article is organized as follows. In the next section we present an overview of BitTorrent. The challenges in measuring BitTorrent are discussed in the following section. We then introduce our measurement methodology and describe the peer distribution in the global Internet. We address the implications of our results and conclude the article in the final section.

BitTorrent Primer

We start with a brief overview of BitTorrent. The terminologies used in the BitTorrent community have yet to be standardized, and our descriptions are mainly adapted from the BT manual [2, 3].

A BitTorrent client, known as a *peer*, implements the P2P communication protocol in the application layer. The protocol breaks up each large file into hundreds or thousands of identically sized *pieces*, typically between 64 kbytes and 4 Mbytes each, which, once downloaded by a peer, can be shared with others while the downloading continues.

The file sharing among peers is coordinated by a centralized *tracker*. To share a file or group of files, a peer first creates a .torrent file, which contains metadata about the file(s), checksums (SHA-1 hash) of data pieces, and the tracker information. Others interested in the same file will first download the .torrent file and connect to the specified tracker, which tells them a list of available peers. The client will then contact each of the peers and gather information about which pieces are available from these neighbors for downloading. There is a user-configurable limit on the maximum number of neighbors a peer can maintain, which defaults to 80. There is also a limit on the number of real connections to neighbors a peer can initiate, which defaults to 40. A *rarest-first* policy then guides the piece downloading from the multiple neighbors, with rare pieces being offered higher priorities. Each downloaded piece's SHA-1 will be compared with that in the .torrent file to ensure correctness. Over time, there will be two types of peers in a swarm:[1] those who are still downloading the file (*leechers*) and those who have the complete file but continue uploading (*seeders*).

BitTorrent also implements a *tit-for-tat* incentive policy, which penalizes peers that do not upload (contribute) to the system, thus discouraging free-riding. The tracker coordination, together with the comprehensive set of scheduling and incentive policies, make BitTorrent very efficient and fair, surpassing the previous generation of P2P file sharing protocols. The random peer selection also ensures its robustness. Unfortunately, such a pure blind selection can be inefficient given that poor peers may only be identified after long contact. More important, it does not address the interplay with Internet ISPs well, so long-haul cross-ISP connections are inevitably introduced. Recent proposals have suggested topology-aware or access-locality-aware overlay construction, which strike to minimize neighbor distances and reduce long-haul traffic [1]. An understanding of peer distribution clearly plays an important role in their successful design and deployment.

Challenges of BitTorrent Measurement

There have been many theoretical analyses and optimizations of the BitTorrent system. There have also been numerous simulation studies, even with dedicated simulators being built [4]. These studies, however, cannot replace real-world measurement, particularly for the global distribution of BitTorrent peers in Internet ASs. Given that the BitTorrent system has already been implemented and its source code is open, measurement would seem easy. Yet there are a number of significant challenges, two critical ones of which involve scalability and representability.

Scalability Challenge

BitTorrent has evolved into an ultra-large file sharing system involving millions of peers. In the past PlanetLab [5] has been extensively used for measuring and evaluating P2P systems, including BitTorrent. This controlled academic network environment works well for understanding certain performance aspects (e.g., local traffic patterns and system dynamics). Unfortunately, it has very limited scale with less than 500 nodes, not all of them active all the time. As such, existing experiments over PlanetLab generally constitute a small number of peers, and such a scale limit affects the measurement results and even conclusions.

As an example, a pioneer work in [3] reported that BitTorrent overlay exhibits strong clustering behaviors. In contrast, our recent study involving 430 peers in long-term examination suggests that clusters may not exist or persist in larger torrents [6]. Figure 1 presents the connectivity matrix of peer connections during our experiment. In an earlier stage (4 h) there is noticeable clustering, evident from the concentration along the diagonal, which is consistent with the report in [3]. However, after long-term evolution when a steady stage is reached (32 h and beyond), the connectivity matrix becomes an almost completely random scattering of points, and the fanout in the lower left is almost not visible. We believe that the short experiment duration (about 1 h) and limited scale (40 peers), which is even smaller than the default number of connections a peer can maintain (80 in practical BitTorrent clients) likely contribute to the clustering observed in [3]. Such inconsistency questions the existence of clustering in ultra-large overlays of real-world torrents, which can hardly be addressed by a pure PlanetLab-based experiment.

Representability Challenge

An even more critical challenge is the representability of PlanetLab-based measurement. It has long been argued that, as an academic research platform, the PlanetLab's topology, capacity, and reachability do not necessarily reflect those of the Internet. Even worse, PlanetLab nodes do not or simply cannot host real-world trackers due to traffic and copyright concerns. A pure PlanetLab-based measurement thus cannot dig out the real-world peer distribution.

On the other hand, given that BitTorrent is an anonymous and distributed system, most of the tracker sites do not disclose the logs of participating peers. This is particularly true considering that many popular torrents involve copyright infringing contents. Traffic traces from a small set of core or edge routers can hardly be used to derive the global peer distribution, either, due to the small and often biased sample sets, not to mention the daunting task of information mining from the huge amount of data without clear semantics. We have seen pioneer work on re-engineering BitTorrent clients and letting them join real-world torrents, interacting with Internet peers to retrieve useful information [7]. Given the deployment difficulties, these experiments are generally limited to a few compromised peers residing on a campus or enterprise network. This results in a long time to scan a large collection of peers, which might not be online simultaneously. Even worse, many peers might not be reachable from the compromised peers. Representability thus remains questionable with partial (and possibly biased and unsynchronized) snapshots.

PlanetLab as a Distributed Probing Platform

To address these challenges, we propose a novel hybrid PlanetLab-Internet measurement methodology. Our design treats PlanetLab as a distributed probing platform, whose nodes serve as a collection of distributed probing agents to interact with real-world trackers and peers. We carefully avoid the potential copyright infringement or traffic overhead for PlanetLab, and thus have successfully obtained special approval from PlanetLab administrators.

We first extracted a large collection of real torrents as advertised by www.btmon.com, one of the most popular torrent sites, from February 2007 to August 2008. We developed a script to automatically detect the *href* field in each given HTML file and downloaded the metainfo files ending with .torrent, which resulted in 74,732 metainfo files. Within our data set, there are 316 bad metainfo files, 1027 unavailable torrents due to tracker failures, and 3340 torrents having only one peer. We excluded these abnormal torrents, and, to balance accuracy and measurement overhead, randomly selected 8893 of the 70,049 normal torrents for our study.

We then ran a modified version of CTorrent (a typical BitTorrent client in FreeBSD) on the PlanetLab nodes. Different from conventional pure PlanetLab experiments in which the clients communicate with others within the PlanetLab only, our modified CTorrent clients actively joined existing torrents in the global Internet and recorded the observable peer information from the trackers and from other peers over time. As

[1] *We use* swarm *and* torrent *interchangeably in this article; both refer to the collection of peers sharing the same file(s).*

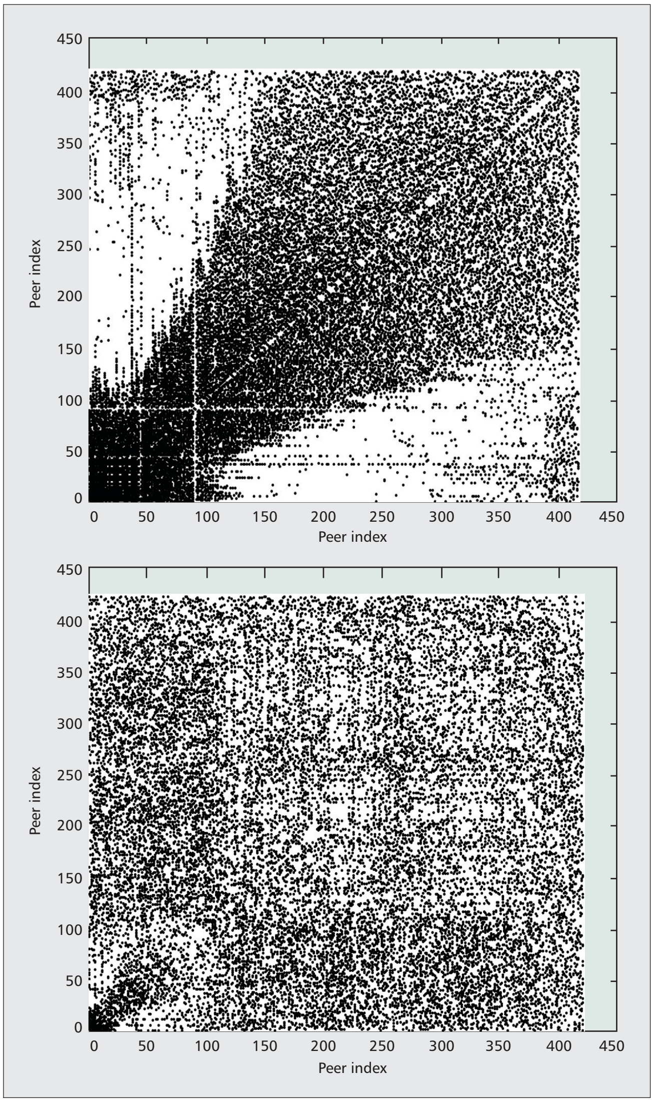

Figure 1. *Connective matrices at hour 4 and 32, respectively. The connectivity matrix is a scatter plot, where a point at location (i, j) in the plot refers to the fact that peer i is connected to peer j. The peer index is created by sorting the peers by their joining time.*

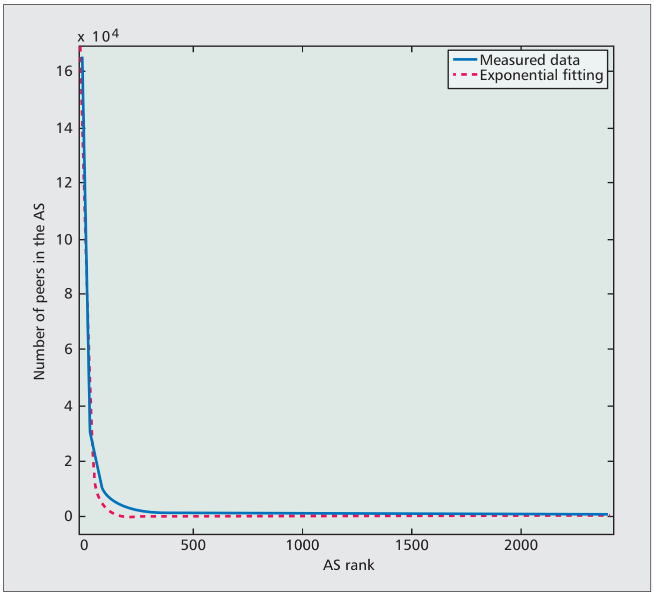

Figure 2. *Total peer popularity of 2864 ASs.*

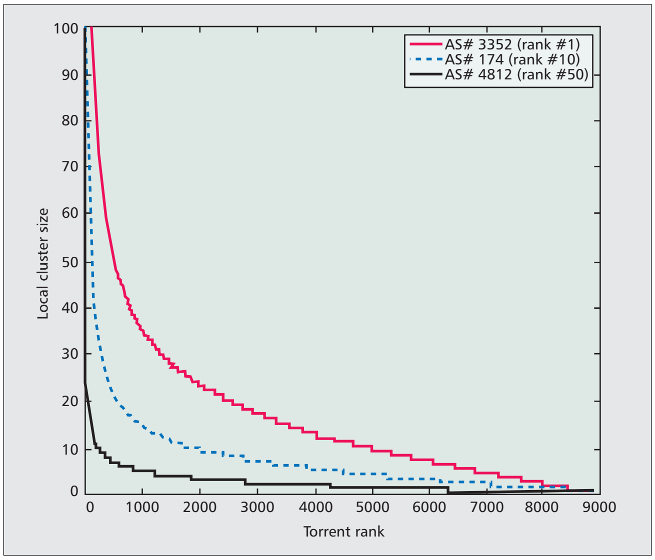

Figure 3. *Distribution of local clusters.*

such, the small set of controlled PlanetLab nodes were able to capture the information of most peers in the torrents; in particular, their IP addresses. With a maximum of 50 initial peers from the trackers, we successfully detected the IP addresses of over 95 percent of the peers for most of the torrents.[2]

Except for retrieving the peer existence and address information, our PlanetLab clients did not download or upload any real data of the shared contents. Hence, no copyrights were violated, and the impact on PlanetLab traffic and the operations of normal trackers/peers were minimized. We have examined the content size of all the torrent files in our experiment. In particular, video torrents generally have large sizes, with a mean of approximately 1000 Mbytes. Also, 90 percent are larger than 100 Mbytes, and 5 percent are even larger than 10 Gbytes, with the maximum being nearly 20 Gbytes. The sizes of non-video content are relatively smaller, with only 30 percent of them larger than 100 Mbytes and over 50 percent less than 20 Mbytes. If we were to participate with brute force in the real downloading of these torrents, the enormous traffic volume, even for the non-video contents, would simply crash PlanetLab. In fact, we have found that about 77 percent torrents are for large video contents [8], and many of them involve copyright violations.

Without real content downloading, the scanning efficiency of our experiment is very high, and most of the torrent can be scanned within a short timeframe (among the 8893 torrents, only 20 of them cannot always be finished scanning within 20 s). In other words, the detected peers in these torrents can be considered to be online simultaneously, which is important to our discussions in the following sections. To avoid biases, we have also filtered out all the PlanetLab nodes in the data presented in the following analysis.

The source code of the modified CTorrent client and the raw dataset (including the torrents information) can be found at http://netsg.cs.sfu.ca/BT locality/dataset.htm.

Peer Distribution: Measurement Results

Our PlantLab probing platform has provided us a sheer number of data traces about real-world BitTorrent peers, and we have found that their IP addresses span quite universally, cover-

[2] *This ratio is calculated by comparing the number of detected peers with the total number of peers as advertised by the tracker of a torrent.*

Rank	Peers	Torrents*	Tracker sites (URLs)
1	607987	19915	open.tracker.thepiratebay.org
2	593205	16724	trackeri.rarbg.com
3	560580	23386	denis.stalker.h3q.com
4	509140	15308	tpb.tracker.thepiratebay.org
5	504173	12117	vtv.tracker.thepiratebay.org
6	442708	12821	vip.tracker.thepiratebay.org
7	414095	10019	eztv.tracker.prq.to
8	262991	6079	tracker.prq.to
9	184843	3016	tk2.greedland.net
10	142220	3114	www.sumotracker.org

*Note that the torrent level popularity is obtained form the metainfo files which can include multiple trackers.

Table 1. *Top 10 most popular trackers.*

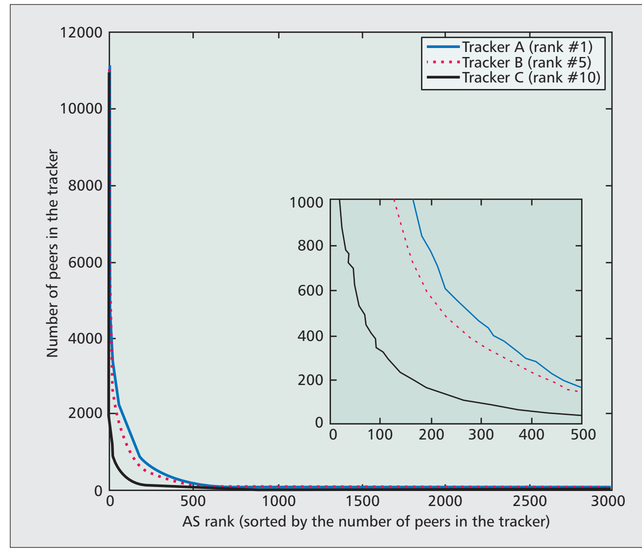

Figure 4. *Distribution of peers' location on trackers.*

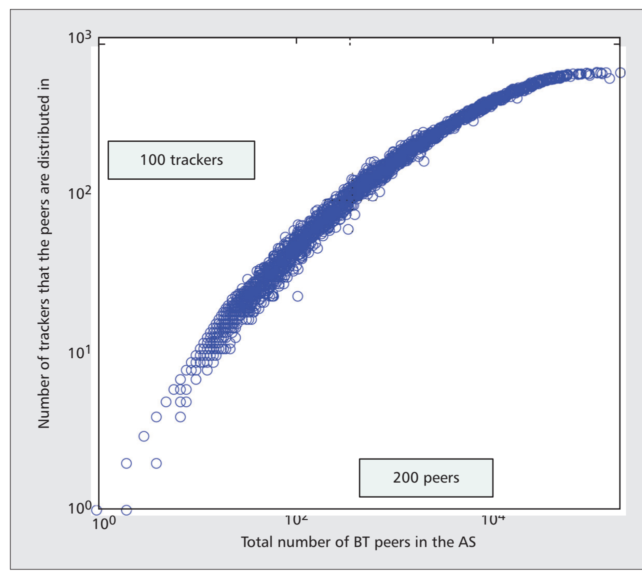

Figure 5. *AS's peer population vs. number of trackers.*

ing almost 80 percent of the whole IPv4 space. The ratio can be even higher given that many missing pieces are indeed reserved prefixes (e.g., 192.168.0.0/16 and 169.254.0.0/16). That said, PlanetLab as a distributed probing platform is quite successful in capturing peers globally, without noticeable blind spots.

Peer Distribution across ASs

Given the IP addresses of the peers, we extracted their corresponding ASs through the `whois` command in Linux. This results in 2405 distinct ASs, and Fig. 2 shows the peer popularity across all torrents in these ASs. We see that it can be fitted roughly by an exponential distribution ($y = a^{bx}$, where $a = 1.261 \times 10^5$, $b = -0.0480$); in other words, despite the common belief that BitTorrent is extremely popular everywhere, a majority of the ASs indeed do not host a huge number of BitTorrent peers; in fact, 65 percent of them have less than 100 peers across all torrents.

On the other hand, the distribution does imply that BitTorrent peers exhibit strong geographical locality that could be explored. We find that the top ranked ASs have very dense peer populations, hosting thousands of peers. These ASs therefore should be the target of applying and optimizing traffic locality mechanisms.

Since the existing locality mechanisms have focused on individual standalone torrents only, it is important to further investigate the distribution of local clusters, where a local cluster is the collection of local peers downloading the same content in an AS. Unfortunately, as shown in Fig. 3, even for very popular ASs, only a few torrents are able to form large local clusters. As an example, in the most popular AS (*AS*3352) most of the torrents (over 95 percent) have less than 50 peers, even though these torrents are of quite large client populations (generally more than 500 peers). A close look reveals that the peers of most torrents are distributed in more than 150 ASs, thus unavoidably involving extensive cross-AS communications.

Peer Distribution across Trackers

We next examine the peer distribution across trackers. Table 1 lists the top 10 tracker sites in terms of peer population. It is obvious that there is a high concentration: these 10 tracker sites manage over 45 percent of all the peers we observed in our measurement, and the top 7 occupy over 35 percent of them. We also cross-check the peer distribution in trackers and ASs. Using the top 1, 5, and 10 tracker sites as examples, we show the distribution of their managed peers in different ASs in Fig. 4. Again, the data can be fitted by an exponential distribution, suggesting strong geographical locality from the tracker sites' view. On the other hand, Fig. 5 shows that the peers in an AS are generally managed by diverse individual trackers. Even for some less popular ASs (with around 200 peers), their peers can be managed by more than 100 trackers. There is also a very strong correlation (0.7046) between the total number of peers in the AS and the number of trackers that manage the peers for this AS. That said, the peers in a large AS will be managed by many more trackers. This indeed makes re-engineering BitTorrent quite difficult even with the available geographical locality.

It is also worth noting that most of the trackers belong to public tracker sites, such as Pirate Bay, which has been involved in a series of lawsuits as plaintiff or defendant. Unless copyrights and related problems are well solved, we can hardly expect to re-engineer these public tracker sites, not to mention organizing them together for global optimization. The concentration also implies that the whole BitTorrent system can be vulnerable, given the great hazard that a tracker site can be ordered to shut down at any time, like the earlier incident of the Napster system.

Another interesting observation, however, is that the latest BitTorrent metainfo file can include multiple tracker sites stored in the *announce-list* section [9]. This multitracker configuration allows peers to connect to more than one tracker at the same time, which brings two tangible benefits: better accommodation of tracker failures and balancing the load among trackers. To understand how popular the multitracker configuration is in practice, we record the announce-list of the torrents in our dataset; the result indicates that more than 90 percent of torrents have specified at least two trackers, and a few torrents even have announce-lists of multiple hundreds of trackers. This is much higher than an earlier measurement in 2007 [10] (observed multitrackers in 35 percent of torrents), and thus suggests the multitracker configuration has been quickly recognized and deployed in the BitTorrent community. This improves the availability of BitTorrent and largely mitigates the impact of the potential shutdown of centralized tracker sites.

Discussion and Conclusion

We now proceed with the opportunities that can be explored and the challenges that must be addressed from the above measurement.

BitTorrent as a Storage System

Our results suggest that while peers exhibit strong geographical locality, focusing exclusively on standalone torrents can hardly be effective given the pervasive distribution of peers across torrents and trackers. Also, if we focus only on local peers that simultaneously participate in the same torrent, once a peer leaves the torrent, its downloaded contents will become invisible immediately.

Fortunately, earlier studies have revealed that over 85 percent of peers indeed remain in the BitTorrent system, participating in other torrents after their departure [11]. If the trackers can keep tracking those peers remaining in the system, the available local peers for most torrents could be increased significantly. With our data, we have validated that the peer population of most torrents (more than 85 percent) can be tripled after 10 h. In other words, beyond individual file swarming, BitTorrent can be treated as a distributed file storage system, albeit with high churns. Our measurement shows that some peers have downloaded as many as 5000 pieces of content across different torrents. Intuitively, these peers have the potential to work as file storage servers and thus improve the overall availability and locality.

The new challenge, however, is on tracing the peers across torrents. That is, if a peer has finished downloading in a torrent (say torrent 1) and left, but remains in other torrents, how can we detect it to recover the previously downloaded content to facilitate the locality for the remaining peers in torrent 1?

Strategic Trackers Re-Engineering

We are particularly interested in *tracker-and-client-based* solutions. For example, to explore locality, a tracker can be modified to replace random peer selection by *AS hop-count*-based selection [12]. While this approach relies only on modifications to end-system implementations, the huge number of peers and trackers observed in our measurement implies that a universal upgrade of all of them can hardly be realized. Instead, we should resort to strategic re-engineering of selected trackers and their associated ASs, and the skewed distributions suggested by our measurement provide valuable guidelines for selection.

For the tracing problem, the multitracker configuration opens great opportunities, too. Specifically, assume torrent 1 and torrent 2 are both managed by tracker *A*; any peer migrating between these two torrents can simply be detected by tracker *A* without communication to other trackers. While this seems to be an ideal case, we have found that with high penetration of multitrackers, it indeed exists and is not uncommon. About 45 percent of transitions can be detected and, with better modeling of peer residency patterns, the ratio can be improved to 75 percent.

Another attractive solution would be a tracker overlay for tracker-to-tracker communications and collaborations [11]. Unfortunately, our measurement does not endorse such advanced collaborations. Besides overheads, enforcing communications between public trackers can be quite difficult given the copyright concerns discussed earlier.

Acknowledgment

This research is supported by a Canada NSERC Strategic Grant, an NSERC Discovery Grant, an NSERC DAS grant, and a MITACS Project Grant.

References

[1] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should Internet Service Providers Fear Peer-Assisted Content Distribution?" *Proc. ACM/USENIX IMC*, 2005.

[2] BitTorrent Protocol Specification; http://www.bittorrent.org/beps/bep_0003.html

[3] A. Logout *et al.*, "Clustering and Sharing Incentives in BitTorrent Systems," *Proc. ACM SIGMATERICS*, 2007.

[4] BT-SIM; http://theory.stanford.edu/~cao/btsimcode.tgz

[5] Planetlab; http://www.planet-lab.org/

[6] C. Dale *et al.*, "Evolution and Enhancement of BitTorrent Network Topologies," *Proc. IEEE IWQoS*, 2008.

[7] M. Piatek *et al.*, "Pitfalls for ISP-friendly P2P Design," *Proc. ACM HOTNETS*, 2009.

[8] H. Wang, J. Liu, and K. Xu, "On the Locality of BitTorrent-Based Video File Swarming," *Proc. USENIX IPTPS*, 2009.

[9] BitTorrent Multi-Tracker Specification; http://www.bittornado.com/docs/multitracker-spec.txt

[10] G. Neglia *et al.*, "Availability in BitTorrent Systems," *Proc. IEEE INFOCOM*, 2007.

[11] L. Guo *et al.*, "Measurements, Analysis, and Modeling of BitTorrent-Like Systems," *Proc. ACM/USENIX IMC*, 2005.

[12] R. Bindal *et al.*, "Improving Traffic Locality in BitTorrent via Biased Neighbor Selection," *Proc. IEEE ICDCS*, 2006.

Additional Reading

[1] S. L. Blond, A. Legout, and W. Dabbous, "Pushing BitTorrent Locality to the Limit," INRIA Tech. Rep., 2008.

[2] H. Xie *et al.*, "P4P: Provider Portal for Applications," *Proc. ACM SIGCOMM*, 2008.

[3] D. R. Choffnes and F. E. Bustamante, "Taming the Torrent: A Practical Approach to Reducing Cross-ISP Traffic in Peer-to-Peer Systems," *Proc. ACM SIGCOMM*, 2008.

[4] G. Dan and N. Carlsson, "Dynamic Swarm Management for Improved BitTorrent Performance," *Proc. USENIX IPTPS*, 2009.

[5] B. Liu *et al.*, "Locality-Awareness in BitTorrent-Like P2P Applications," *IEEE Trans. Multimedia*, vol. 11, no. 3, 2009, pp. 361–71.

[6] Y. Yang, A. L. H. Chow, and L. Golubchik, "Multi-Torrent: A Performance Study," *Proc. IEEE/ACM MASCOTS*, 2008.

Biographies

JIANGCHUAN LIU [S'01, M'03, SM'08] (jcliu@cs.sfu.ca) is currently an associate professor in the School of Computing Science, Simon Fraser University, British Columbia, Canada, and was an assistant professor in the Department of Computer Science and Engineering at the Chinese University of Hong Kong from 2003 to 2004. His research interests include multimedia systems and networks, wireless ad hoc and sensor networks, and peer-to-peer and overlay networks. He is a member of Sigma Xi. He is an Associate Editor of *IEEE Transactions on Multimedia*, an editor of *IEEE Communications Surveys & Tutorials*, and an Area Editor of *Computer Communications.* He is TPC Vice Chair for Information Systems of IEEE INFOCOM 2011.

HAIYANG WANG (hwa17@cs.sfu.ca) is currently a Ph.D. student in the School of Computing Science, Simon Fraser University. He is working in the Multimedia and Wireless Networking Group, and his research interests include peer-to-peer networks, multimedia systems/networks, IP routing, and QoS.

XU KE [SM] (xuke@csnet1.cs.tsinghua.edu.cn) received his B.S., M.S., and Ph.D. degrees in computer science from Tsinghua University, China, in 1996, 1998, and 2001, respectively. Currently he is a professor in the Department of Computer Science of Tsinghua University. His research interests include next-generation Internet, switch and router architecture, P2P, and overlay networks. He is a member of ACM.