Torrents on Twitter: Explore Long-Term Social Relationships in Peer-to-Peer Systems

Haiyang Wang, *Student Member, IEEE*, Feng Wang, *Member, IEEE*, Jiangchuan Liu, *Senior Member, IEEE*, Ke Xu, *Senior Member, IEEE*, and Di Wu, *Member, IEEE*

***Abstract*—Peer-to-peer file sharing systems, most notably BitTorrent (BT), have achieved tremendous success among Internet users. Recent studies suggest that the long-term relationships among BT peers can be explored to enhance the downloading performance; for example, for re-sharing previously downloaded contents or for effectively collaborating among the peers. However, whether such relationships do exist in real world remains unclear. In this paper, we take a first step towards the real-world applicability of peers' long-term relationship through a measurement based study. We find that 95% peers cannot even meet each other again in the BT networks; therefore, most peers can hardly be organized for further cooperation. This result contradicts to the conventional understanding based on the observed daily arrival pattern in peer-to-peer networks. To better understand this, we revisit the arrival of BT peers as well as their long-range dependence. We find that the peers' arrival patterns are highly diverse; only a limited number of stable peers have clear self-similar and periodic daily arrivals patterns. The arrivals of most peers are, however, quite random with little evidence of long-range dependence. To better utilize these stable peers, we start to explore peers' long-term relationships in specific swarms instead of conventional BT networks. Fortunately, we find that the peers in Twitter-initialized torrents have stronger temporal locality, thus offering great opportunity for improving their degree of sharing. Our PlanetLab experiments further indicate that the incorporation of social relations remarkably accelerates the download completion time. The improvement remains noticeable even in a hybrid system with a small set of social friends only.**

***Index Terms*—BitTorrent, long-term relationship, self-similar, social networks.**

I. Introduction

PEER-TO-PEER (P2P) networks have emerged as a successful architecture for content sharing over the Internet. BitTorrent (BT), the most popular P2P application, has attracted significant attention from network operators and researchers for its wide deployment. Recent studies suggest that the long-term relationships among BT peers can be explored to enhance the downloading performance; for example, the re-sharing of old contents contents[1][2]. The cooperation among closely related peers has also been explored to achieve better sharing efficiency[3][4]. However, whether such long-term relationships can be maintained still remain unknown.

In this paper, we for the first time examine the challenges and potentials of long-term social relationships in P2P networks, particularly *Twitter-trigger BT swarms* (The swarms whose downloads are initialized/shared in Twitter communities). We have collected trace-data from more $100,000$ real world swarms spanning over 80 days. We find that peers' online patterns in conventional BT swarms are highly diverse: less than 5% peers can meet each other again in our entire measurement duration [1]. In particular, their online patterns are not well-overlapped to provide constant help to each other. This observation raises a big challenge to utilize social relations because the peers may not even have the chance to help their friends.

This result contradicts to the conventional understanding based on the observed daily arrival pattern in peer-to-peer networks[2]. To better understand this, we revisit the individual peer arrivals as well as their long-range dependence. Surprisingly, we find that only a limited number of peers have very stable self-similar and periodic daily arrival patterns (which we call them "*stable peers*"). It is worth noting that the online patterns of these stable peers have strong long-range dependence and can be well predicted by their historical behaviors. The arrivals of most peers are, however, quite random with a clear absence of long-range dependence. Therefore, even if we can extend the standard BT protocol beyond stand-alone file swarming, the long-term relationships can hardly be built among most peers.

To better utilize these stable peers, we start to explore peers' long-term relationships in specific swarms instead of conventional BT networks. Fortunately, we find that the peers' online patterns are better overlapped in *Twitter-trigger BT swarms*, where more than 35% peers can meet each other again. Our PlanetLab experiments indicate that the incorporation of social relations remarkably accelerates the downloading time for BT peers. This improvement remains noticeable even in a hybrid system with a small set of social friends only.

[1]Even when we consider their cooperation across multiple torrents.

[2]This pattern indicants that most peers are very likely to be online at the same time everyday; it is thus hard to understand why 95% peers can only meet each other once.

Manuscript received February 3, 2012; revised July 14, 2012. The associate editors coordinating the review of this paper and approving it for publication were B. Lin, J. Xu, and P. Sinha.

H. Wang is with the School of Computing Science, Simon Fraser University, British Columbia, Canada (e-mail: hwa17@cs.sfu.ca).

F. Wang is with the Department of Computer and Information Science, The University of Mississippi, MS, USA (e-mail: fwang@cs.olemiss.edu).

J. Liu is with the School of Computing Science, Dalian University of Technology, Dalian, China, and the School of Computing Science, Simon Fraser University, British Columbia, Canada (e-mail: jcliu@cs.sfu.ca).

K. Xu is with the Department of Computing Science, Tsinghua University, Beijing, China (e-mail: xuke@csnet1.cs.tsinghua.edu.cn).

D. Wu is with the School of Computing Science, Dalian University of Technology, Dalian, China (e-mail: wudi@dlut.edu.cn).

Digital Object Identifier 10.1109/TNSM.2012.091912.120243

The rest of this paper is organized as follows: In Section 2, we illustrate the related works. In Section 3, we present the measurement methodology and analyze the pitfalls of long-term relationships. To better understand this challenge, Section 4 quantifies the online overlap of peers, and Section 5 gives the underlying reasons of the results. Section 6 sheds new light into the torrents sharing on social networks applications, and Section 7 provides the experimental evaluation to our practical cooperation protocol. After the further discussion in Section 8, Section 9 concludes the paper.

II. Related Works

There have been numerous studies on the implementation, analysis, and optimization of the BitTorrent system [5]. As a well known long-term relationship in BitTorrent, multiple-torrent-based studies have recently attracted attention following an earlier work of Guo *et al* [1]. They revealed that more than 85% of all peers participate in multiple torrents, and proposed an inter-torrent approach through tracker-level collaborations. Dan *et al.* [2] further investigated how the separated torrents can be merged together to improve the performance of an entire torrent. Piatek *et al.* [6] identified the performance problems when the BT publishers package a number of related files and disseminates it through a single larger swarm (*content bundling*). This study also designed a new one-hop reputation protocol for BitTorrent system. Menasche *et al.* [7] further quantified the content availability in swarming systems of content bundling. Lev-tov *et al.* [8] discussed the details of file selection and applied a stochastic games and Markov model to optimize the BitTorrent system with multiple files.

On the other hand, it also has been realized that the existing BT system hinders decent peers or peers of close relationship from more efficient cooperation. Therefore, the peer cooperation potential in private [4] [9] or even some public swarms [3] have been widely suggested. Long-term relationships are also considered to improve the availability [1] as well as the sharing efficiency [10] in BT systems. Moreover, the existence of peer communities [11] raises an opportunity to organize some peers together for further cooperation.

Differing from the existing studies, we take a first step towards the real-world applicability of peers' long-term relationship through a measurement based analysis. We shed new light on the arrival pattern of the peers and some interesting findings are discussed.

III. Pitfalls of Long-term Relationship among BitTorrent Peers

The measurement of real world BT systems is a challenging job as discussed in previous studies such as [12]. In order to obtain more detailed information, we carefully collect the data from a local ISP through both passive traffic monitoring and active swarm probing approaches as follows:

First, we collect over $100,000$ torrent files from a popular torrent sharing site `www.btmon.com`. To learn peers' online behaviors in these swarms, we passively monitored the BitTorrent traffic on the out-going switch of a local ISP from Oct. 2009 to Jan. 2010, for over 80 days. In particular, we generate a tracker list based on the collected torrent files, which result in 683 active trackers. According to this tracker list, we obtain the updating message between these trackers and the peers that are located in this ISP[3]

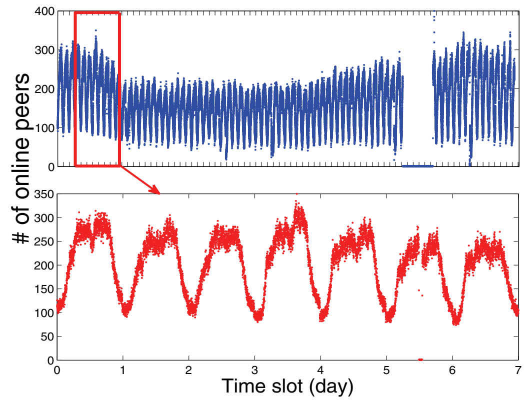


Fig. 1: # of online peers in the measurement.

To obtain more complete peer information in these swarms (because most of their peers may not belong to this ISP), we actively probe the peer information from Planetlab [13] nodes. We run a modified version of cTorrent [14] on over 250 PlanetLab nodes across these torrent files. These Planetlab clients actively joined the torrents and recorded the observed peer information (more details can be found in [15]). In this way, we successfully detected the IP addresses of over 95% peers for most of the swarms[4].

We will now examine the long-term relationships among BitTorrent peers. We strive to clarify: '*Can long-term relationships be built across peers in conventional BT networks?*'

We define K as the set of all the trackers, and thus $|K| = 683$. We first collect the online information of the peers from all the trackers. Each tracker k generates a peer availability matrix A_k that indicates the online time slots of the peers: Each component of A_k, $A_k(i,j)$ is a binary value, indicating whether peer i is connected to tracker k at time slot j (1-yes, 0-no). In our measurement, the maximum value of i is $43,360$ and the maximum value of j is $120,000$ minutes. After that, we merge all 683 matrixes together to get a global online matrix G. Each component of G, $G(i,j)$ is a binary value indicating whether peer i is connected to the BT networks at time slot j. (1-yes, 0-no). As shown in Figure 4, the peer's online durations are indicated by several solid lines between solid cycles(refer to peer arrivals) and empty cycles(refer to peer departures).

We now check peers' off-line durations in our data set. As shown in Figure 2, over 70% peers will return to the BT networks after more than 10 days since their last downloading. Given most torrents' lifespans are less than 240 hours [1], the old torrents may have been already dead because the users may remove their old contents after such a long time. We further check the number of peer encounters (how many times a peer's online pattern is overlapped with another peer) in Figure 3,

[3]Note that the peer' IP addresses are statically assigned in this ISP.. These peers are filtered/identified by the torrent information in their updating messages.

[4]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.

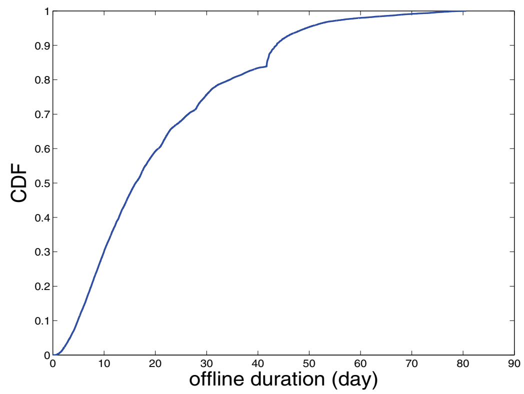


Fig. 2: CDF of peers' off-line duration.

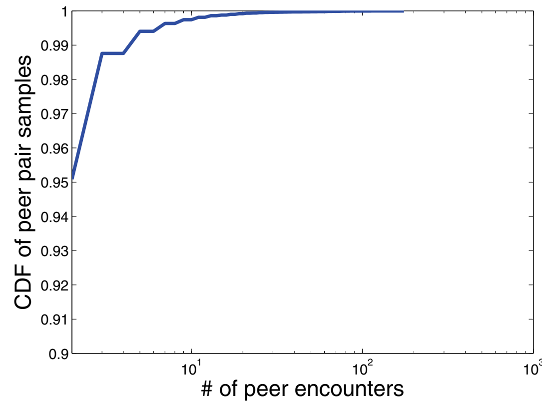


Fig. 3: Number of peer encounters.

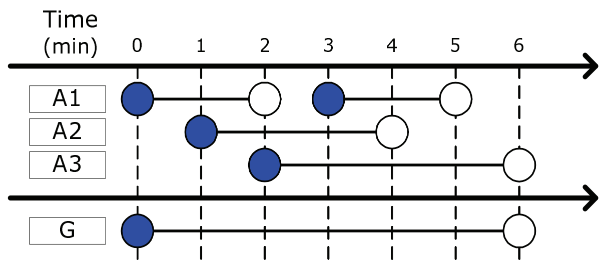


Fig. 4: Computing of G.

where only pairs encountered at least once are considered. We can see that the peers in are not likely to meet others again; less then 5% peers can meet others more than once in the BT networks over 80 days. This raises a significant challenge to utilize long-term relationships because the peers can hardly meet/help their potential friends.

Giving the clear daily patterns (shown in Figure 1), the peers should have a good chance to help others. It is thus necessary to understand why most peers can hardly meet each other. Therefore, we further investigate peers' arrivals in the following section.

IV. Similarity of Peers' Online Pattern

A. Computation of Online Similarity

We are trying to identify what kind of peers are more likely to meet each other in the BitTorrent networks. As we discussed in Section **??**, we can obtain a global online matrix of peers $G = \sum_{k \in K} A_k$, representing the online pattern of $n = 43,360$ peers over $m = 120,000$ minutes. Let $G(i, M)$ denote the ith row of G (where M refers to the set of time slots and $|M| = m$). We have the overlapped time slots of two peers, n_1 and n_2, described as:

$$L_{n_1,n_2} = G(n_1, M) \bullet G(n_2, M).$$

L_{n_1,n_2} is a $1 \times m$ matrix, each component of which L_{n_1,n_2}, $L_{n_1,n_2}(j)$ is also a binary value, indicating whether peer n_1 and n_2 are online at the same time at time slot j. The length (number of online slots) of this overlap over time m can be described as:

$$K(L_{n_1,n_2}) = \sum_{j=1}^{m} L_{n_1,n_2}(j),$$

where $K(L_{n_1,n_2})$ is an integer indicating the length of the online overlap of peer n_1 and n_2. We also use $K(G(n_1, M))$ and $K(G(n_2, M))$ to refer the total online time of peer n_1 and n_2, respectively. Therefore, the time similarity of peer n_1 and n_2, $S(n_1, n_2)$, is defined as follows:

$$S(n_1, n_2) = \frac{K(L_{n_1,n_2})}{\text{Max}\,\{K(G(n_1, M)), K(G(n_2, M))\}}$$

An intuitive explanation of $S(n_1, n_2)$ is the normalized time overlap of peer n_1 and n_2. In particular, considering the possible diversity of peers' total online duration, we use the larger one between n_1 and n_2 for the normalization.

B. Observations

To present all $43,360$ peers, our online similarity matrix has over 1.9×10^{10} components. We therefore present a sample illustration of the similarity among 70 peers in Figure 5, where the intensity indicates the similarity value among the trackers (dark: similar; light: not similar). We can see that most of these components have very low similarity values (generally lower than 0.025). This observation confirms that that the online patterns of most peers are quite different with each other.

To further understand their property, we check two representative peers in Figure 6, where peer $\#313$ is a stable peer with long online duration (total online time larger than 16 hours), and peer $\#312$ is an unstable peer with relatively short online duration (total online time less than 5 hours). We can see that for peer $\#312$, it only has positive similarity with 862 (out of $43,360$) peers; most of their values are quite low (under 0.05) and this value decrease exponentially fast in a very small scale. On the other hand, for peer $\#313$, it has positive similarity with over $40,000$ peers and more than $2,000$ of them have values higher than 0.05. This indicates that peer $\#313$ is more eligible to be optimized by long-term relationships. In Figure 7, we further compare 10 stable peers to 10 unstable peers. It is easy to see that the similarity distributions of these two groups are clearly different. This indicates that the the long-term relationships can be obtained

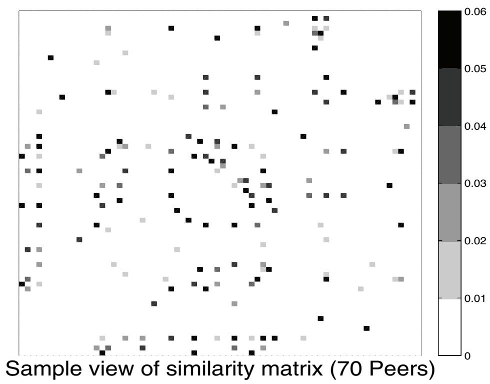


Fig. 5: A simple view of similarity matrix.

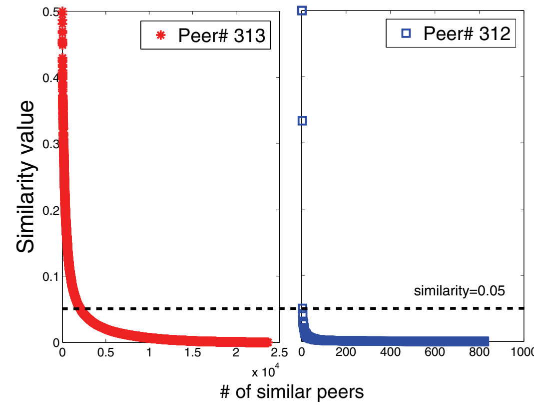


Fig. 6: Similarity distribution of peer #313 and #312.

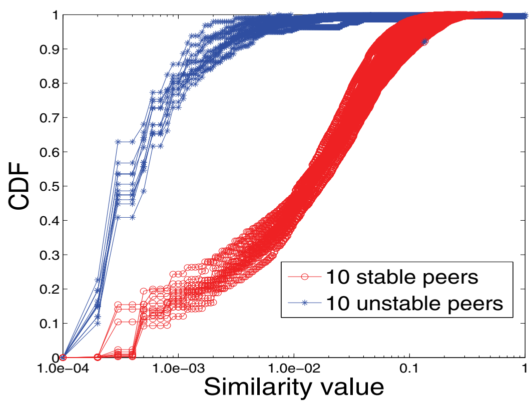


Fig. 7: CDF of similarity in two groups.

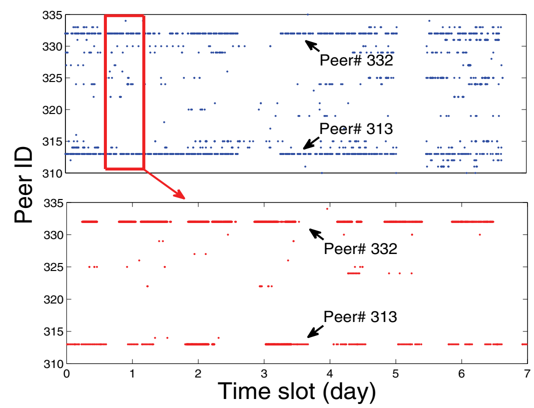


Fig. 8: Two peers with similar online pattern.

among stable peers. It is therefore important to further clarify the online pattern of these peers.

V. Understand the Online Pattern of Peers

A. A Close Look of Individual Peers

We will now examine the online patterns of peers. In particular, we will clarify whether all the BT peers are following similar daily arrival pattern with clear self-similar features[16]. We first take a close look to some individual peers. Figure 8 presents an illustration of two highly overlapping peers in our dataset. It is easy to see that these two peers join the BitTorrent networks regularly every day and their online time slots are highly overlapped following clear 7-day pattern. Other peers, on the other hand, are having quite random behaviors. These peers join the BT networks for several hours, leave, and stay offline for a relatively long time before next arrival.

Based on this observation, we divide the BT peers into two different groups. The first group includes $13,000$ peers that have relatively stable online time. In particular, their total online durations are more than 1% of the entire measurement duration[5]. The other group includes the remaining peers with total online time less than 16 hours. We will further discuss the self-similar as well as the long-range dependence properties in these two groups respectively.

[5]Note that the entire measurement duration is 80 days; 1% means these peers' total online time is more than 16 hours

B. Self-similar and Long-Range Dependence

It is well known that the term "self-similar" was formally defined by Mandelbrot [17] [18]. In the classic studies, the degree of self-similarity can be measured by the estimation of *Hurst parameter*. Where the Hurst parameter H is a measure of the extent of long-range dependence in a given time series. H takes on values from 0 to 1, and the value of 0.5 indicates the absence of long-range dependence. The more H is close to 1, the greater the degree of persistence or long-range dependence is. This value can be inferred generally through graphical methods such as R/S-statistic, variances of the aggregated processes and periodogram-based analysis.

In the recent years, many improved algorithms have been proposed to compute the Hurst parameter. We thus apply one popular way form [19] to our dataset. We divide our daily 24-hour trace into 4 equal time bins corresponding to 2 normal/low hours in the morning and midnight and 2 busy hours around noon. Therefore. each bin has the length of 6 hours. For comparison, we used a simulated Poisson data as a baseline.

As shown in Figure **??**, we can see that the Hurst parameters of the stable peers are quite high, as most of them are around 0.7 and some can achieve over 0.9. Comparing to the simulated Poisson data, which are generally around 0.5, it is easy to see that the arrivals of stable peers are highly dependent in 6-hour intervals and thus have clear self-similar features. On the other hand, in Figure **??**, the Hurst parameters of unstable peers are generally distributed around 0.5, which is very similar with that of the Poisson data.

In order to confirm this observation, we further check the autocorrelation coefficient of peers' arrival with different time bins. Let $\overline{\chi}$ be the complex conjugate of χ. The autocorrelation of $\underline{a}$ for a given shift τ is defined by:

$$C_{\underline{a}}(\tau) = \sum_{i=0}^{t-1} \chi(a_{i+\tau})\overline{\chi(a_i)}, 0 \leq \tau \leq t-1 \qquad (1)$$

where τ is a phase shift of the sequence $\{a_i\}$ and the indexes are computed modulo t, the period of $\underline{a}$. $\{a_i\} = \underline{a}$ refers to the input sequence (a row in global online matrix G).

Note that for self-similar processes, the autocorrelation function will decay very slowly (hyperbolically) toward zero, but may never reach zero. On the other hand, for the short-range dependent process such as Poisson or compound Poisson, the auto correlation function will decrease to zero very quickly. As shown in Figure 9, we present the autocorrelation coefficient of the stable peers, unstable peers and the simulated Poisson data.

Figure 9a shows the autocorrelation coefficient in 6-hour intervals. It is quite clear that the function of stable peers decay very slow follows a power-law-like curve. The function of unstable peers and simulated Poisson data, on the other hand, decay very fast to a close-to-zero level which shows the absence of long-range dependence. When we further increase the length of the interval to one day, one week and one month, we can see that the autocorrelation function of stable peers becomes more and more stable converging to 0.3. On the other hand, the autocorrelation function of unstable peers and the Poisson data still decrease very fast to a near-zero value. This observation further confirms that the arrival pattern of unstable peers is quite random. It does not has long-range dependence even we consider a very long time interval, e.g., one month. Therefore, the arrival of unstable peers can be better fitted by Poisson processes This is also the underlying reason why long-term relationships can hardly be built among most peers.

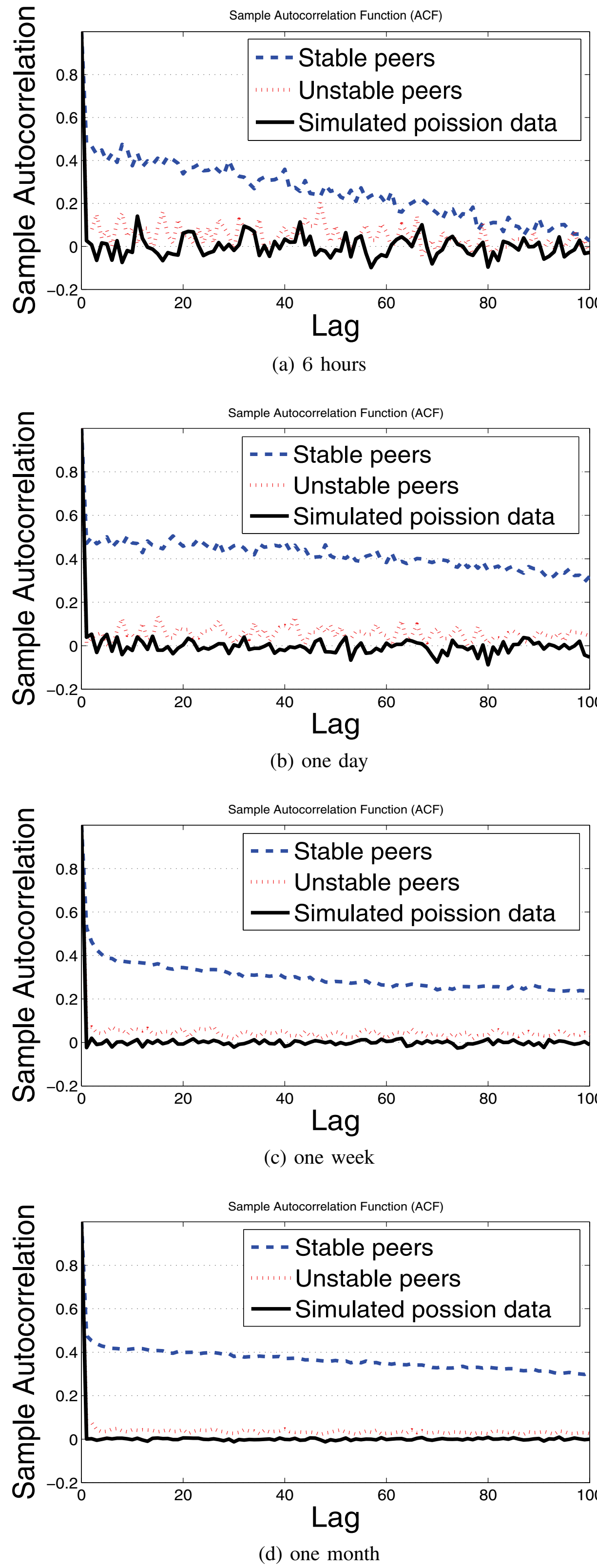
The number of stable peers is small, yet their longer lifespans and highly overlapped patterns make them very important to assist the downloading of other peers. In particular, we find that the average ratio of stable peers (per snapshot) is around 45%. How to find these stable peers is therefore an instant question. Fortunately, as a well-known long-term relationship among users, the social networks shed new light on this problem..

VI. Opportunities: Torrent Sharing On Social Networks

We will now present the measurement analysis of peers' social relationship as well as the torrent sharing on social applications.

A. BitTorrent and Social Applications

Despite its name, peer-to-peer file sharing is usually a solitary pursuit, where the peers swap bits of contents, while the peers themselves remaining anonymous to one another. Yet, more and more users as well as the BitTorrent company



(a) 6 hours

(b) one day

(c) one week

(d) one month

Fig. 9: The autocorrelation function for the data in 6 hours, one day, one week and one month.

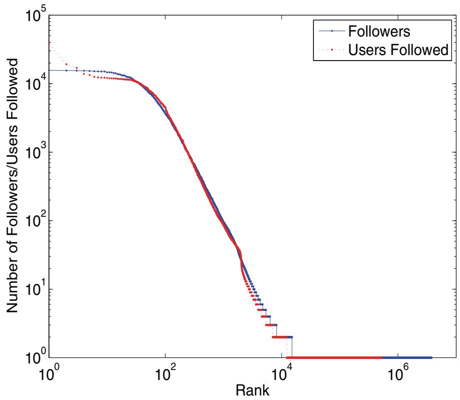


Fig. 10: Number of followers/users followed against their rank of Twitter users.

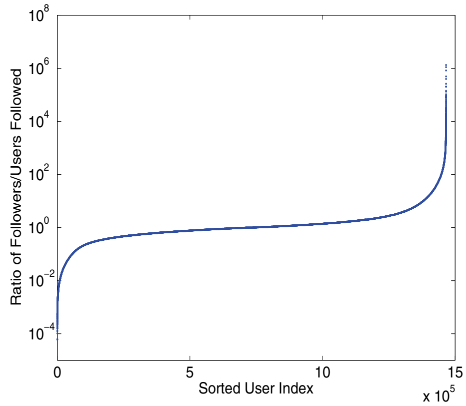


Fig. 11: Distribution of the ratio of followers and users followed.

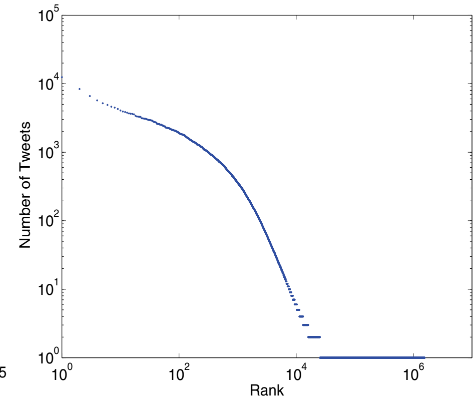


Fig. 12: Number of tweets a user post against rank.

itself is trying to enable social-network-based functions in BT. In particular, one of their designs is to make it easy for friends downloading the same swarms to talk about it. A new feature in the latest version of the uTorrent [20] client called "Torrent Tweets" allows users to talk about a given download from the application and see what everyone else is saying on Twitter[6] [21]. Of course, BitTorrent's uTorrent is just one of the many BitTorrent clients out there, and Torrent Tweets has not yet truly became a standard for BitTorrent protocols. However, these social network related functions have already start to change the way of Internet torrents sharing.

Nowadays, more and more torrents are shared on social applications such as Twitter. Similar with the increasing video sharing on Facebook [22], the torrents sharing on Twitter has been widely applied by a great number of users. In particular, we have found that more than $10,0000$ groups on Twitter site are built for torrent sharing. It is well known that Twitter emphasizes the up-to-date sharing of instant information among friends. In particular, once a user updates a message/torrent link on their space, his/her followers will be able to see it at the same time through updating notifications to their cell phones. We believed that this feature will potentially change the downloading behavior of peers and thus interesting to be investigated.

B. Peers' Long-Term Relationship in Social Communities

Recall that we have obtained $100,000$ torrents from Internet. In order to clarify the trend of torrents sharing on social applications, we further crawl the Twitter pages and check whether these torrents are also shared among Twitter communities. Our result shows that more than 2% ($2,106$ out of $100,000$) of torrents in our dataset are shared on Twitter. We call these swarms *Twitter swarms* and others *Normal swarms*[7].

We obtain the peer information in these Twitter swarms and check the number of peer encounters between our peer pair samples in Figure 14. Note that only pairs encountered at least once are considered. We can see that peers' online patterns

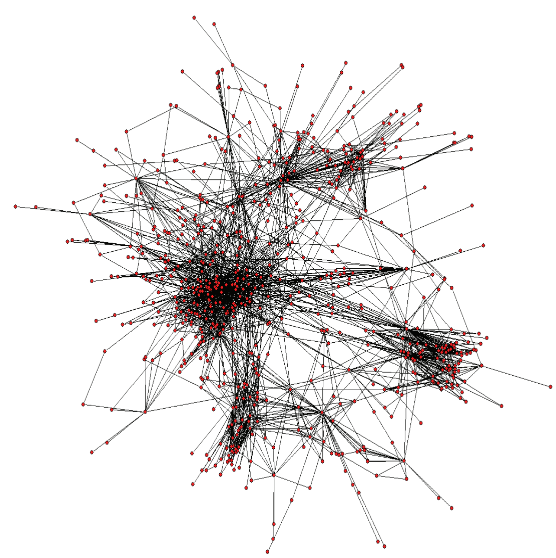


Fig. 13: An illustration of Twitter friends.

are much better overlapped in the Twitter-triggered swarms (compared to Figure 3). In particular, the ratio is increased from 5% to 35% which indicates that more peers are eligible to provide constant helps to others[8].

Due to this high encounter ratio, we further investigate the total length of overlapping time slots in Twitter swarms. As shown in Figure 15, we can see that most (around 60%) peers overlapped with others for more than 15 hours in our measurement. This is relatively a long time that could be utilized by enabling their social relationships. Furthermore, The Twitter communities consist of trusted friends, a better sharing incentive can naturally be expected. An intuitive explanation (of Figure 14 and Figure 15) is that Twitter emphasizes the up-to-date sharing of instant information among friends. Once a user updates a message/torrent on their space, his/her followers will be able to see this message at the same time (through updating notifications) and then start to download. Therefore, the peers are very likely to have similar online behaviors.

[6]It is worth noting that Twitter itself is also highly dependent on the BitTorrent to manage its thousands of data servers. In Twitter's new setup, BitTorrent-powered system has made Twitter server deployment 75 times faster than before.

[7]Note that we are not aiming to identify the Twitter users in BitTorrent swarms

[8]Note that a study from Piatek et al. [23] shows that the peers can hardly have direct data exchange with others again (be assigned as neighbors again). Our study is, however, focusing on peers' online patterns and seeking for the potential of building direct data exchange among social friends.

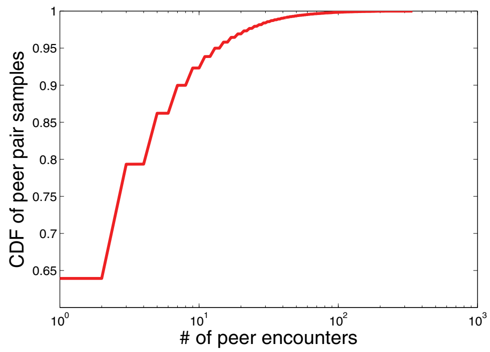


Fig. 14: # of peer encounters (in Twitter swarms).

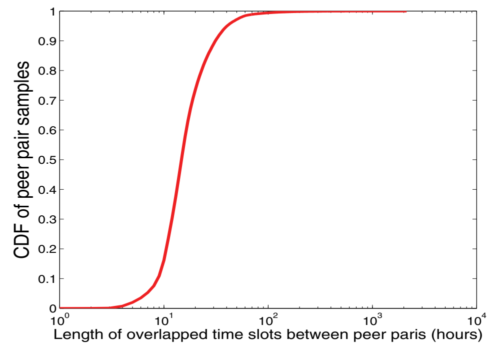


Fig. 15: Length of overlap time slots between peer pair samples (in Twitter swarms).

C. Discussions On Social Relations in Twitter

Since the torrent files are mostly included in users' tweets, the investigation of users' social relations can help us better understand the relationship between BitTorrent and social applications. In this part, we will present some of our preliminary measurement results of Twitter users. We have collected Twitter data using a customized distributed crawler, run by hundreds of nodes in PlanetLab network. The crawler sends requests using Twitter API, and obtains the information of tweets and users[9]. To get the information of a tweet, the crawler sends an HTTP request "http://twitter.com/statuses/show/id.xml", in which id is the ID of the tweet. The information of the tweet, as well as the information of the uploader, is recorded in the downloaded XML file.

In Twitter, a user can follow his/her friends or anyone he/she is interested in. We first look at the pattern of the *followers* (# of users that are following this user) and *users followed* (# of users that are followed by this user). In Figure 10, both distributions are Zipf-like after top-20. We also look at the distribution of the ratio of followers and followed, as shown in Figure 11. There are a number of users having much more followers than the number of users he/she followed, and they are probably celebrities; yet, most of the users have a comparable number of followers. An illustration of Twitter friend relationships is shown in Figure 13 including 1000 tweets with clear clusters.

We also investigate the number of tweets the users post. As shown in Figure 12, the distribution is also roughly Zipf-like, indicating there are a few very active users, and also a great number of users only posted very few tweets since joining Twitter. Since the user information is collected along with the tweet information, all the users we collected posted at least one tweet; however, we believe there are a significant number of users haven't posted anything since joining Twitter.

We can see that the Zipf-like distribution of Twitter community is quite similar with the BitTorrent community [15]. Given the growing trend of spreading torrents through social networks, We believe that significant gain can be expected through peers' long-term cooperation.

[9]The user's information is included in the corresponding tweet's information.

VII. Can Social Networks Accelerate Content Sharing?

We will now discuss the performance gain of the peer cooperation based on social relationships. A possible social network based protocol is proposed and evaluated through a trace-based Planet-lab experiment.

A. Collaboration Among BT Peers: A Simple Solution

We assume that peers' social relationships can be obtained by the trackers (either by the interaction with social applications, such as Twitter, or by our proposed *social index*[24]), and the trackers will select the majority, but not all, of the peers' social friends to build peers' neighborhoods (with maximal 8 social friends out of 10 neighbors in our design).

The standard choking algorithm in BitTorrent is designed by only changing who's choked once every 10 seconds. This is processed by unchoking the 4 peers which it has the best downloading rates from and are interested. If a leecher has completed the downloading (becames a seeder) it will use its uploading rate rather than its downloading rate to decide whom to unchoke (note that the optimistic unchoking is not discussed in here).

It is worth noting that for any leecher who wants to get data from other leechers, the key requirement is that this leecher should be unchoked by other peers. This design guarantees the instant rewards for every bit that the leechers uploaded (except for optimistic unchoking cases) which is considered robust to peers' possible selfish behaviors. However, it also hinders decent peers or peers of close relations from more efficiently cooperation; for example, the friend peers in social networks. Therefore, we make a very simple modification to leechers' choking algorithm. In particular, the leechers will use the uploading rate to choke their social friends (as a seeder in the standard BT protocol). In this design, the leechers will unchoke the 3 peers which it as the best uploading rates (in the past 10 seconds) among its social friends and 1 peer which as the best downloading rates outside the friend communities. Note that we only modified chocking protocol in this design because it is the one of the most functions that related to peer collaboration. Such a modification can help us minimize the impact to the existing BT protocols.

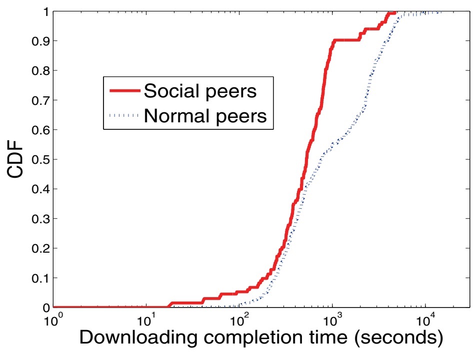


Fig. 16: Downloading completion time.

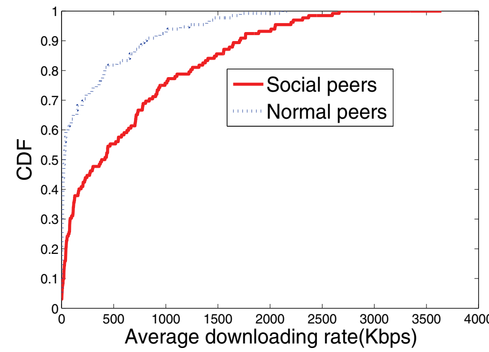


Fig. 18: Downloading rate.

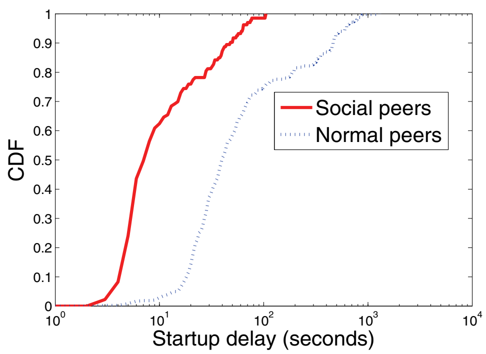


Fig. 17: Startup delay.

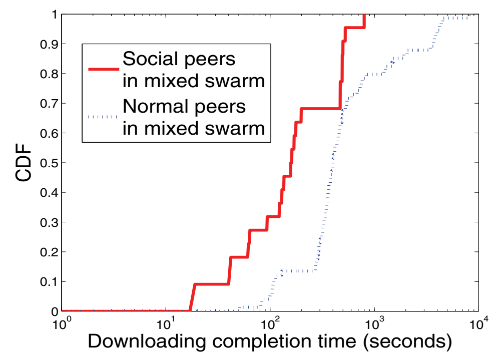


Fig. 19: Downloading completion time.

B. Performance Evaluation

In order to evaluate the benefit of social network based content delivery, we carry out Planet-lab experiments with a modified version of rTorrent client [25]. Our evaluation contains two parts: First, investigate the possible gain in an extreme case where all the peers are social friends; Second, further clarify this benefit in hybrid swarms with a small set of social friends only.

In the first experiment, we investigate the sharing efficacy in two BT swarms S_{social} and S_{normal} (both with 350 peers). S_{social} consist of social friends, and S_{normal} consist of normal peers. Therefore, standard BT protocol is applied to the clients in S_{normal} whereas our modified choking algorithm is applied to the clients in S_{social}. The content size is 900MB with piece size of 1024kB (a very popular piece size for large contents). We used a local server in our campus to run both the seeder and tracker functions, and the seeder's maximal uploading capacity is set to 10M. There are 350 peers arriving over a very short period of 2 minutes. Note that these parameters are selected based on the configuration of Planet-lab environment. Note that the peers in S_{normal} will leave the swarm as soon as they have finished the downloading. On the other hand, the social peers will continue to contribute their uploading if their friends are still downloading the content (this issue is further discussed in section 9).

Figure 16 shows the downloading completion time of swarm S_{social} and S_{normal}. It is easy to see that the proposed social network based enhancement can significantly improve peers' downloading completion time. In S_{social}, 70% peer will finish their downloading within 800 seconds whereas the maximal downloading completion time is $4,000$ seconds. On the other hand, in S_{normal}, only 40% peer can finish their downloading within $4,000$ seconds and the maximal downloading completion time reaches $20,000$ seconds. To further understand this benefit, Figure 17 shows that the startup delay (the delay of getting the first piece) is also greatly improved. In our new protocol, most peers (90%) in S_{social} will get their first piece very fast (within 1 minute). Yet only 60% peers in S_{normal} can get their first within 1 minute based on the optimistic uncorking. An intuitive explanation is that peers' average downloading rate is greatly improved due to the social relationship based enhancement. As shown in Figure 18, 30% peers in S_{social} can achieve downloading rate of 1M, while less than 10% peer can have such a high rate in S_{normal}.

The above experiment demonstrates the gain in entirely collaborative social swarms (all peers are social friends). The real world swarms, however, consist of a majority number of normal (selfish) peers. It is thus interesting to see whether the peers can still benefit in hybrid swarms with a small set of social friends only.

In this experiment, we use the trace from a real world Twitter swarm that consist of 96 peers. The content is a 300MB video file with piece length of 512kB. Based on this information, we simulate the downloading of this swarm on Planet-lab with exactly the same configuration. We modified the client of 22 social peers with our proposed uploading choking algorithm and applied standard BT protocol to other peers. In particular, the social peers will use the uploading rate based choking algorithm to communicate with their friends and use standard choking algorithm to communicate with other peers. All peers' arrivals are within a very short period of 1 minute. Normal peers will leave the swarm as soon as they have finished downloading. On the other hand, the social peers will continue to contribute their uploading if their friends are

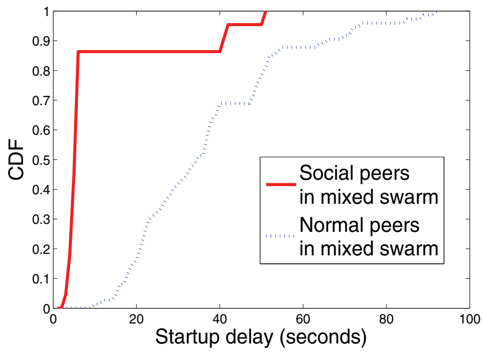


Fig. 20: Startup delay.

still downloading the content. Note that the tracker in this experiment is modified and will process the biased neighbor selection (with maximal 8 social friends out of 10 neighbors in total). The availability information of each peer is also obtained from our real world trace.

In Figure 19, we can see that the social collaboration of a small number of peers can still enable considerable benefit to peers' downloading. In particular, 90% social peers will finish their downloading within 400 seconds whereas only 60% normal peers can finish the downloading within 400 seconds. The startup delay in Figure 20 also indicates the benefit of using social networks to accelerate BitTorrent, where most social peers can get their first piece very fast within 10s.

VIII. Further Discussions

This paper takes a first step towards the challenges and potentials of finding long-term relationship among BitTorrent peers. There are still some open issues that can be further explored.

Remodeling the BitTorrent System: In this study, we have found that the arrival of most peers can be better fitted by Poisson processes while others are more likely to be self-similar. This observation can help us to clarify the dynamics in the BitTorrent system. For example, giving peers' arrival patterns, we can build better models for the downloading performance and the content availability across BT users.

Peer Cooperation in Social Communities: Due to the assumption of peers' selfish behaviors, content delivery among cooperative peers has seldom been discussed in the BitTorrent systems. Yet, giving the social relationship across the peers, it is thus interesting to see how to achieve an optimal sharing efficiency under this new context. One question is that how to optimize the content sharing in hybrid swarms with a small set of social friends only. Moreover, the seeder's location (whether the first seeder is in social networks or not) will also bring more challenges to explore such a problem.

Free Riding: Free riding is a very important issue in BitTorrent networks. Although better sharing incentives can be expected across social network friends, it is hard to say whether some social friends are also free riders. Some users may not even care about this type of free riding; yet, it would be better if this behavior can be detected in social networks.

IX. Conclusions

In this paper, we for the first time examined the challenges and potentials of long-term social relationships in P2P networks. Due to the popular torrent sharing on social applications, our analysis showed that the BitTorrent system has enough potential to apply social network based enhancements. The PlanetLab experiments further indicate that the incorporation of social relations remarkably accelerates the downloading time even in a hybrid system with a small set of socially active peers only. We believed that the torrents sharing will become more and more popular on social applications. Such a trend will bring great opportunity to improve the sharing efficiency in P2P file sharing systems.

References

[1] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, "Measurements, analysis, and modeling of BitTorrent-like systems," in *Proc. 2005 ACM/USENIX IMC*.
[2] G. Dan and N. Carlsson, "Dynamic swarm management for improved BitTorrent performance," in *Proc. 2009 USENIX IPTPS*.
[3] N. Andrade, M. Mowbray, A. Lima, G. Wagner, and M. Ripeanu, "Influences on cooperation in BitTorrent communities," in *Proc. 2005 ACM P2PECON*.
[4] P. Dhungel, D. Wu, Z. Liu, and K. Ross, "BitTorrent darknets," in *Proc. 2010 IEEE INFOCOM*.
[5] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proc. 2004 ACM SIGCOMM*.
[6] M. Piatek, T. Isdal, A. Krishnamurth, and T. Anderson, "One hop reputations for peer to peer file sharing workloads," *Proc. 2008 NSDI*.
[7] D. S. Menasche, A. A. A. Rocha, B. Li, D. Towsley, and A. Venkataramani, "Content availability and bundling in swarming systems," in *Proc. 2009 CoNext*.
[8] R. Axelrod, *The Evolution of Cooperation*. Basic Books, 1985.
[9] M. Meulpolder, L. D'Acunto, M. Capota, M. W. and J.A. Pouwelse, D. Epema, and H. Sips, "Public and private BitTorrent communities: a measurement study," in *Proc. 2010 USENIX IPTPS*.
[10] D. Choffnes, J. Duch, D. Malmgren, R. Guierm, F. Bustamante, and L. A. N. Amaral, "Strange bedfellows: community identification in BitTorrent," in *Proc. 2010 USENIX IPTPS*.
[11] D. Choffnes, J. Duch, D. Malmgren, R. Guierm, F. Bustamante, and L. A. N. Amaral, "Strange bedfellows: community identification in BitTorrent," in *Proc. 2010 USENIX IPTPS*.
[12] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in BitTorrent?" in *Proc. 2007 USENIX NSDI*.
[13] Planetlab. Available: http://www.planet-lab.org/
[14] Ctorrent. Available: http://ctorrent.sourceforge.net/
[15] J. Liu, H. Wang, and K. Xu, "Understanding peer distribution in global Internet," *IEEE Network Mag.*, 2010.
[16] J. R. Douceur, "The Sybil attack," in *Proc. 2002 International Workshop on Peer-to-Peer Systems*.
[17] B. Fan, D.-M. Chiu, and J. Liu, "BitTorrent-like file sharing protocol design," in *Proc. 2006 ICNP*.
[18] N. Liogkas, R. Nelson, E. Kohler, and L. Zhang, "Exploiting BitTorrent for fun (but not profit)," in *Proc. 2006 USENIX IPTPS*.
[19] J. Shneidman, D. Parkes, and L. Massoulie, "Faithfulness in Internet algorithms," in *Proc. 2004 PINS*.
[20] uTorrent. Available: http://www.utorrent.com/
[21] Twitter. Available: http://twitter.com/
[22] Facebook. Available: http://www.facebook.com/
[23] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson, "One hop reputations for peer to peer file sharing workloads," in *Proc. 2008 USENIX NSDI*.
[24] H. Wamg, F. Wang, and J. Liu, "On long-term social relationships in peer-to-peer systems," *Proc. 2011 IWQOS*.
[25] rTorrent. Available: http://libtorrent.rakshasa.no/

Haiyang Wang is a Ph.D. student 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.

Feng Wang received both the Bachelor's degree and Master's degree in Computer Science and Technology from Tsinghua University, Beijing, China in 2002 and 2005, respectively. He received the PhD degree in Computing Science from Simon Fraser University, Burnaby, British Columbia, Canada in 2012. He is a recipient of the Chinese Government Scholarship for Outstanding Self-financed Students Studying Abroad (2009) and IEEE ICME Quality Reviewer Award (2011). He is currently an assistant professor in the Department of Computer and Information Science at the University of Mississippi, MS, USA. His research interests include peer-to-peer networks, wireless sensor networks, cyber-physical systems, socialized content sharing and cloud computing. He is a member of IEEE. He serves as TPC member in IEEE ICME'2011, IEEE ICME'2012 and IEEE GLOBECOM'2013.

Jiangchuan Liu 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 Senior Member of 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*. He is TPC Vice Chair for Information Systems of IEEE INFOCOM'2011.

Ke Xu received the 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 network. He is a Senior Member of IEEE and a member of ACM.

Di Wu is currently an Associate Professor in the School of Computer Science, Dalian University of Technology Dalian, Liaoning Province,China, and was a visiting scholar at Simon Fraser University of Canada from 2009 to 2010. His research interests are in VANET, ADHOC, Optimization calculation theory and engineering application, etc.