

Exploring Peer-to-Peer Locality in Multiple Torrent Environment

Haiyang Wang, *Student Member, IEEE*, and Jiangchuan Liu, *Senior Member, IEEE*

Abstract—The fast-growing traffic of Peer-to-Peer (P2P) applications, most notably BitTorrent (BT), is putting unprecedented pressure to Internet Service Providers (ISPs). P2P locality has, therefore, been widely suggested to mitigate the costly inter-ISP traffic. In this paper, we for the first time examine the existence and distribution of the locality through a large-scale hybrid PlanetLab-Internet measurement. We find that even in the most popular Autonomous Systems (ASes), very few individual torrents are able to form large enough local clusters of peers, making state-of-the-art locality mechanisms for individual torrents quite inefficient. Inspired by peers' multiple torrent behavior, we develop a novel framework that traces and recovers the available contents at peers across multiple torrents, and thus effectively amplifies the possibilities of local sharing. We address the key design issues in this framework, in particular, the detection of peer migration across the torrents. We develop a smart detection mechanism with shared trackers, which achieves 45 percent success rate without any tracker-level communication overhead. We further demonstrate strong evidence that the migrations are not random, but follow certain patterns with correlations. This leads to *torrent clustering*, a practical enhancement that can increase the detection rate to 75 percent, thus greatly facilitating locality across multiple torrents. The simulation results indicate that our framework can successfully reduce the cross-ISP traffic and minimize the possible degradation of peers' downloading experiences.

Index Terms—BitTorrent, traffic locality, measurement.



1 INTRODUCTION

PEER-TO-PEER (P2P) communications have gained tremendous popularity in the past decade. The most successful peer-to-peer file sharing application, BitTorrent (BT), enjoys phenomenal growth since its deployment in 2001, and now contributes to almost 35 percent of Internet's data exchanges [1]. 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 the peers has also put unprecedented pressure to Internet Service Providers (ISPs). In particular, even though many BT peers interested in identical contents are located in the same or nearby Autonomous Systems (ASes), they are unnecessarily connected in the existing BT systems, thereby persistently increasing the costly cross-AS/ISP traffic.

To alleviate the cross-AS traffic, many solutions have been proposed beyond the straightforward throttling of P2P flows [2]. Among them, P2P locality [3] has been widely suggested, which explores the access localities to reduce the long-haul traffic. Yet, so far the distribution of BT peers has seldom been examined in the global Internet [4]. As such, the potential benefit and even the applicability of the locality mechanisms in the real-world remain unclear.

In this paper, we for the first time examine the existence and distribution of peer locality through a large-scale

hybrid PlanetLab-Internet measurement. Our experiment uses the PlanetLab test bed [5] as a large collection of distributed probing nodes to interact with real-world trackers and peers, and yet it carefully avoids the potential copyright infringement or traffic overhead to the PlanetLab. Our measurement lasts three months, collecting information from more than 800,000 peers. The results demonstrate that the BitTorrent peers do exhibit strong geographical locality that could be explored. Unfortunately, if we focus only on individual torrents, very few torrents are able to form large enough local cluster of peers. Even for the most popular ASes, this ratio is less than 5 percent, which makes state-of-the-art locality mechanisms for individual torrents quite inefficient.

Recent measurements, on the other hand, suggest that over 85 percent of the peers indeed participate in multiple torrents [6], which is also validated by our data. Inspired by this, we develop a novel framework that traces and recovers the available contents at peers across multiple torrents, thus effectively promoting the locality. We address the key design issues in this framework, particularly, the detection of peer migration across the torrents. We demonstrate that the detection does not necessarily involve complex and costly tracker-level cooperations. Instead, a clever use of shared trackers can successfully detect around 45 percent of the peer migrations without extra communication overhead. We further demonstrate strong evidence that the migrations are not totally random, but follow certain patterns with correlations. This leads to *torrent clustering*, a practical enhancement with automated tracker selection. We also present a simple implementation of the torrent clustering, which effectively increase the detection rate to 75 percent, and thus greatly facilitates locality across torrents.

The performance of our locality mechanism across multiple torrents has been evaluated through extensive

- The authors are with the School of Computing Science, Simon Fraser University, Burnaby (Metro-Vancouver), BC V5A 1S6, Canada. E-mail: {hwa17, jcliu}@cs.sfu.ca.

Manuscript received 9 Apr. 2011; revised 25 Aug. 2011; accepted 7 Sept. 2011; published online 30 Sept. 2011.

Recommended for acceptance by A. Kshemkalyani.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-04-0207. Digital Object Identifier no. 10.1109/TPDS.2011.253.

trace-driven simulations with various detection rates. Compared to state-of-the-art locality mechanisms for individual torrents, our solution improves the local content availability, thus significantly reducing cross-AS traffic. In addition, it brings minimal impact to the peer downloading experiences.

The rest of this paper is organized as follows: in Section 2, we define the terminologies and list the related works. We then present our measurement results in Section 3, which reveal the challenges to the design and implementation of P2P locality. In Sections 4 and 5, we explore the P2P locality across multiple torrents, and present an effective detection mechanism for peer migration. Section 6 further describes an enhancement through torrent clustering, and Section 7 provides some related discussions. Finally, after the trace-driven evaluation in Section 8, we conclude the paper and offer some future directions in Section 9.

2 TERMINOLOGY AND RELATED WORK

2.1 Terminologies

The terminologies used in the BitTorrent community have yet to be standardized. For clarity of exposition, we first define a series of terms to be used in this paper.¹

Torrent. A BT torrent is the set of peers cooperating to download the same content using the BitTorrent protocol.

Metainfo file. A metainfo file (torrent file) contains all the information used to download the content of interest, including the number of pieces, SHA-1 hashes, the tracker information etc.

Tracker. A tracker is the only centralized component in a torrent. It is not involved in the actual distribution of the content, but keeps track of all peers currently participating in the torrent, and also collects statistics.

Multitracker. The multitracker configuration extends the metainfo file to enable multiple trackers for one torrent. Should one tracker fail, the others can continue supporting the torrent.

Local cluster. It is a set of local peers that download the same content in a given AS. The size of local clusters indicates the amount of local resources available in the AS.

2.2 Related Works

There have been numerous studies on the implementation, analysis, and optimization of the BitTorrent system; see surveys in [9]. P2P locality has recently attracted particular attention following the pioneering work of Karagiannis et al. [3]. Based on real traces and simulated torrents, they proposed the concept of locality in peer-to-peer systems and evaluated its benefit. Blond et al. [10] showed through a controlled environment that high locality values (defined by [3]) yield up to two orders of magnitude savings on cross-AS traffic, without any significant impact to the peers' download completion time. Xie et al. [11] further suggested cooperation between peer-to-peer applications and ISPs by a new locality architecture, namely, P4P, which can reduce both the external traffic and the average downloading time.

1. Our definitions are mainly adapted from the BT manual [7] and [8]. We notice that there are some slight differences across these sources, which, however, will not affect our general observations and conclusions.

Choffnes and Bustamante [12] proposed Ono, a BitTorrent extension that leverages a Content Distribution Network (CDN) infrastructure, which effectively locates peers that are close to each other. Bindal et al. [13] also examined a novel approach to enhance BitTorrent traffic locality, namely, *biased neighbor selection*. Using this method, a peer chooses the majority, but not all, of its neighbors from peers within the same ISP.

Our work extends these studies through an Internet-wide measurement that reveals the global distribution of BitTorrent peers as well as the associated tradeoffs of locality. In particular, we demonstrate that the effectiveness of a locality mechanism can be limited within individual torrents (on which the previous studies have focused), and it is necessary to explore the locality across the multiple torrents.

Guo et al. [6] revealed that more than 85 percent of all peers participate in multiple torrents and noted the peer migration behavior. This *migration behavior* indicates that some BT peers have the potential to serve others even when they have already left the swarm. They proposed an intertorrent approach through tracker-level collaborations. The main idea is to build a tracker site overlay for tracker-level collaboration; the peers migrating between different torrents can then be detected and recovered as potential seeders for the torrents. Dan and Carlsson [14] further investigated how the separated torrents can be merged together to improve the performance of an entire torrent. The measurement from Piatek et al. [15] however found that about 91 percent of peers in any single swarm do not arise in any other swarms. This observation seems to contradict the study in [6]; yet this is mainly due to the difference of their objective as well as their measurement schemes. On other hand, the measurement study by Neglia et al. [16] investigated the availability of BitTorrent system among different tracker configurations. The popularity and the performance of the multitracker configuration [17] were discussed. Their study showed that around 35 percent of the torrents enable multitracker configurations. Pouwelse et al. [18] further discussed the relationship between BT trackers and torrents, and examined the tracker availability across multiple websites, albeit with individual torrents.

It is worth noting that, the studies of content bundling [19] also provide useful insights to understand multiple torrents behavior. A pioneering work from Menasche et al. [19] studied the content unavailability problem in the BitTorrent system. This study for the first time proposed a model to analyze the availability and the performance implications of bundling through an extensive measurement. Follow up studies such as [20] and [21] also studied some other aspects for content bundling in BitTorrent systems.

Considering the measurement and incentive of BitTorrent system, a recent study from Dhungel et al. [22] examined BitTorrent darknets from macroscopic, medium-scopic and microscopic perspectives and investigated the properties of private BitTorrent sites. The study from Otto et al. [4] presented a comprehensive view of BitTorrent, using data from a representative set of 500,000 users sampled over a two year period, located in 169 countries and 3,150 networks. This study showed that the BT traffic exhibits significant locality across geography and networks. Compare to this

study, our work is more focusing on the peer distribution in different torrents/locations. We find that very few individual torrents are able to form large enough local clusters of peers, and this is generally due to the skewed distribution of torrents' popularity. Fan et al. [23] investigated the fundamental tradeoff between keeping fairness and providing good performance for BitTorrent system. Piatek et al. [24] showed that a "win-win" outcome is unlikely to obtain for the ISPs during the locality; the reason is that reducing interdomain traffic reduces costs for some ISPs, while it also reduces revenue for others. Cuevas et al. [25] also investigated the maximum transit traffic reduction as well as the "win-win" boundaries across the ISPs.

Our work was motivated by these studies; yet we explore the multitracker configuration across multiple torrents simultaneously, providing a seamless and lightweight solution to locality in the real BitTorrent system.

3 PEER DISTRIBUTION: A HYBRID PLANETLAB-INTERNET MEASUREMENT

To understand the potentials and difficulties of applying the locality mechanisms, we first examine the global distribution of BitTorrent peers in the Internet Autonomous Systems. This seemingly easy task indeed involves many challenges. First, 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 of the popular torrents involve copyright-infringing contents. On the other hand, traffic traces from a small set of core or edge routers can hardly be used to derive the global peer distribution, due both to the small sample size and the lack of semantics within the data.²

3.1 Hybrid PlanetLab-Internet Measurement Methodology

To address these challenges, we have applied a hybrid PlanetLab-Internet experiment for the measurement (the term "hybrid" means that the swarms consist of peers from both Internet and PlanetLab platforms). Our design uses the PlanetLab [5] as a large collection of distributed probing nodes to interact with real-world trackers and peers.

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, 1,027 unavailable torrents due to tracker failures, and 3,340 torrents having only 1 peer. We excluded these abnormal torrents, and to balance accuracy and measurement overhead, randomly selected 8,893 out of the 70,049 normal torrents for our study.

We then ran a modified version of CTorrent (a typical BitTorrent client in FreeBSD) [26] 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

2. This method was also applied by some other studies such as [19].

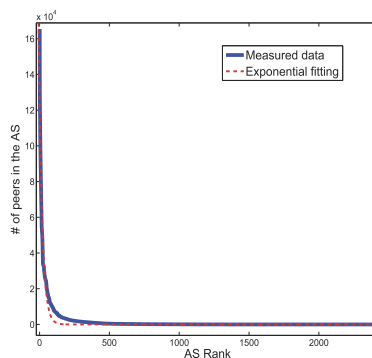


Fig. 1. Total peer popularity of 2,864 ASes.

joined existing torrents in the global Internet and recorded the observable peer information from the trackers and from other peers over time. As 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 peers for most of the torrents.³

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 to the PlanetLab traffic and to the operations of normal trackers/peers were minimized. The scanning efficiency of the experiment is also very high, with most of the torrent being finished scanning within a short timeframe (<20 sec); in other words, the detected peers can be considered as being online simultaneously, which is important to our discussions in the following sections.⁴ To avoid biases, we have also filtered out all the PlanetLab nodes in the data of following analysis.

Our source code of the modified CTorrent client and the full raw data set (including the torrents information) can be found at: http://netsg.cs.sfu.ca/BT_locality/data_set.htm.

3.2 Peer Distribution in ASes and Torrents

Given the IP addresses of the peers, we extracted their corresponding ASes through the "whois" command in Linux. This resulted in 2,405 distinct ASes, and Fig. 1 shows the peer popularity across all torrents in these ASes. We can see that it can roughly be fitted 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 ASes indeed do not host a noticeable number of BitTorrent peers, e.g., 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. As shown in Table 1, the top-ranked ASes have very dense peer populations, hosting thousands

3. 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.

4. Note that our measurement cannot guarantee to discover the entire peer information in a short timeframe for some huge torrents (with more than 5,000 peers). However, they constitute only a small portion in our data sets (20 out over 8,893 torrents), and thus will not affect our main observations.

TABLE 1
Top 10 ISPs/ASes in Terms of Peer Population

Rank	AS#*	Peers	Internet Service Providers
1	3352	165469	TELEFONICA-DATA-ESPANA
2	3662	129047	DNEO-OSP7-COMCAST
3	6461	127297	MFNX MFN-METROMEIDA
4	2119	113597	TELENOR-NEXTEL T.NET
5	19262	101390	TRANSIT-Verizon ISP
6	3301	97658	SWEDEN TELIANET
7	3462	96564	HINET-DATA CBG
8	4134	87392	CHINANET-BACKBONE
9	6327	86964	SHAW COMMUNICATION
10	174	74453	COGENT COGENT/PSI

*AS numbers assigned by IANA.

of peers. These ASes, therefore, should be the target of applying and optimizing P2P locality mechanisms.

Since the existing locality mechanisms have focused on individual 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. 2, even for the very popular ASes, only a few torrents are able to form large local clusters. As an example, in the most popular AS (AS3352), 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 ASes (the big picture of this distribution is shown in Fig. 3), thus unavoidably involving extensive cross-AS communications. We have also quantified the likelihood of the existence of local clusters through an entropy-based model; please refer to Appendix, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPDS.2011.253>.

Such results suggest that a locality mechanism designed exclusively for individual peers can be ineffective for many of the torrents. In addition, since it only works with local peers that simultaneously participate in the same torrent; once a peer leaves the torrent, its downloaded contents will become invisible immediately. Fortunately, recent studies

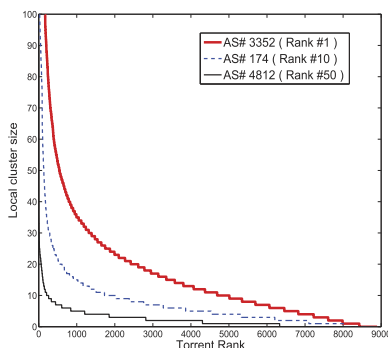


Fig. 2. Distribution of local clusters.

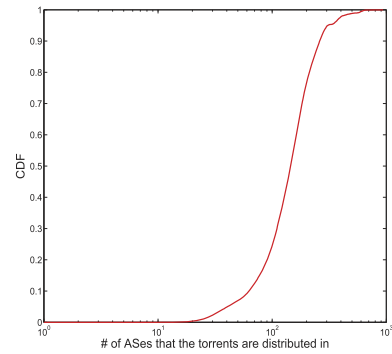


Fig. 3. Peer distribution of the torrents.

have revealed that over 85 percent of the peers indeed remain in the BT system, participating in other torrents after their departure [6]. Assume that the trackers can keep tracking those peers remaining in the system, the available local peers for most torrents could be increased significantly. Fig. 4 validates the potentials of this locality approach across multiple torrents, where the peer population of most torrents (more than 85 percent) is tripled after 10 hours.

4 P2P LOCALITY ACROSS MULTIPLE TORRENT: AN OVERVIEW

We now proceed with a framework design for exploring P2P locality across multiple torrents. We particularly focus on the *tracker-and-client-based* solutions [13], which rely only on modifications to end-system implementations. These locality solutions typically replaces the random peer selection by an *AS hop count*-based metric. Upon a request, the modified tracker sorts all other peers in the torrent in ascending order of their AS hop count to the requesting peer, and then sends the prefix of this sorted list (e.g., first 50 peers) to the requesting peer. The requesting peer would then choose the majority, but not all, of its neighbors from peers within the same ISP. Typically, 35 peers within the same ISP (AS hop count 0) can be returned together with 15 other random peers [13].

For the individual torrent scenario, many neighbor selection approaches have been proposed [13], [27], which could also be applied in the multiple torrent scenario. The new challenge, however, is the detection of peer migrations

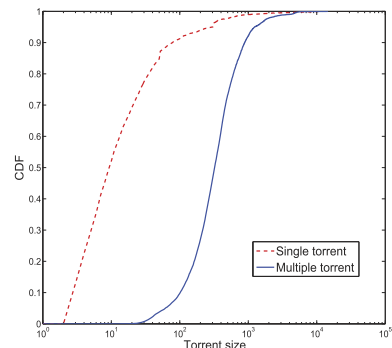


Fig. 4. Single torrent versus multiple torrents.

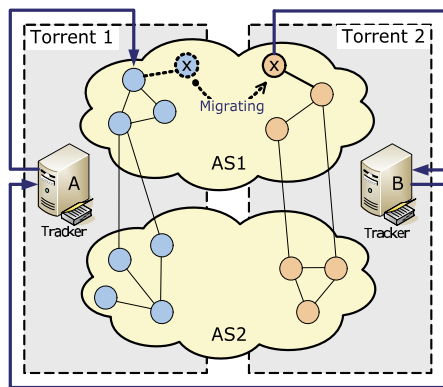


Fig. 5. Multiple torrent-based P2P locality.

among torrents. That is, if a peer has finished downloading in a torrent (say Torrent 1) and left, but remains in other torrents,⁵ how can we detect it, so as to recover the previously downloaded content to facilitate the locality for the remaining peers in Torrent 1? This is illustrated in Fig. 5, where peer x leaves Torrent 1, but remains in Torrent 2. If this migration can be detected, peer x can still serve as a potential seeder for Torrent 1, which will greatly promote the locality for the peers in AS1.

We can see that the solution may need a tracker overlay for tracker-to-peer and tracker-to-tracker communications; in particular, adding extra collaboration among the trackers to trace the migration of peer x [6]. Unfortunately, besides the overheads, enforcing communications between the public trackers can be quite difficult. Table 2 lists site information of the Top 10 most popular trackers in our measurement. We can see that many of them belong to Pirate Bay, which has been involved in a series of lawsuits, as plaintiffs or as defendants. Unless the problem can be well solved, we can hardly expect to organize these public tracker sites together for optimization. We, thus, resort to solutions that minimize the communications, especially tracker-to-tracker communications.

5 DETECTING PEER MIGRATION WITH SHARED TRACKERS

We first consider the migration detection with shared trackers. Assume Torrents 1 and 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 now show that it indeed exists and is not uncommon.

Our observation starts from the fact that the latest BitTorrent metainfo file can include multiple tracker sites stored in the *announce-list* section [17]. This multitracker configuration allows peers to connect to more than one tracker at the same time, which brings two tangible benefits: 1) better accommodates tracker failures, and 2) balances load among the trackers. Fig. 7 offers an example with the multitracker configuration, where Torrent 1 is managed by

5. For ease of exposition, we will focus on the scenario that the migrating peer remains in only one another torrent. Our solution, however, can be easily extended to the scenario that the peer remains in more than one torrent.

TABLE 2
Top 10 Most Popular Trackers

Rank	AS#	Torrents	Tracker Sites (URLs)
1	21202	23386	denis.stalker.h3q.com
2	43350	19915	open.tracker.thepiratebay.org
3	43350	16724	trackeri.rarbg.com
4	21202	15308	tpb.tracker.thepiratebay.org
5	21202	12821	vip.tracker.thepiratebay.org
6	21202	12117	vtv.tracker.thepiratebay.org
7	21202	10019	eztv.tracker.prq.to
8	21202	6079	tracker.prq.to
9	4134	3845	bt1.btally.net
10	15497	3197	inferno.demonoid.com

both trackers A and B , and Torrent 2 is managed both by trackers B and C . In this case, if there is a BT peer x migrating from Torrents 1 to 2, tracker B will receive the arrival message of peer x twice with different content identifications (one arrival message for each torrent). Therefore, tracker B can actually be aware of any peer migration between Torrents 1 and 2 without any tracker-level collaboration.

The question now becomes 1) how popular is the multitracker configuration in the real world? and 2) how many migrations can be detected by this configuration in practice? To answer the first question, we consider all the 1,192 trackers in our measurement. We record the announce list of the torrents in our data set, and show the cumulative distribution of the trackers that have been used in Fig. 6. It indicates that more than 90 percent torrents have specified at least two trackers, and a few torrents even have announce lists of multihundred trackers. This is much higher than an earlier measurement in 2007 [16] (observed multitrackers in 35 percent of the torrents), and thus suggests the multitracker configuration has been quickly recognized and deployed in the BitTorrent community.

To answer the second question, we model the relationships among different torrents as two $n \times n$ matrixes, M_1 and M_2 , where n is the number of torrents in the whole

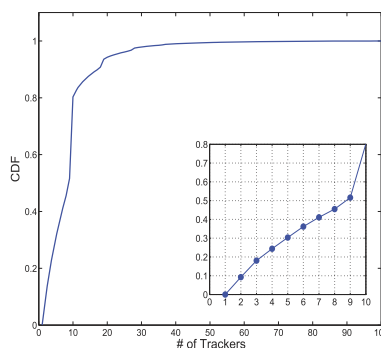


Fig. 6. Number of trackers used by torrents.

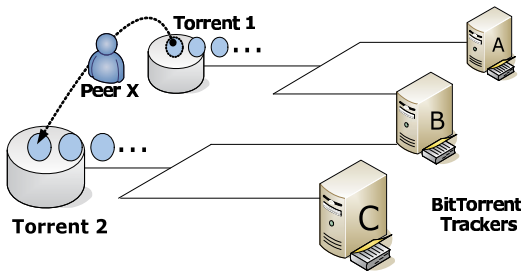


Fig. 7. Peer migration in the shared tracker environment.

system. Each component of M_1 , $M_1^{i,j}$ is of a binary value, indicating whether torrents i and j have at least one common tracker (1-Yes, 0-No); similarly, each component of M_2 , $M_2^{i,j}$ indicates whether torrents i and j share at least one migrating peer.

It is easy to verify that a dot product between these two matrixes, $M_3 = M_1 \cdot M_2$, gives the detectable migrations by the shared tracker approach. Specifically, $M_3^{i,j} = 0$ indicates that peer migrations between torrents i and j are either undetectable or do not exist at all; otherwise, the migrations between these two torrents will be detected even when $M_2(i, j) > 1$. In our measured data, matrix M_3 has 2,538 nonzero entities, where M_2 has 5,707. Therefore, the peer migrations among about 45 percent torrents can be detected with shared trackers.

Once detected, the shared tracker can then use the biased neighbor selection [13] to improve the P2P locality. It may also forward the migration information to other trackers; however, this collaboration is not compulsory in our framework.

6 TORRENT CLUSTERING: AN ENHANCEMENT TO DETENTE PEER MIGRATION

While the 45 percent detection rate is encouraging, particularly considering that it involves no cooperation overhead, there remain a significant number of migrations to be discovered and utilized. We next present *torrent clustering*, a practical enhancement that further improves the detection rate and therefore the effectiveness of locality across torrents.

Our enhancement is motivated by the observation that the migrations are not totally random, but follow certain patterns with strong correlations. This is quite evident from a graph visualization of M_2 in Fig. 8 (with 400 sampled nodes), where noticeable clusters (A-E) exist (each node in the graph refers to a torrent; each edge indicates the peer migration among two torrents). We further quantify this by evaluating the graph's *clustering coefficient*.⁶ For our measurement data, we find that the clustering coefficient is over 0.27, which is quite high as compared with that of random graphs (nearly 0). This value is also very close to the clustering coefficients in some typical social graphs [29], [30]. As such, if the torrents of the same cluster can be organized and managed by the same tracker, we can naturally expect a

6. The clustering coefficient of node i is the fraction of all possible edges between neighbors of i that are present, while the clustering coefficient of a graph is the average of the coefficient across all nodes [28].

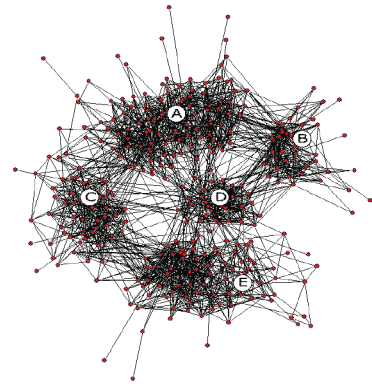


Fig. 8. A sample graph visualization of M_2 .

high detection rate of migration. Note that if all swarms can possibly be managed by one tracker (ignore the tracker capacity), the tracker/torrent availability will become a critical issue to the system. Fortunately, this very big torrent cluster can also be managed by more than one tracker based on the BitTorrent's multitracker protocol. Therefore, we believe that the availability problem can be well addressed if we carefully assign trackers to Internet torrents.

The current BitTorrent implementation does not specify how a torrent maker/user should select a tracker. In most cases, the maker just manually grabs a list of available trackers from certain tracker sites and chooses some of them to serve its torrent. The process is cumbersome, and more importantly, the quality of such a manual configuration depends highly on the knowledge and experience of the makers. Our measurement shows more than 30 percent metainfo files have cited same tracker sites multiple times in their announce lists for no reason. Such a misconfiguration can neither optimize the availability nor the workload distribution of the torrents, but potentially lead to the swarm splitting problem as discussed in [16] and [14].

Considering this, our torrent clustering introduces an automatic process that simplifies and optimizes tracker selection. It will add a periodically updated configuration file to BT clients. This configuration file contains the relation mapping among the trackers and torrents. When a BT user is to create a torrent for a given content, a preferred announce list to the metainfo file will be automatically assigned according to the mapping, which then directs the torrent to the trackers that are serving related torrents.

The relation among the torrents can be extracted from distributed machine learning algorithms [31]. Yet we have found that the content size can serve as a good practical hint. Figs. 9 and 10 show the content size changes between different pairs of torrents. In particular, for any pairs of torrents in our data set (for example torrents a and b). Assume that the content size of torrent a is $S(a)$ MB and the content size of torrent b is $S(b)$ MB. We use $|S(a) - S(b)|$ to refer the absolute content size change between these two torrents. In Figs. 9 and 10, each point on the y-axis refers to a migration (and edge in M_2), and each point on the x-axis refers to the absolute content size change of the migration. Fig. 9 shows the absolute content size change between some highly related torrents (with 5 or more peers migrating between them), and Fig. 10 shows the absolute content size

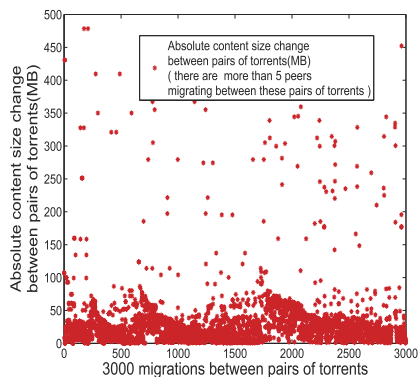


Fig. 9. Content size change between pair of torrents (more than 5 peers are detected to migrate between these torrents).

change between the torrents with only 1 migrating peer. It is easy to see that the peers are more likely to migrate across the torrents that have similar content size. This is further quantified in Fig. 12, which shows the cumulative distribution of *relative size changes* between torrents with peer migrations.

$$R_{t_i, t_j} = \frac{|C_{t_i} - C_{t_j}|}{|C_{t_i} + C_{t_j}|/2},$$

where C_t indicates the content size of torrent t . We can see that 70 percent of tightly related torrents (e.g., pairs experienced over 10 peer migrations) have relative size changes below 10 percent, which is much smaller as compared to those weakly related torrents (relative changes span to 2). This is likely because a peer of 10 migrates among torrents of the same type of contents, e.g., movies, CDs, or softwares, whose data sizes are generally close.

We have also validated the effectiveness of torrent clustering with this simple hint through a trace-driven simulation. In particular, we clustered the real-world torrents based on the size of their sharing contents (the relationship between torrents and trackers will be changed after this clustering process) and see if the peer migrations can be better detected by the trackers. Fig. 13 shows that more than 75 percent peer migrations can be detected if the maximum content size change is less than 70 MB within each torrent cluster. This ratio is reasonably good for practical use. We are currently working on incorporating

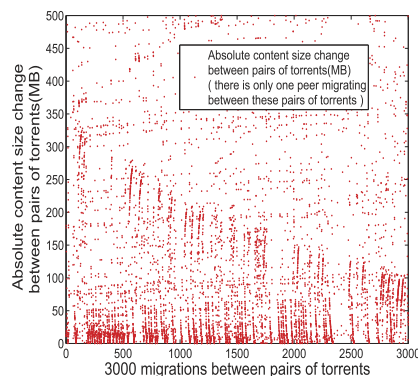


Fig. 10. Content size change between pair of torrents (only 1 peer is detected to migrate between these torrents).

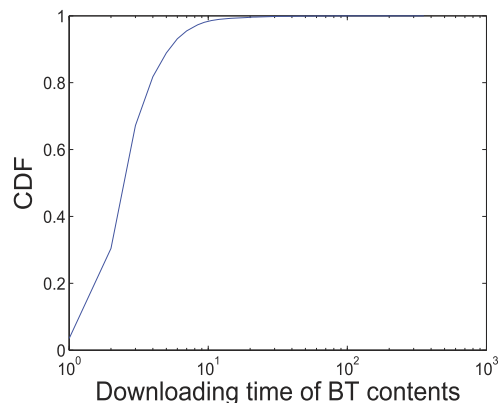


Fig. 11. Downloading time of BT contents.

other torrent features and more advanced learning tools, in particular, the restricted Boltzmann machine [32] and deep autoencoder [33], to possibly achieve better detection rate.

It is worth noting that our solution works well with progressive partial deployment, because adding extra trackers to torrents' announce list will not affect the availability of the existing metainfo file. The torrent clustering would also bring other benefits, e.g., content-aware searching and customized QoS for distributing multimedia files, though a complete discussion is out the scope of this paper.

7 DISCUSSIONS

This study takes a first step toward investigating a multiple-torrent-based traffic locality. There are many research issues that can be further explored.

First, what if the BT peers simply remove their old contents? In fact, this problem is seldom discussed in the existing studies because its measurement is related to users' privacy. We are currently reinvestigating the online behavior (as well as the offline duration) of BT peers. Based on our preliminary measurement results, we find that, unless the BT users always remove their downloaded contents very soon after the downloading (generally within 2 hours), the multiple-torrent-based sharing will remain efficient for enhancing local content sharing. As shown in Fig. 11, we can see that most (70 percent) BT contents are downloaded within 2-3 hours. Considering the flash crowd arrival of BT peers [6], it is reasonable to believe that the peers are not necessary to hold their old contents for a longtime since very few peers will join the swarm after the flash crowd period. It

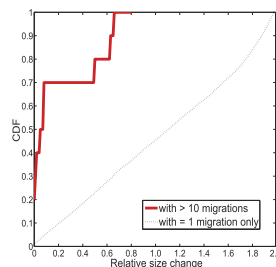


Fig. 12. Relative size change.

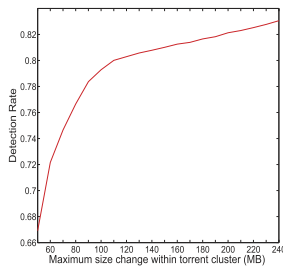


Fig. 13. Detection rate.

is worth noting that most BT contents, especially video contents, are relatively large, and the users may not simply remove these contents very soon after the downloading.

Second, incentive is also an important concern. Many studies have investigated the sharing incentive in P2P systems and for the multiple torrent environment as well. We believe our framework can apply similar incentive mechanisms as proposed in [21] and [34]. For example, one possible approach is to let peers decide to whom to send content based on the rate offered by their neighbors, irrespective of the swarms in which they are involved in. We are also trying to address the incentive problem based on social relationships among BT peers. We find that the downloading of many torrents are initialized among friends through social network applications such as Twitter. We believe that the sharing incentive as well as the downloading performance can be significantly improved in these trusted peer communities.

8 PERFORMANCE EVALUATION

We now evaluate the performance of our locality mechanism in the multiple torrent environment. We also compare it with other state-of-the-art locality solutions; in particular, the biased neighbor selection for individual torrents [13], [27]. To achieve a fair comparison and also to examine the diverse factors that would affect their performance, we also use the discrete-event BitTorrent simulator developed by Stanford University [35] as [13] did; we summarize the key network settings as follows (more configuration details can be found in [13]).

All peers inside the ISPs are modeled after cable modem and DSL nodes, and have asymmetric upload/download bandwidth. The upload bandwidth of these peers is 100 kbps and downloading bandwidth is 1 Mbps. Considering the peer arrival/departure, most peers are joining the network at once, i.e., the flash crowd scenario. We focus on this feature since it is the most challenging for ISPs to handle. For each torrent, there is one original seeder that will always stay online (with 400 Kbps uplink bandwidth), and other peers (except for the migrating peers) will leave the BT network forever as soon as they finish downloading. This is in accordance with the measurements because only 85 percent peers are participating in multiple torrents.

For the multiple torrent scenario, we assume that 1,000 peer migrations occur during a 48-hour simulation, which is consistent with the data in Fig. 4. We then evaluate the locality performance with different peer distributions and migration detection rate. The downloaded content of

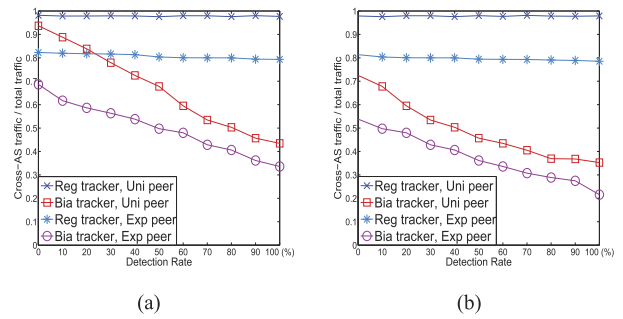


Fig. 14. Percentage of cross-AS traffic, with regular/biased tracker and uniform/exponential peer distribution. (a) Small torrent with 100 initial peers. (b) Large torrent with 600 initial peers.

detected peers will be recovered for locality. These extra peers however will not simply serve as selfless seeders, but rather normal peers that expect data, albeit from other related torrents through a cross-torrent credit approach [34]. This will eliminate biases related to seeding incentives, which remains an open problem in the existing BitTorrent networks.

We will focus on two metrics: cross-AS traffic and downloading completion time of peers, which reflect the potential benefit and impact of P2P locality, respectively.

We first calculate the percentage of the cross-AS traffic over the total downloading/uploading traffic of the peers in different torrents in the multiple torrent environment. Figs. 14a and 14b show the results of two typical torrents. The first is a relatively small torrent with 100 initial peers, and the second is a large torrent with 600 initial peers. With regular unmodified trackers, we can see that the cross-AS traffic is quite high (over 95 percent) when the peers are uniformly distributed among the ASes; for the exponential peer distribution, the cross-AS traffic is relatively lower, implying that certain peer localities have been naturally utilized. Even though this exponential peer distribution is more realistic as validated in our earlier measurement (Figs. 1 and 2), the regular trackers do not take full advantage of the localities, and hence the cross-AS traffic remains high (around 80 percent). On the other hand, the biased tracker design prioritizes local peers for sharing, which, as shown in Figs. 14a and 14b, significantly reduces the cross-AS traffic. This is particularly true for larger torrents that enable more opportunities for local connections.

Note that, when the detection rate is 0, the multiple torrent setting degenerates to a single torrent setting with no previously downloaded content being recovered from migrating peers. In this case, the cross-AS traffic is the highest in the figures. With biased trackers, the percentage of cross-AS traffic is also decreasing with the increase of migration detection rate. This suggests that the combination of locality and multiple torrent is quite effective in reducing cross-AS traffic. Recall that, for detection with shared tracker only, we have a success rate of 45 percent (see Section 5), which translates into percentages of cross-AS traffic of roughly 50 percent and 35 percent for the 100-peer and 600-peer torrents, respectively. And the amount will be further reduced to about 42 percent and 28 percent with torrent clustering, which are only half of those with regular trackers. Even for uniform peer distribution, the traffic

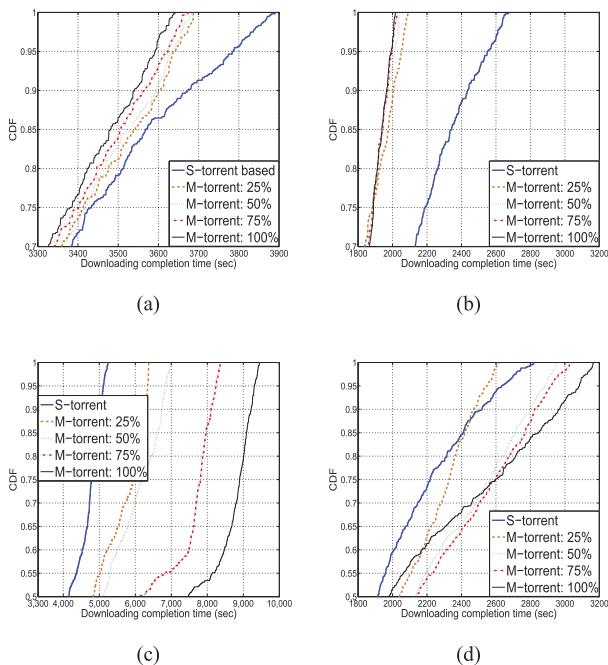


Fig. 15. Downloading completion time of 600 peers. (a) Uniform distribution with regular tracker. (b) Uniform distribution with biased tracker. (c) Exponential distribution with regular tracker. (d) Exponential distribution with biased tracker.

reduction is still remarkable, suggesting the necessity for exploring locality.

We next examine whether the reshaping of the traffic will affect user experience; in particular, whether it will slow down the peer completion time. Figs. 15a to 15d present the cumulative distribution of the downloading completion time with different peer distributions and torrent-tracker combinations. In the figures, we use *M-torrent* and *S-torrent* to represent the multiple-torrent-based and single-torrent-based solutions, respectively (the percentage values refer to the possible detection rate of peers' migration behavior). We show the results of the larger torrent with 600 peers, and we have observed similar curves for torrents of other sizes.

We first look at the case of peers uniformly distributed among ASes, as shown in Fig. 15a. Surprisingly, although no extra peers will serve as selfless seeders, the downloading completion time of the peers is still improved by the multiple torrent approach. Moreover, as shown in Fig. 15b, all peers will finish their downloading within 2,700 sec in the individual locality torrent; this completion time will be further improved to 2,100 sec with the proposed multiple torrent-based locality. Note that the peers are assumed to be uniformly distributed among different ASes, all ASes, therefore, have enough local resources to utilize. Intuitively, potential benefits can be obtained by accessing these local peers.

However, for the exponential peer distribution (a more realistic case yet seldom been discussed in the previous studies), the downloading completion times of most peers are increased as shown in Fig. 15c. In particular, if the peers are connected to regular trackers, the multiple torrent-based approach will slow down the downloading completion time of all peers significantly. The peers' downloading completion time is almost doubled when the detection rate reaches

TABLE 3
Downloading Completion Time with Different Migration Detection Rate (Regular Tracker)

Rate	Max	Min	Median	Mean	Std
	$\times 10^3$ sec				
0%	5.25	0.19	4.15	3.75	1.29
20%	5.97	0.18	4.68	4.25	1.60
40%	6.38	0.17	4.83	4.37	1.60
60%	6.96	0.20	5.87	4.53	1.90
80%	8.37	0.17	6.13	5.52	2.51
100%	9.45	0.19	7.47	6.41	2.87

to 100 percent. This result shows that the exponential peer distribution across the ASes will potentially reduce peers' downloading experience with an increase of torrents' population. An intuitive explanation is that the flash crowd of peers as well as the trackers' random peer selection will put more pressure to the cross-ISP links and unavailable cause link overload (especially for the most popular ASes). Moreover, we have also observed that a great number of peers in the most popular ASes have very close downloading completion time (also leave the BT networks at similar time). Their departure will also reduce downloading performance of other peers in the BitTorrent system. Fortunately, as shown in Fig. 15d, the biased trackers can well address such a problem and peers' completion times only slightly increase. Note that, to clarify the possible degradation of the downloading performance, we have ignored the first quartile (25th percentile) of the CDF where the lines are too close to each other. The detailed data can be found in Tables 3 and 4.

For easy comparison, we also show the completion times of the four typical torrent-tracker combinations in Fig. 16, where the peers are sorted in ascending order of their downloading completion time (the detection rate of *M-torrent* is set to 100 percent). It clearly shows that the combination of locality and multiple torrent will minimize the impact to the peer downloading experiences.

9 CONCLUSIONS AND FUTURE WORKS

In this paper, we for the first time investigated the existence and distribution of peer locality across different ASes

TABLE 4
Downloading Completion Time with Different Migration Detection Rate (Biased Tracker)

Rate	Max	Min	Median	Mean	Std
	$\times 10^3$ sec				
0%	2.81	0.25	1.91	1.91	0.47
20%	2.53	0.20	1.93	1.89	0.42
40%	2.61	0.30	2.04	1.98	0.44
60%	2.95	0.36	2.13	2.10	0.55
80%	3.04	0.25	2.14	2.13	0.55
100%	3.16	0.31	1.97	2.08	0.61

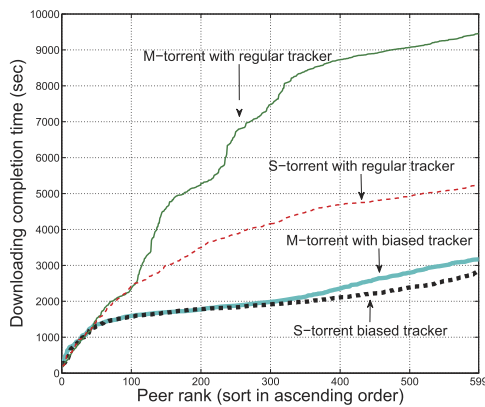


Fig. 16. Comparison of downloading completion time.

through a large-scale hybrid PlanetLab-Internet measurement. We found that the BitTorrent peers do exhibit strong geographical locality. However, the effectiveness of a locality mechanism can be quite limited when focusing on individual torrents, given that very few torrents are able to form large enough local clusters.

Inspired by the multiple torrent nature of many peers, we proposed a novel framework that traces and extracts the available contents at peers across multiple torrents, thus effectively improving the locality. A series of key design issues were addressed in this framework; in particular, the detection of peer migration across the torrents. Since we can hardly expect to organize the public Internet trackers together for detection, we developed a smart detection mechanism with shared trackers, which incurs no extra communication overhead. It was further enhanced through a torrent clustering approach that explores peer migration patterns.

The performance of multiple torrent-based locality was evaluated through extensive trace-driven simulations. Compared to the locality for individual torrents, our solution has successfully promoted the local content availability, thus significantly reducing cross-AS traffic and yet keeping minimal impact to peers' downloading experiences. Our contributions are listed as follows:

1. To understand the potentials and difficulties of applying the locality mechanisms, we examined the global distribution of BitTorrent peers in the Internet Autonomous Systems via a real-world measurement.
2. We found that even in the most popular Autonomous Systems, very few individual torrents are able to form large enough local clusters of peers, making state-of-the-art locality mechanisms for individual torrents quite inefficient.
3. We developed a novel framework that traces and recovers the available contents at peers across multiple torrents, and thus effectively amplifies the possibilities of local sharing.

As a future work, we are particularly interested in a better understanding and unitization of the peer migration patterns. Moreover, the geographical locality is also an open issue as discussed in [4], especially when we consider the relationships between different ASes/ISPs. Solutions to

these problems will certainly help us further improve the system performance.

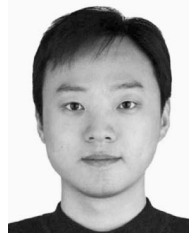
ACKNOWLEDGMENTS

This work was supported by a Canada NSERC Discovery Grant and a Discovery Accelerator Supplements (DAS) Grant.

REFERENCES

- [1] CacheLogic, <http://www.cachelogic.com/>, 2011.
- [2] M. Dischinger, A. Mislove, A. Haeberlen, and K.P. Gummadi, "Detect Bittorrent Blocking," *Proc. Eighth ACM SIGCOMM Conf. Internet Measurement (IMC)*, 2008.
- [3] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should Internet Service Providers Fear Peer-Assisted Content Distribution?," *Proc. Fifth ACM SIGCOMM Conf. Internet Measurement (IMC)*, 2005.
- [4] J.S. Otto, M.A. Sanchez, D.R. Choffnes, F.E. Bustamante, and G. Siganos, "On Blind Mice and the Elephant: Understanding the Network Impact of a Large Distributed System," *Proc. ACM SIGCOMM*, 2011.
- [5] Planetlab, <http://www.planet-lab.org/>, 2011.
- [6] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, Analysis, and Modeling of BitTorrent-Like Systems," *Proc. Fifth ACM SIGCOMM Conf. Internet Measurement (IMC)*, 2005.
- [7] The BitTorrent Protocol Specification, http://www.bittorrent.org/beps/bep_0003.html, 2011.
- [8] A. Logout, N. Liogkas, E. Kohler, and L. Zhang, "Clustering and Sharing Incentives in BitTorrent Systems," *Proc. ACM Conf. Measurement and Modeling of Computer Systems (SIGMATERICS)*, 2007.
- [9] D. Qiu and R. Srikant, "Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks," *Proc. ACM SIGCOMM*, 2004.
- [10] S.L. Blond, A. Legout, and W. Dabbous, "Pushing BitTorrent Locality to the Limit," technical report, INRIA, 2008.
- [11] H. Xie, R.Y. Yang, A. Krishnamurthy, Y.G. Liu, and A. Silberschatz, "P4P: Provider Portal for Applications," *Proc. ACM SIGCOMM*, 2008.
- [12] 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.
- [13] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang, "Improving Traffic Locality in BitTorrent via Biased Neighbor Selection," *Proc. IEEE 26th Int'l Conf. Distributed Computing Systems (ICDCS)*, 2006.
- [14] G. Dan and N. Carlsson, "Dynamic Swarm Management for Improved BitTorrent Performance," *Proc. Eighth Int'l Conf. Peer-to-Peer Systems (IPTPS)*, 2009.
- [15] M. Piatek, T. Isdal, A. Krishnamurth, and T. Anderson, "One Hop Reputations for Peer to Peer File Sharing Workloads," *Proc. Fifth USENIX Symp. Networked Systems Design and Implementation (NSDI)*, 2008.
- [16] G. Neglia, G. Reina, H. Zhang, D. Towsley, A. Venkataramani, and J. Danaher, "Availability in BitTorrent Systems," *Proc. IEEE INFOCOM*, 2007.
- [17] BitTorrent Multi-Tracker Specification, <http://www.bittornado.com/docs/multitracker-spec.txt>, 2011.
- [18] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, and H.J. Sips, "The Bittorrent P2P File-Sharing System: Measurements and Analysis," *Proc. Fourth Int'l Workshop Peer-to-Peer Systems (IPTPS)*, 2005.
- [19] D.S. Menasche, A.A.A. Rocha, B. Li, D. Towsley, and A. Venkataramani, "Content Availability and Bundling in Swarming Systems," *Proc. Fifth Int'l Conf. Emerging Networking Experiments and Technologies (CoNext)*, 2009.
- [20] J. Han, T. Chung, S. Kim, H. Kim, T. Kwon, and Y. Choi, "An Empirical Study on Content Bundling in BitTorrent Swarming System," *AsiaFI Summer School*, arXiv:1008.2574v1, 2010.
- [21] N. Lev-tov, N. Carlsson, Z. Li, C. Williamson, and S. Zhang, "Dynamic File-Selection Policies for Bundling in BitTorrent-Like Systems," *Proc. 18th Int'l Workshop Quality of Service (IWQoS)*, 2010.

- [22] P. Dhungel, D. Wu, Z. Liu, and K. Ross, "BitTorrent Darknets," *Proc. IEEE INFOCOM*, 2010.
- [23] B. Fan, J.C. Lui, and D. Chiu, "The Design Tradeoffs of BitTorrent-Like File Sharing Protocols," *IEEE Trans. Networking*, vol. 17, no. 2, pp. 365-376, Apr. 2009.
- [24] M. Piatek, H.V. Madhyastha, J.P. John, A. Krishnamurth, and T. Anderson, "Pitfalls for ISP-Friendly P2P Design," *Proc. Eighth ACM Workshop Hot Topics in Networks (HOTNETS)*, 2009.
- [25] R. Cuevas, N. Laoutaris, X. Yang, G. Siganos, and P. Rodriguez, "Deep Diving into BitTorrent Locality," *Proc. IEEE INFOCOM*, 2011.
- [26] Ctorrent, <http://ctorrent.sourceforge.net/>, 2011.
- [27] B. Liu, Y. Cui, Y. Lu, and Y. Xue, "Locality-Awareness in BitTorrent-Like P2P Applications," *IEEE Trans. Multimedia*, vol. 11, no. 3, pp. 361-371, Apr. 2009.
- [28] D. Watts and S. Strogatz, "Collective Dynamics of Small-World Networks," *Nature*, vol. 393, no. 6684, p. 409, 1998.
- [29] A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," *Proc. Seventh ACM SIGCOMM Conf. Internet Measurement (IMC)*, 2007.
- [30] X. Cheng and J. Liu, "NetTube: Exploring Social Networks for Peer-to-Peer Short Video Sharing," *Proc. IEEE INFOCOM*, 2009.
- [31] R. Agrawal, T. Imieliski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," *Proc. ACM Int'l Conf. Management of Data (SIGMOD)*, 1993.
- [32] R. Salakhutdinov and G. Hinton, "Semantic Hashing," *Int'l J. Approximate Reasoning*, vol. 50, pp. 969-978, Dec. 2008.
- [33] G.E. Hinton and R.R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504-507, July 2006.
- [34] Y. Yang, A.L.H. Chow, and L. Golubchik, "Multi-Torrent: A Performance Study," *Proc. IEEE Int'l Symp. Modeling, Analysis and Simulation of Computers and Telecomm. Systems (MASCOTS)*, 2008.
- [35] BT-SIM, <http://theory.stanford.edu/~cao/btsim-code.tgz>, 2011.
- [36] H. Wang and J. Liu, "Modeling and Improving the ISP-level Incentive of Deploying P2P Locality," Technical Report, School of Computing Science, Simon Fraser Univ., 2009.



Haiyang Wang is currently working toward the PhD degree in the School of Computing Science, Simon Fraser University, British Columbia, Canada. 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. He is a student member of the IEEE.



Jiangchuan Liu (S'01-M'03-SM'08) is currently working as 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 TPC vice chair for Information Systems of IEEE INFOCOM'2011. He is a senior member of the IEEE and a member of Sigma Xi. He is an associate editor of *IEEE Transactions on Multimedia*, an editor of *IEEE Communications Surveys and Tutorials*, and an area editor of *Computer Communications*.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.