Peer-to-Peer Netw. Appl. (2010) 3:237–253
DOI 10.1007/s12083-010-0076-0

Measurement and enhancement of BitTorrent-based video file swarming

Haiyang Wang · Jiangchuan Liu · Ke Xu

Received: 3 December 2008 / Accepted: 3 June 2010 / Published online: 6 July 2010

Abstract BitTorrent, the most popular P2P file sharing application over the Internet, has also been widely used for video file distribution, albeit in a download-and-play mode. In this paper, we extensively examine the characteristics of BT swarms with different contents through a large-scale measurement. Our study especially focuses on the existing video file swarms, trying to understand the potentials and challenges of providing streaming service in BitTorrent. Our results from intra-swarm, inter-swarm and AS-level measurements demonstrate that the video file swarms in the BitTorrent system are quite different from the non-video swarms. In particular, the preference bias terminates the long-term relationship between peers and raises a significant challenge to develop BT based streaming service. Fortunately, we find that most peer sets in video file swarms are good enough to support streaming service. Strong relationship between video file swarms is also observed from the inter-swarm measurement. This inter-swarm relationship can provide extra peer information for video file swarms. Therefore, the organization of existing peer information could be the key to mitigate the preference bias. To this end, a hypercomplex based virtual channel mechanism is proposed. We find that, quaternions can provide efficient and meaningful approximation in the bitfield operations. Using quaternion based three space rotation, our approach can help peers to avoid preference bias during the data delivery. This is especially helpful at the beginning of the downloading since the peer only has few local pieces. More importantly, this mechanism is transparent and friendly to the existing BT swarms with both video and non-video contents.

H. Wang · J. Liu (✉)
School of Computing Science, Simon Fraser University, British Columbia, Canada
e-mail: jcliu@cs.sfu.ca

K. Xu
Department of Computer Science and Technology, Tsinghua University, Beijing, China

Keywords BitTorrent · Peer-to-Peer · Streaming · Measurement

1 Introduction

Peer-to-Peer (P2P) networking has emerged as a successful architecture for content sharing over the Internet. BitTorrent (BT), the most popular P2P application, has attracted attention from network operators and researches for its wide deployment. Its popularity stems from the common belief that BT is efficient, robust and scalable, i.e., efficient at replicating content, especially for big objects [1]; resilient to the massive and sudden peer failures [2]; and scalable with client population increase [3].

The success of BitTorrent has greatly motivated the designed of data-driven mesh overlays for streaming applications [4]. On the other hand, BitTorrent itself has also been extensively used for video file distribution. Recent developments thus show a great interest of enhancing BT itself to provide streaming services, beyond the existing download-and-play mode [5, 6]. The current solutions however try to modify the BT implementation, in particular, the piece selection mechanism. Yet, these operations are at the core of BitTorrent, and changing them would significantly affect the efficiency of BitTorrent. Moreover, such an implementation will also be incompatible with the existing BT swarms.

In order to explore the possibility of involving streaming service in BT system, we extensively examine the BT networks with different contents through a large-scale measurement. Our measurement results suggest that there are great challenges in providing streaming services compatible with the existing BitTorrent. In particular, the intra-swarm measurement results indicate that peers always failed to take good advantage of its neighborhood information and suffered from preference bias. When the local peer has few pieces, the downloading of such a peer is always switching from one set of peers to another rapidly. On the other hand, we notice that the peer set size in most BT video file swarms are good enough for streaming service. Our inter-swarm measurements also show that, extra peer information can be recovered from the strong relationship between video file swarms. Therefore, the organization of existing peer information can be a key to avoid the preference bias, and how to make good use of the existing neighborhood information is the critical problem therein.

To meet this requirement, a hypercomplex based virtual channel mechanism is proposed. Our solution strikes to use every single peer information to optimize the downloader. On the other hand, this improvement is friendly to the existing BT protocols and will not reduce the efficiency of file sharing.

The rest of this paper is organized as follows: In Section 2, we illustrate related works. Section 3 presents our measurement results and the analysis of existing BT video file swarms. After the description of the main challenge, we proposed a hypercomplex based virtual channel mechanism to improve the data exchange efficiency of the video swarms in Section 4. The simulation and analysis of our approach are summarized in Section 5 and the paper is concluded in Section 6.

2 Background and related work

A large number of applications have emerged in recent years for P2P video services, among which CoolStreaming and PPlive are two representatives [4]. Both of them have shown their great success in theory and practice. It worth noting that these applications are based on the *data-driven* approach [7]. Such a approach is illuminated by the BitTorrent-like protocol with a real-time scheduling algorithm. However, these applications are exclusively dedicated to video demands; Other contents cannot be shared through the system. Introducing a streaming functionality to BT systems has the potential to address this problem. Vlavianos et al. [5] proposed a novel BT streaming protocol based on the modification of piece selection mechanism. Yet, this modification will reduce the efficiency of existing file delivery because some peers are no longer perform the rarest first algorithm. Moreover, real-world measurements are also required to examine the capability of existing video file swarms in the BitTorrent system.

As a matter of fact, there are many measurement studies about BitTorrent and other similar P2P applications. In the case of intra-swarm measurement, swarm diameter and clustering of BT swarms are well studied by many researchers, such as [1, 8], from both theoretical and experimental approaches. C. Dale et al. [9] find that the intra-swarm topology of BitTorrent is more likely to be a random graph rather than a clustering small world. On the other hand, since the topology of BT swarms is related with the deployment and the implementation of trackers, some studies are based on the analysis of tracker's log files. These works are relatively long-term measurements and mainly focused on the evolution of BT swarms. They demonstrate that the structure of BT swarms are highly dynamic, the swarm evolution may depending on many aspects: peer arrival pattern and average connection time [10], peers downloading speed [11], choke algorithm and the content pieces selection algorithm [10, 12].

The study of inter-swarm characteristics are relatively few. One study in [13] presents an analysis of content that peers offer to others in eDonkey 2000. The correlation between files is evaluated, as a result, at least 30% of the eDonky files have a correlation larger than 60% to some other files. Another paper from L. Guo et al. [14] find an interesting peer migration behavior in BT system and define it as the inter-torrent relationship. The authors investigate the existence and degree distribution of BT's inter-torrent relationship and build a probability model for this multi-torrent environment. Moreover, a multi-torrent collaboration system is proposed to improve the service availability. In our study, we examine this property directly by active measurement. The results are exclusively focused on the video file swarms and some new characteristics are presented.

The AS-level measurements of BT swarms are even fewer. S. Sen et al. [15] analyzed P2P traffic(FastTrack, Gnutella, and Direct-Connect) by measuring flow-level information collected at multiple border routers across a large ISP network. Some AS level properties such as the density of an AS and their distributions are discussed. However, our measurements are focused on the BT networks and aim to explore the potential requirements and the possible improvements of such a system. Some unique properties such as AS-clustering and user popularity are discussed in our study.

3 Measurement and analysis of existing video file swarms in the BitTorrent system

The objective of this section is to investigate the advantages and potential problems to apply streaming service in the BT networks. Although BitTorrent is designed mainly for file sharing, a large amount of video contents are already existed in such a system. Therefore, the BT peers that sharing the video contents are the potential users for the ideal BT streaming system. The study of these users(swarms) can provide useful information for the design of the BT based streaming service. In this section, we for the first time analysis the properties of BT swarms with different contents exclusively, and our measurement results are mainly focused on the video file swarms.

3.1 Terminology

The terminology used in the BitTorrent community is not standardized. Therefore, we defined the terms used in this paper for the sake of clarity.

Metainfo file The metainfo file (torrent file) contains all the information used to download the contents. In particular, it includes the number of pieces, SHA-1 hashes and the tracker information.

Swarm The BitTorrent swarm (torrent) is the set of peers cooperating to download the same content using the BitTorrent protocol. In this paper, the BT swarms are classified according to the type of the sharing contents (video file and non-video swarms).

Peer set The peer set is the set of neighbors for a given BT peer. The data exchange in BitTorrent system can only happens within one peer set.

Tracker The tracker is the only centralized component in the BitTorrent system. It keeps the peer information for the swarms that currently participating in the download.

Intra-swarm characteristic The intra-swarm characteristic is the statistics of the BitTorrent system within the BT swarms. In particular, the swarm size(peer population), peer set size, content size and piece size are all belong to the intra-swarm characteristics.

Inter-swarm characteristic The inter-swarm characteristic is the statistics of the BitTorrent system between different BT swarms. In particular, the peer migration between different BT swarms is belong to the inter-swarm characteristics.

3.2 Methodology and data sets

In our study, we considered more than 150,000 metainfo files, found mainly through www.btmon.com. We developed a script, which can automatically detect the 'href' field in a given HTML file and download files end with '.torrent'. Since the video and non-video metainfo files are collected separately, their uploading time are not within the same period. In what follows, we refer to the following sets.

SET1 (Newer-video set): Set of 30415 video metainfo files advertised by www.btmon.com from Dec 25 2007 to Aug 12 2008.
SET2 (Older-video set): Set of 91515 video metainfo files advertised by www.btmon.com from May 10 2006 to Dec 24 2007.
SET3 (Non-video set): Set of 36252 Non-video metainfo files advertised by www.btmon.com from Feb 12 2007 to Aug 12 2008.

Within all data sets, there are 934 bad metainfo files, 2,804 unavailable swarms due to the tracker failure and 15,040 swarms have less than two peers. None of these abnormal swarms is included in our study.

In order to get the global view of BT swarms, we carry out an Internet based measurement using the PlanetLab [16] nodes. We run a modified version of CTorrent [17] (CTorrent is a very typical BitTorrent client in FreeBSD) on more than 200 PlanetLab nodes. This program was modified to log mainly the peer level information such as IP addresses and etc. Since the contents of many Internet swarms may involve copyright problems, no content will be downloaded in our measurement. Moreover, a preprocess is applied to filter the peer information of probing nodes in the raw data.

On the other hand, most of our data sets are obtained from a single web site. In order to avoid data set related bias, it is important to confirm the representativeness of our data set before the measurement. Figure 1 shows the IPv4 prefixes that we probed from a subset of SET1, where all IP addresses are coded for privacy issues. In this figure, the diagonal refers to the universal set of all IPv4 prefixes and the dotted areas in the line denote the missing prefixes in our data sets. We observe that most dotted areas are due to the preserved IPv4 prefixes such as 192.168.0.0/16 and 169.254.0.0/16. This prefix

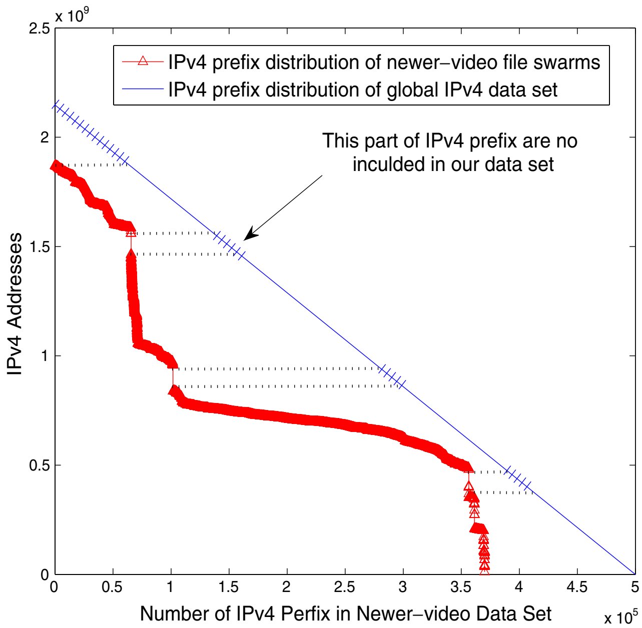

Fig. 1 IP prefix distribution of SET1

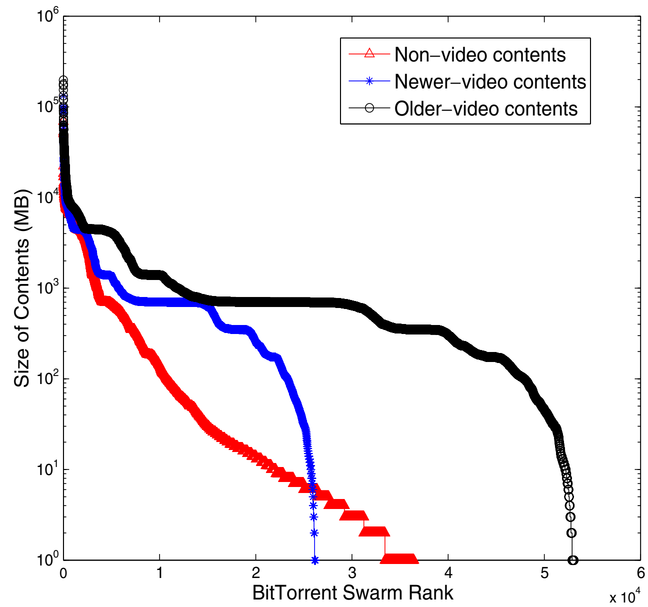

Fig. 2 Content size of BitTorrent swarms (sort in descending order)

distribution shows that, the user distribution of our data sets is quite universal, the users in nearly 80% of the IPv4 prefixes are discussed in our study.

3.3 Intra-swarm characteristics

In this part, we present the measurements and investigations of BitTorrent's intra-swarm characteristics. We especially focused on video contents and discussed the data exchange efficiency of BT video file swarms based on the measurement results.

Content size is an important characteristic in all P2P systems. Figures 2 and 3 show the distribution of content size among different data sets. We first observe that the contents shared by BT video file swarms are mostly very large. In the video file swarms, the mean object size is approximately 1,000 MB and 90% of video contents are larger than 100 MB. Moreover, there are 5% of the video contents with the size larger than 10 GB and the maximum video size reaches nearly 20 GB. On the other hand, the size of non-video swarms is relatively small, with only 30% of the non-video contents larger than 100 MB. The statistical characteristics are presented in Table 1. Compared to other applications, the content size of BT swarms is the largest.

L. Cherkasova et al. [18] also aimed to establish a set of properties in content delivery video networks (CDN). The measurement of enterprise media servers shows that 79% of files are belonged to medium/short video group with file duration shorter than 40 min (with the common assumption of frame rate and resolution, the file size will be less than 70 MB).

All the evidence show that BitTorrent video file swarms have the largest content size within most Internet applications. Moreover, among all existing techniques, CDN video sharing system is the only one that can provide streaming service with similar content size. Yet, there is no high speed centralized servers in BitTorrent, the service capability of the BT streaming

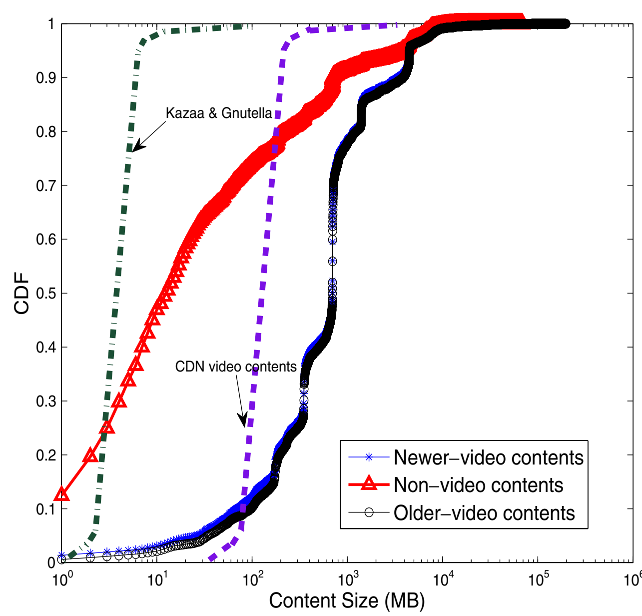

Fig. 3 Cumulative distributions (CDFs) of content size

Table 1 Facts of views, content size

(MB)	Min	Max	Mean	Median	Std
Non-video	0	68,996	471.3	12	1,562.2
Newer-video	0	132,629	1,188.6	696	2,974.0
Older-video	0	196,780	1,270.9	697	3,389.3

system cannot be guaranteed. The effect of big content size in BT streaming system will be discussed later in this section.

Although the BT video file swarms have very large contents, some of these contents consist of a set of files rather than a big single file. We plot the cumulative distribution of file numbers in Fig. 4. It shows that, multi-file content is very popular in all types of BT swarms. Moreover, the file number of video content is larger than that of non-video contents. This is intuitive because most non-video contents are archived into compressed files before the transition. Compare this result with the content size distribution in Fig. 2, we find that the average file size in BT video file swarms reaming the largest with the average size around 90 MB.

In order to explore the different video formats in the BitTorrent system. The popularity of different video contents in SET1 is presented in Fig. 5. We learn that The AVI video is the dominant type in BT video file swarms, whereas other video types are all relatively few. It is also worth noting that more than 95% huge video contents (with the size larger than 2.5 GB) are archived files. However, these files are relatively few in

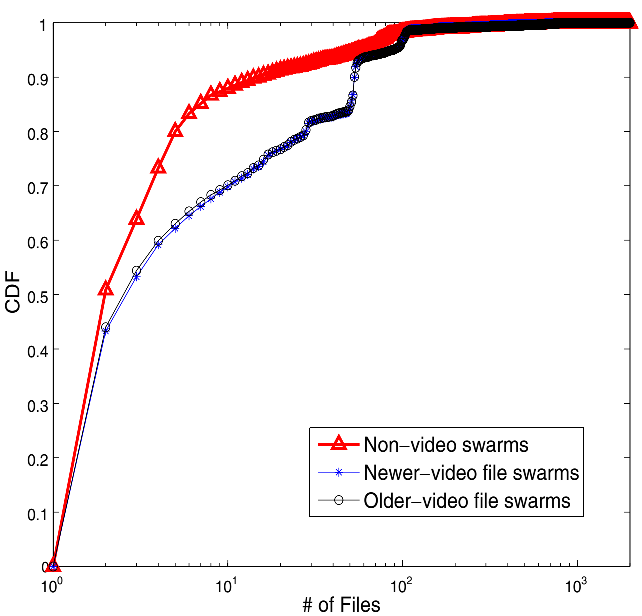

Fig. 4 Number of files in BitTorrent swarms (CDFs)

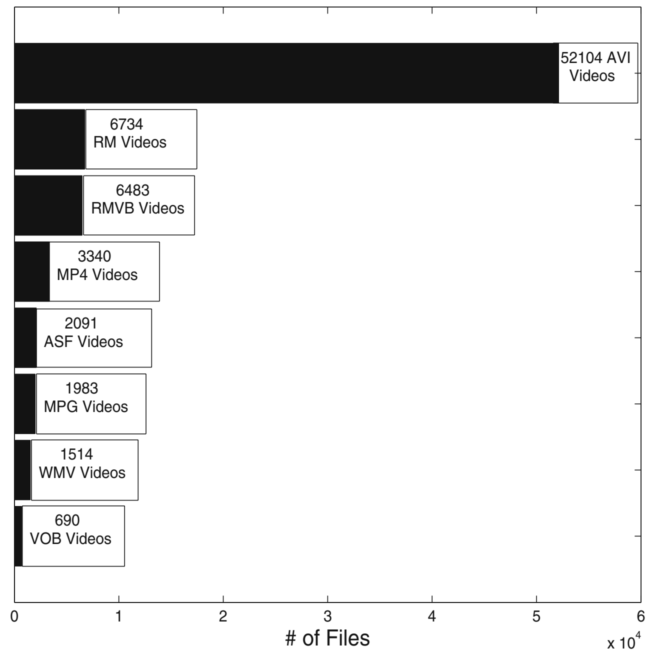

Fig. 5 Popularity of different video formats

our data sets and the further investigations are beyond the scope of this article.

The piece length is a critical parameter in BT networks. P. Marciniak et al. [19] show that for small-sized BT contents, small pieces(around 16 KB) enable shorter download times. However, in order to distribute larger contents in a shorter time, the optimal piece size will be highly depending on the content size. Figure 6 shows our observation of BitTorrent's piece size. Two types of video file swarms have relatively larger piece length with the mean value around 850 KB, whereas the piece size of most non-video swarms are less than 330 KB. The Top-3 popular piece sizes in video file swarms are 1,024, 512 and 256 KB which are proven to have small overheads to distribute large contents. However, in streaming systems such as PPlive, the overhead rate is even smaller and usually around 6.2% [20]. In PPlive, Chunk is defined and used for the purpose of advertising to neighbors what parts of movie a peer holds in PPlive and its size is fixed as 2048 KB.

Figure 7 shows the distribution of BT swarm size. This distribution is relevant to the popularity of different BT contents. We learn that although the video file swarms are mostly larger than non-video swarms, the popularity of both follows the Zipf-like distribution. However, This slop is different with the measurement result of PPlive [21] which shows the peer number of top 100 channels have a linear relationship with the channel rank in log-log scale. In particular, as shown in Fig. 7 with the dotted line, there are very few channels

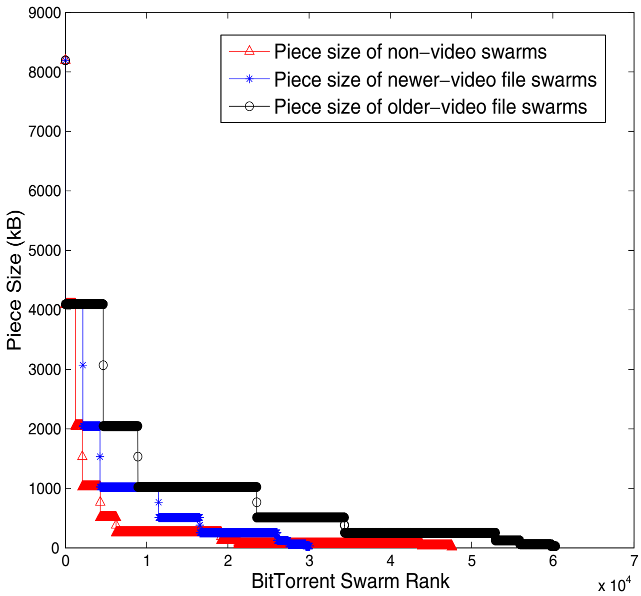

Fig. 6 Piece length of BitTorrent swarms

Table 2 Facts of views, swarm size

Number	Min	Max	Mean	Median	Std
Non-video	0	10,919	35.6	4	355.4
Newer-video	2	9,123	70.7	9	384.6
Older-video	0	12,795	82.2	8	616.8

in the PPlive with less than 100 peers. On the other hand, the size of more than 90% BT video file swarms are smaller than 100. The statistical characteristics of swarm size are shown in Table 2.

Figure 8 shows the peer set size and its ratio to the total swarm size. These Figures indicate that the size of most BT swarms are relatively small and around 20 peers. Moreover, around 90% peers in BT video file swarm have the peer set size larger than 50% of the swarm size. By Comparing these results with PPlive [22] (the average degree of PPlive network is around 30 even in a very popular channel), we learn that the number of neighbor peers is quite similar. Therefore, assume that the peers' upload capacity follows the same distribution, the BT streaming system could achieve a similar performance with that of PPlive.

However, in the standard BitTorrent protocol, peer will only upload to a set of unchoked peers(one peer needs at least one piece from another peer). D. Qiu et al. in [23] used this concept to analysis the efficiency of BT file distribution and defined the efficiency for BitTorrent-like fragment exchange scenario. Yet, in the standard streaming systems, more neighbors are required at the same time to achieve a constant downloading rate. Therefore, we analysis the data exchange efficiency of the ideal BT streaming system as the probability that a given number of peers are all interested in one downloader:

Assuming that each peer chooses pieces randomly from the entire content and the pieces each has are independently chosen. M is the total content size and m is the piece size. k is the minimal number of neighbors required for streaming service and n_i is the number of

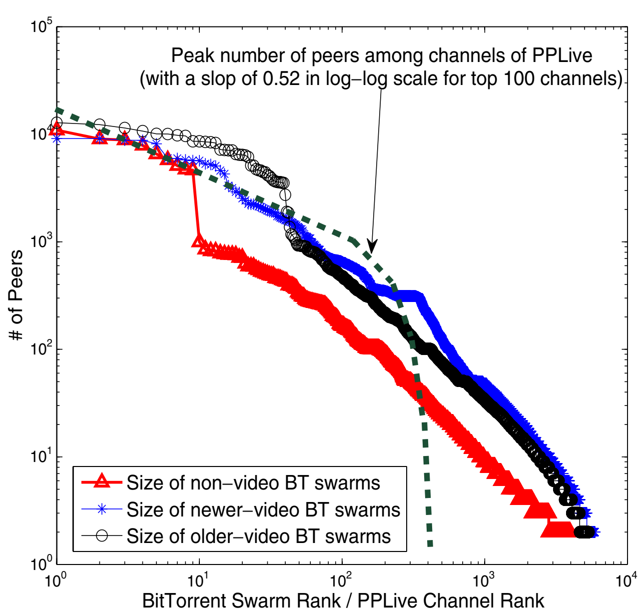

Fig. 7 BitTorrent swarms size

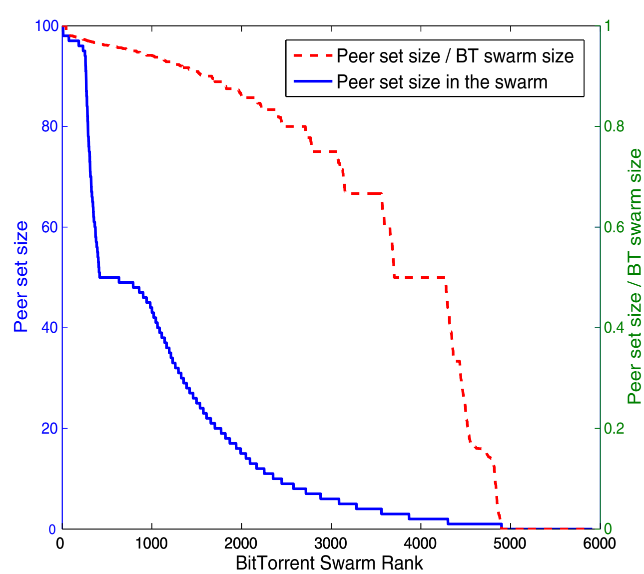

Fig. 8 Peer set size of BT video file swarm

piece that peer i has. The probability that there are k peers interested in peer i is:

$$
\begin{aligned}
&\mathbb{P}\{\mathit{There\ are\ k\ peers\ interested\ in\ peer\ i}\}\\
&= \mathbb{P}\{\mathit{Peer\ j\ is\ interested\ in\ i}\}^k\\
&= \{1 - \mathbb{P}\{\mathit{Peer\ j\ need\ no\ piece\ from\ i}\}\}^k\\
&= \left\{1 - \sum_{n_j=1}^{M/m-1} \sum_{n_i=0}^{n_j} \frac{1}{(M/m)^2} \mathbb{P}\left\{\begin{matrix} \mathit{j\ need\ no\ piece\ from\ i} \\ \mathit{given\ n_i, n_j} \end{matrix}\right\}\right\}^k\\
&= \left\{1 - \sum_{n_j=1}^{M/m} \sum_{n_i=0}^{n_j} \frac{1}{(M/m)^2} \left[\binom{\frac{M}{m} - n_i}{n_j - n_i} \Big/ \binom{\frac{M}{m}}{n_j - n_j}\right]\right\}^k\\
&= \left\{1 - \sum_{n_i=0}^{M/m} \frac{1}{M/m} \left(\frac{\frac{M}{m} - n_i}{\frac{M}{m}(n_i + 1)}\right)\right\}^k \qquad (1)
\end{aligned}
$$

We draw the efficiency of this BT video file swarm in Fig. 9 with varying k (according to our measurement results, in a general BT video file swarm with content size $M = 1$ GB and piece size $m = 512$ KB). This figure indicates that the probability is quite low if a downloader wants to obtain a constant number of *available peers* (these peers are all interested in the downloader). Figure 10 further discusses the reaction of efficiency with different content sizes. It shows that large contents will reduce the efficiency of the swarm, especially when peers have fewer pieces. Moreover, the efficiency in Figs. 9 and 10 only show the probability that all available peers will serve the downloader with a maximal duration of $t \geqslant m/x'$ (assume the source rate of the fastest downloader in the peer set is x'.). There is no guarantee that all these peers will keep unchoking the downloader in the future. In order to validate this analysis, we choose a typical swarm in SET1 with the size equal to 137 and content size of 1.3 GB. We connect to this swarm using a server located in our campus. The BT client in this server is modified to log both the number of available peers and source rate in each time slot. The result is shown in Fig. 11, where the total

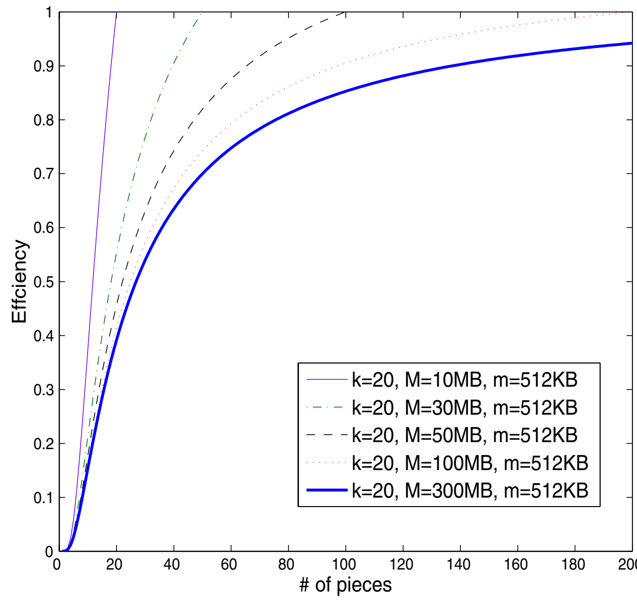

Fig. 10 Efficiency of video swarms with different content size

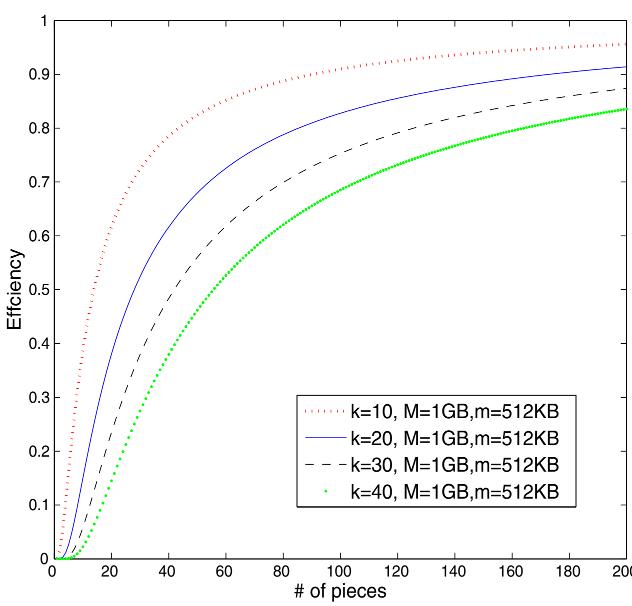

Fig. 9 Efficiency of video swarms with different requirements

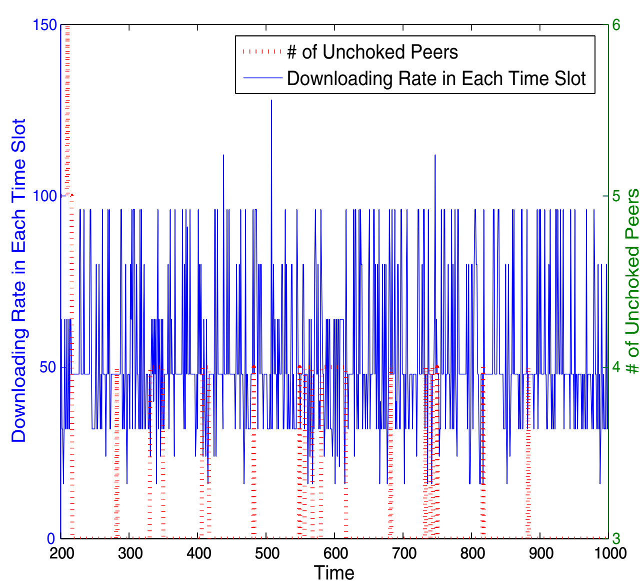

Fig. 11 Switching of available peers in BT video file swarm

swarm size is unchanged between time 200 and 1,000. This figure indicates that although there is no peer failure and departure, the number of available peers for the downloader is keep switching between 3 and 4 for a long time. According to the log file, peer set size of the downloader is 37. However, most neighbors are keep choking the downloader and no peer is serving the downloader continually for more than 11 min. Authors in [14] also investigate the source rate variation of a typical swarm, where the downloading speed at different time is highly diverse. In general, we learn that most BT clients know the distribution of the contents and the requirements of other peers. However, they just fail to take good advantage of these useful peer information and suffered from preference bias.

3.4 Inter-swarm characteristics

The intra-swarm level measurement shows that, BT video file swarms are different with non-video swarms and existing streaming systems. The unique characteristics such as large content, small peer set and the lack of long-term relationship rise a big challenge to adopt streaming service on such a system. However, there are millions of BT swarms on the Internet, the inter-swarm relationships, such as peer migration, may provide potential improvements for the ideal BT streaming service.

Using video contents as an example, the inter-swarm topology in BT networks are mainly due to two reasons: First, old peers that had downloaded a content from a BT video file swarm may come back for other video contents. Second, some peers may also download more than one video files simultaneously. Therefore, we can model the relationship among different torrents in the P2P system as an undirected graph. Each node in the graph represents that some peers in swarm i have downloaded the file from swarm j, even though they are not in swarm j currently. This potential relationship can be used to optimize the availability and the peer set size [14]. In order to investigate the inter-swarm relationship between BT video file swarms, we carry out another Internet based experiment. We randomly choose 8,893 meatinfo files in SET1 and again using more than 200 PlanetLab nodes to do the probing. The client programs are modified to probing the IP list of each video file swarm within 120 s. Using this method, all video file swarms are probed more than 200 times and the information of most peers can be obtained.

According to pervious intra-swarm measurement results, the size of typical BT video file swarms are less than 100. Therefore, we analysis 644 video file swarms with average size of 83.5. We present the degree distribution of the graph in Fig. 12. In this figure, each point in the x-axis denotes a BT video file swarm (a metainfo file), sorted in an ascending order of swarm size. Compare to the measurement result of [14], the most interesting difference is that, the degree of BT video file swarms are remarkably higher. In our measurement, most video file swarms are sharing more than 15 peers with another. Since only the small swarms are examined in our experiment, this relationship is quite strong and could be even stronger between large video file swarms. We can also find that degree of these small video file swarms to all 8,893 swarms are related with its swarm size. Peers in larger video file swarms are more likely to migrate to other swarms. The Fig. 13 can also confirm this observation. In this figure, the connectivity matrix between these 644 swarms are presented and swarms are sorted in an ascending order. Each point in the matrix indicates that at least five peers are shared between these two swarms.

3.5 AS-level characteristics

Beside the inter-swarm relationship, it is also important to explore the AS level characteristic of BT video file networks. First of all, it is relevant to the user requirements in different ISPs; On the other hand, AS-level information may also provide potential improvements to the BT system. For example: Assuming the availability of BT networks can be guaranteed (there is at least one seeder in each swarm), It is well-known that if the BT trackers can assign peer set according to their locations instead of random, the downloading rate and RTT can both be optimized.

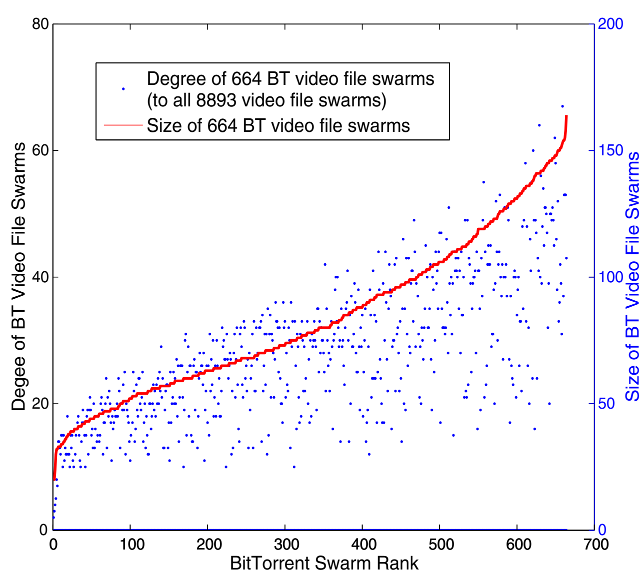

Fig. 12 Degree distribution of BT video file swarms

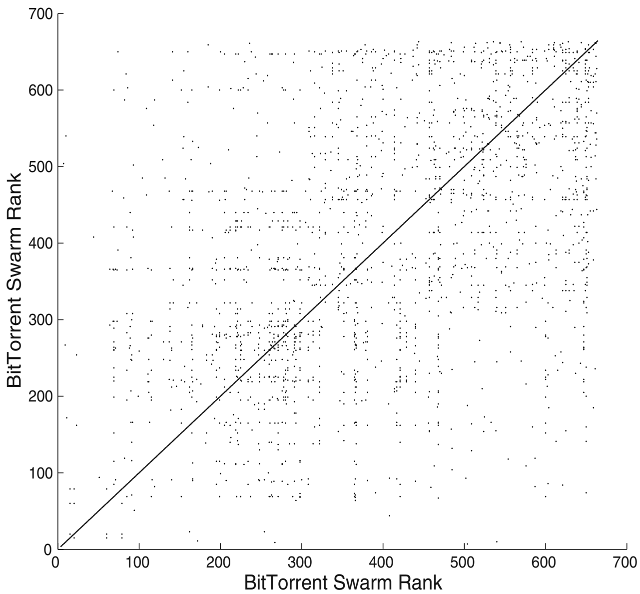

Fig. 13 Connectivity matrix of 644 video file swarms

We again use the 8,893 BT video file swarms in SET1, and collect the AS information of every peer in each swarm. This probing is based on the 'whois' command on Linux system and most replies are from 'whois.cymru.com'. From Fig. 14, the AS popularity of video BT peers fits the exponential distribution, that is, among all 2,405 ASes in our measurement, most of them have less than 10,000 video BT peers. According to our measurement results, we also present the Top-10 ISPs/ASes with most video BT peers in Table 3. This result can be regard as the potential requirements of BT streaming service in these popular ASes.

Table 3 Top 10 ISPs (BT video user)

AS #	Peers	AS Name—internet service provider
3,352	165,469	TELEFONICA-DATA-ESPANA(TDE)
3,662	129,047	DNEO-OSP7-COMCAST CABLE
6,461	127,297	MFNX MFN-METROMEIDA FIBER
2,119	113,597	TELENOR-NEXTEL T.NET
19,262	101,390	VZGNI-TRANSIT-Verizon ISP
3,301	97,658	TELIANET-SWEDEN TELIANET
3,462	96,564	HINET-DATA CBG
4,134	87,392	CHINANET-BACKBONE
6,327	86,964	SHAW-SHAW COMMUNICATION
174	74,453	COGENT COGENT/PSI

Assume that the inter-AS RTT is relatively larger than that of intra-AS RTT. In a given swarm, if most peers are belonged to the same AS, the average delay between all user pairs will be smaller. We use ω/τ to characterize the AS-clustering property of BT video file swarms, where ω is the size of the largest AS-cluster and τ is the size of the video file swarm. In Fig. 15, each point in the x-axis denotes a BT video file swarm sorted in an ascending order of swarm size. We observe that this ratio is relevant to the swarm size. Bigger AS-clusters are more likely to exist in larger swarms. Moreover, the minimal ratio is also increasing in a small slope. This figure indicates that AS-clusters does exist in BT video file swarms, when the swarm size is very large, this ratio can even reach 47% in some cases. In order to quantify this characteristic, we further examine the size of the TOP-4 AS-clusters in the BT video file swarms in Fig. 16. We find that although large

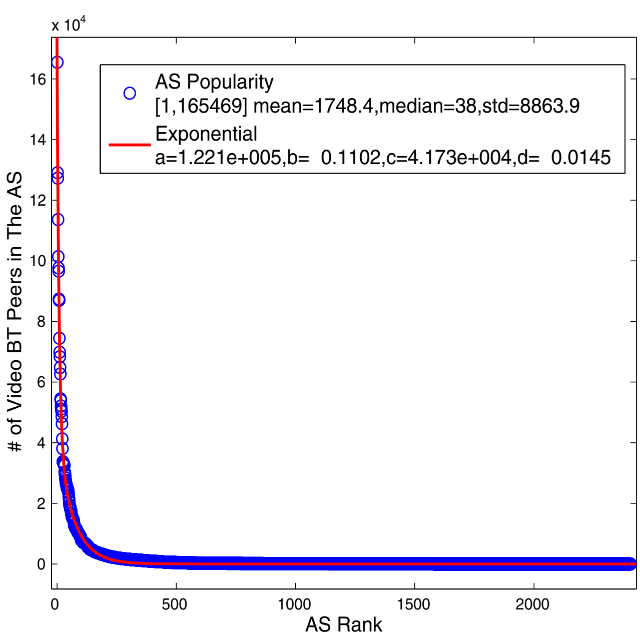

Fig. 14 AS popularity of existing BT video file swarms

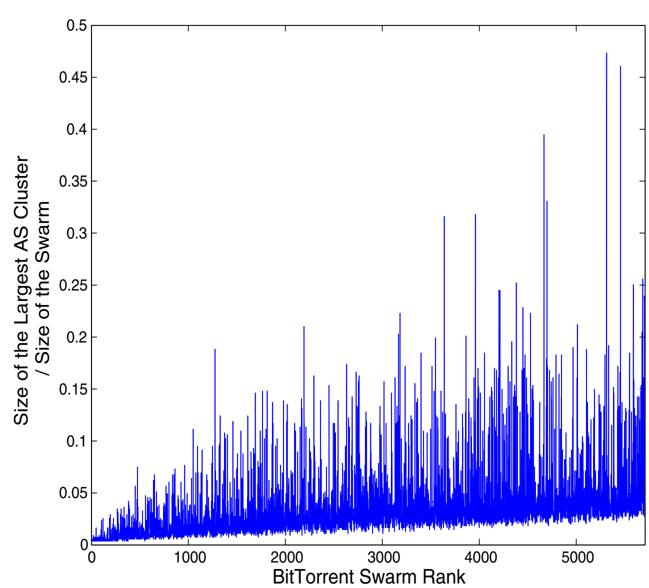

Fig. 15 The ratio of AS-cluster to the swarm size

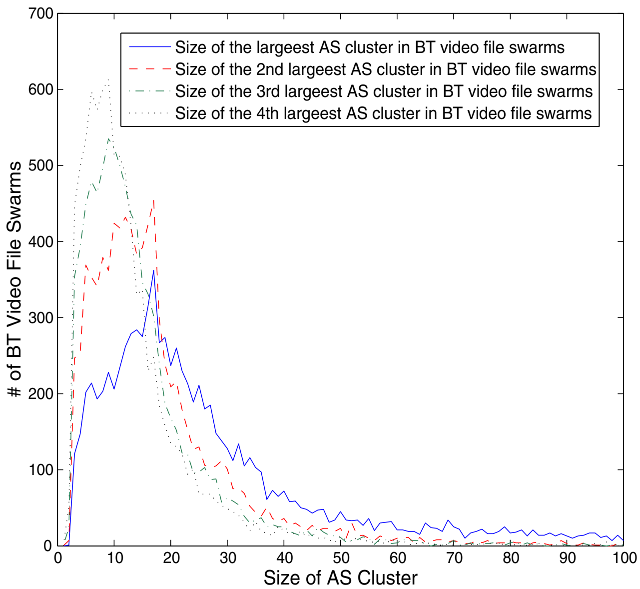

Fig. 16 Size of Top-3 AS-clusters in video file swarms

AS-clusters are existing in BT video file swarms, most AS-clusters are remaining small and around ten to 20 peers. This result shows that, large AS-cluster is just a special case in BT video file swarms; Peers in most swarms are more likely to be uniformly distributed in all ASes. In this case, possible improvement from the AS-cluster is limited for the BT streaming service.

Till now, we have discussed the characteristics of existing BT video file swarms in three different levels: The intra-swam measurements raise a big challenge to implement a BT based streaming system. Huge content, limited swarm and the lack of long-term relationship is obvious not suitable for streaming applications. Therefore, the existing BitTorrent framework may not be our best choice to build a well-performed streaming system.

On the other hand, the intra-swarm measurement results indicate that peers always failed to take good advantage of all its neighborhood information and suffered from preference bias. The inter-swarm measurements also show that extra peer information can be recovered from the strong relationship between video file swarms. Therefore, a suitable peer organization method has the potential to mitigate the preference bias in the idea BT based streaming system.

4 Quaternion representation of the bitfield and a possible extension

In the previous section, we have learnt that the preference bias is the main challenge to built a BT based streaming system. In particular, the dynamics of peers' preference terminates the long-term relationship and peers strike to meet such a constraint during the downloading. The main reason is that the peer selection algorithm of BitTorrent is not aimed to build stable connections. During the peer selection, the relationship between two BT peers is defined as either 'interested' or 'not interested'; Other details in the bitfield, such as the number of interested pieces, are largely ignored. We believe that, if we select neighbor peers carefully according to their bitfields relationship, the preference bias problem in the BT based streaming system could be mitigated or minimized to an acceptable level. Therefore, the organization and comparison between different bitfields is a critical problem therein.

In order to address such a problem, we proposed the quaternion based representation of the bitfield in this section. A possible extension is also discussed to improve the data exchange efficiency and to mitigate the preference bias at the beginning of the downloading.

4.1 Discussion of different bitfield representations

In this part, we will discuss why different representation methods will affect the bitfield operations in BitTorrent system.

According to the standard BitTorrent protocol, the peer selection is based on the comparison of bitfields and the uploading capacity. In general, this module is used to decide which neighbors can server the local peer better than that of others.

In the dedicated streaming systems, peer selection is mainly based on the classification of peers. In particular, peers in these systems can be well organized according to their different playback positions; Most of the time, a peer's downloading targets are those whose playback positions are ahead and closer to the peer's playback position. However, in the ideal BT streaming system, it is hard to infer the 'distance' between two bitfields because most peers are file sharing users and working in a download-and-play mode.

In order to solve such a problem, one possible solution is to count the total amount of the interested pieces between peers. However, in this case, the piece position and the distribution are ignored. On the other hand, we can also compare the bitfields bit-by-bit and record the position of interested pieces. Yet, there are generally thousands of pieces in a single BT content; these discrete piece positions are also too abstract to infer the 'distance' between two bitfields. Therefore, a coherent bitfield representation is required to describe different metrics together in one structure. To this end, inspired by the color representation and comparison in computer vision studies, we are trying to fit the bitfield into a higher dimensional coordinate system and treat all parts of bitfield as a whole unit.

Using 2D-Space as an example. Figure 17a and b show an example if we simply put a bitfield (length equal to 12 bits) in the 2D coordinate system. In these figures, each point in the x-axis denotes the number of pieces that the peer has in the upper 6 bits; each point in the y-axis denotes the number of pieces that the peer has in the lower 6 bits. Therefore, the distance between different bitfields could be approximated by the Euclidean distance between different nodes (Fig. 17a). Moreover, the peers can further be classified by the position of their bitfield in the 2D-coordinate system (Fig. 17b).

Note that the above example is just an analogy that demonstrates the possible alternative of bitfield representation. Our solution, on the other hand, are beyond the simple using of the node distance. In particular, we proposed a 3D representation of bitfield and proved that the 3D rotation of quaternion can help us to quantify the distance between bitfields and find the suitable neighbors.

On the other hand, matrix is also an option for the bitfield representation. However, quaternion operations are more efficient than that of matrix and more suitable for real-time decisions. This is intuitive because quaternion is more compact than that of matrix representation [24]. It also worth noting that, a higher dimensional representation can of course provides better approximation to the piece distribution; However, the operation overhead will also be increased. The tradeoff between the dimension and the overhead is another critical problem yet beyond the scope of this article.

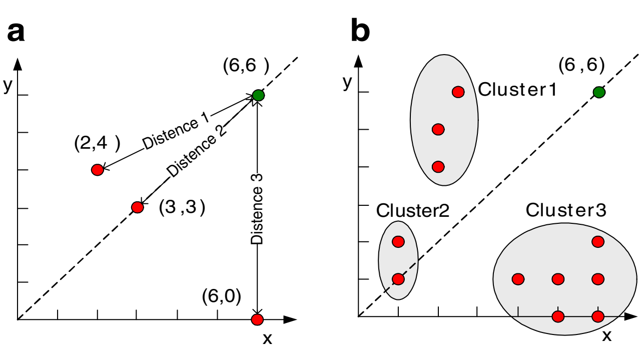

Fig. 17 Bitfield in 2D-coordinate

4.2 Quaternion based bitfield organization

The quaternions were discovered by Hamilton in Dublin in 1843 [24] and were extensively studied in the Math field.

1) *Definitions* A quaternion may be represented in hypercomplex form as

$$q = a + bi + cj + dk \tag{2}$$

where a, b, c and d are real and i, j and k are operators obeying the following multiplication rules

$$i^2 = j^2 = k^2 = ijk = -1$$
$$ij = k,\ \ jk = i,\ \ kj = j$$
$$ji = -k,\ \ kj = -i,\ \ ik = -j$$

For these rules it is clear that multiplications is not commutative. This is the only significant way in which the quaternions depart from the usual rules of algebra.
A quaternion has a real part, (a in Eq. 2) and an imaginary part. The latter has three components and is thus a vector quantity, often denoted by $V(q) = bi + cj + dk$. For this reason, the real part is sometimes referred to as the *scalar* part of the quaternion $S(q)$ and the whole quaternion may be represented by the sum of its scalar and vector part as $q = S(q) + V(q)$. A quaternion with a zero real or scalar part is called a *pure quaternion*
The modulus and conjugate of a quaternion follow the definitions for complex numbers

$$|q| = \sqrt{a^2 + b^2 + c^2 + d^2} \tag{3}$$

$$\bar{q} = a - bi - cj - dk = S(q) - V(q) \tag{4}$$

A quaternion with unit modulus is called a *unit quaternion*. The inverse of a quaternion $q^{-1}q = qq^{-1} = 1$ is given by

$$q^{-1} = \frac{\bar{q}}{|q|^2} \tag{5}$$

2) *Polar form* Euler's formula for the complex exponential gengralises to hypercomplex form

$$e^{\mu\phi} = cos\phi + \mu sin\phi \tag{6}$$

where μ is a unit pure quaternion. Any quaternion can be represented in polar form as

$$q = |q|e^{\mu\phi} \tag{7}$$

where, μ and ϕ are referred to as the *eigenaxis* and *eigenangel* of the quaternion, respectively. μ identifies the direction in three-space of the vector part and may be regarded as a true generalization of the complex operator i, since $\mu^2 = -1$. ϕ is analogous to the argument of a complex number, but is unique only in the range $[0, \pi]$ because a value greater than π can be reduced to this range by negating or reversing the eigenaxis. We can visualize the eigenaxis as the imaginary axis of an Argand diagram [25], the real axis of which is aligned with the scalar axis of the quaternion four-space. the eigenaxis is perpendicular to the real axis, but need not be aligned with any of the three imaginary axes.

3) *Bitfield in three-space and rotations* In this part, we will discuss the quaternion representation and rotations of the bitfield. It is well known that the bitfield of every BT leecher is highly dynamic; It changes from time to time until the leecher has a full copy of the content and become a seeder. If we compare every single bits every time between every peer, the computational complexity will be unacceptable for a real time decision. Therefore, we first divide the whole content into three parts with identical length. Operation *count()* is defined to count the number of pieces that peer i has in each part.

$$count_i(a, B_i) = \sum_{j=\frac{aM}{3m}+1}^{\frac{(a+1)M}{3m}} b_j \tag{8}$$

$$b_j = \begin{cases} 1, \ if \ peer \ i \ has \ piece \ b_j \\ 0, \ otherwise, \end{cases} \quad a = \{0, 1, 2\}, \ b_j \in B_i$$

where M is the content size and m is the piece size. B_i refer to the bitfield of peer i. In this case, the bitfield of a given peer i can be represented by a normalized triple: $t_i(a, B_i) = \left[\frac{count_i(0,B_i)}{M/m}, \frac{count_i(1,B_i)}{M/m}, \frac{count_i(2,B_i)}{M/m}\right]$ Thus, any triple can be mapped to a vector in a 3D space $D(x, y, z)$ and each axis in this 3D space is the number of pieces(normalized) that peer i has in part a. In this case, each bitfield can be represented in quaternion form as pure quaternions in Eq. 9.

Using quaternion representation, these three parts will be processed equally in operations. Therefore, we do not have to process each part independently, but rather, treat all parts of bitfield as a whole unit. The advantages of quaternion representation will be further discussed in a latter example in this section.

$$q_i = \left(\frac{count_i(0, B_i)}{M/m}\right) i + \left(\frac{count_i(1, B_i)}{M/m}\right) j + \left(\frac{count_i(2, B_i)}{M/m}\right) k \tag{9}$$

Since the objective of a BT client is to get a full copy of the content, where $\frac{count_i(0,B_i))}{M/m} = \frac{count_i(1,B_i))}{M/m} = \frac{count_i(2,B_i))}{M/m} = 1/3$. This requirement denotes a vector at the diagonal in $D(x, y, z)$. Therefore, we can give the polar form of this quaternion with a rotation angel θ as:

$$R = e^{\mu\theta/2} = Cos(\theta/2) + 3Sin(\theta/2) * 1/3 \tag{10}$$

A three-space rotation about the axis μ through angel θ is represented by the quaternion operator

$$R[\ ]\bar{R} \tag{11}$$

where $R = e^{\mu\theta/2}$ is the objective vector of BT clients and $\bar{R}$ is the inverse of R. The square brackets indicate a space for the quaternion quantity to be operated upon. This operator is employed to rotate a given bitfield in this paper. According to Coxeter [26], who gives a proof of this action, it was first published by Cayley [27] in 1845. Therefore, we can get the π rotation of q_i by operation $q_i' = R[q_i]\bar{R}$, where $\theta = \pi/2$. This three space rotation can be visualized in Fig. 18. In this figure, the quaternion q is rotated by angle π perpendicular to the axis μ. When we represent bitfields in this way, a three-space rotation as given by Eq. 11 will always map a normalized bitfield value into another in three-space, but not necessarily with in the bounds of the $D(x, y, z)$.

We now discuss the reason why we want to process this rotation. Our basic assumption is that, the objective of BT peers is to get a full copy of the content, where $\frac{count_i(0,B_i))}{M/m} = \frac{count_i(1,B_i))}{M/m} = \frac{count_i(2,B_i))}{M/m} = 1/3$. Thus, every peer has the tendency to balance the ratio of each part during the downloading. A specific problem is that: For a given peer i who has a full copy of part one but no piece at all in the remaind parts (the triple is $[1/3, 0, 0]$), what

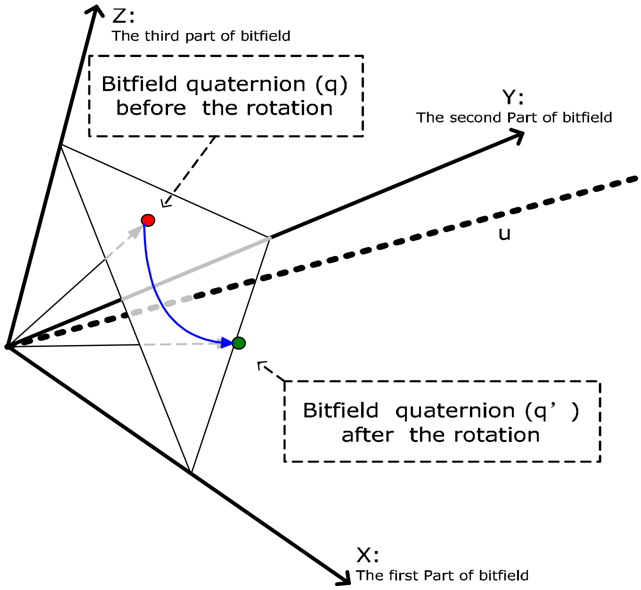

Fig. 18 Rotation of bitfield in three space

is the *ideal neighbor* of peer i (the neighbor who can serve peer i for the longest duration)? An intuitive answer should be a peer j with bitfield [0, 1/3, 1/3]. However, the exchange between BT peers is bidirectional, this relationship will end as soon as peer j obtained a whole copy of the first part. Fortunately, we can use a simple three space rotation to provide a better solution. In a more intuitive example of Fig. 19, we can find the triple

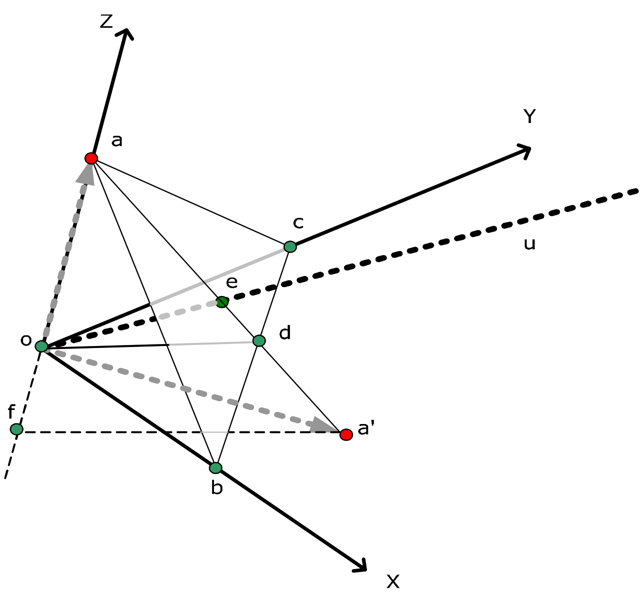

Fig. 19 The ideal neighbor of triple [0,0,1]

of the ideal neighbor using basic solid geometry rules as follows:
In Fig. 19, the π rotation of vector $\overrightarrow{oa}$ perpendicular to axis μ is given at $\overrightarrow{oa'}$. According to [26], $\overline{oa} = \overline{oa'}$ and $\overline{ae} = \overline{ea'}$. $\angle aeo = \pi/2$ is the angle between $\overline{oe}$ and plane abc. Node f denotes the value of $\overrightarrow{oa'}$ on z-axis and $\angle ofa' = \pi/2$. Therefore, we have:

$$
\begin{aligned}
\overline{ae} &= 2/3 * \overline{ad} \\
&= 2/3(\overline{ab} * cos\angle bad) \\
&= 2/3(\overline{oa}/sin\angle oab)cos\angle bad \\
&= \frac{\sqrt{6}}{3}\overline{oa}
\end{aligned}
$$

Since triangle Δaoe and $\Delta aa'f$ are similar triangles, we can get:

$$cos\angle oae = \overline{ae}/\overline{oa} = cos\angle faa' = \frac{\overline{ao} + \overline{of}}{2 * \overline{ae}} \qquad (12)$$

Thus, $\overline{of} = \frac{1}{3}\overline{oa}$. Without loss of generality, we let the length of $\overline{oa} = 1$ and $\overrightarrow{oa}$ denote the triple of [0,0,1] for peer i. Therefore, the value of $\overrightarrow{oa'}$ on z-axis is $-\frac{1}{3}$. We can compute another two values of this triple similarly and get the triple of peer i' as: $[\frac{1}{3}, \frac{1}{3}, -\frac{1}{3}]$. This example means that, for a given peer [0, 0, 1] the ideal neighbor should have bitfield similar to $[\frac{1}{3}, \frac{1}{3}, -\frac{1}{3}]$. In this case, the serving time is maximized in both sides between this leecher pair. Note that negative values are not existed in real bitfields, these theoretical values only representant the preference of the peer itself.
Above is just an intuitive example. In more general cases, the three space rotation of quaternion can give us an very clean and convenient form of the 'ideal neighbor', and help us with the comparison of different bitfields.

4) *Quantify the distance between different bitfields* Using quaternion based rotation, we are able to treat all parts of bitfield as a whole unit and find the position of the ideal neighbor very quickly. For example: If there are four BT peers $\{p_1, p_2, p_3, p_4\}$. The bitfields of these peers are $\{B_1, B_2, B_3, B_4\}$. We first generate a *bitfield belt* with all possible combinations of peers. $Q = \{(B_1, B_2), (B_1, B_3), (B_1, B_4), (B_2, B_3), (B_2, B_4), (B_3, B_4)\}$. (As shown in the topper part of Fig. 20.) In order to compute the relationship between these bitfields, a filter process is defined:

$$Q' = \begin{bmatrix} 1 \\ R \end{bmatrix} [Q] \begin{bmatrix} 1 \\ R \end{bmatrix} \qquad (13)$$

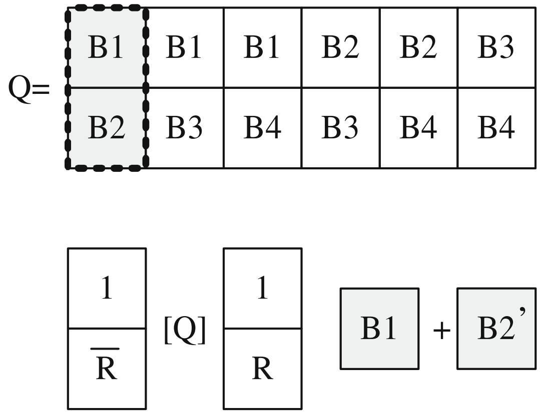

Fig. 20 Filter process on the bitfields [0,0,1]

As shown in Fig. 20, after this process, the filter in the top rows will keep bitfield values as where they are, whereas the bottom rows rotate the values by π perpendicular to the axis μ. This kind of filter operations is proven to be efficient and already applied to many edge detectors in computer vision studies [28, 29]. After this process, if bitfield pair (B_1, B_2) is very similar, the output will be very near the axis μ; Otherwise, the output will be far from axis μ. Assuming $q = xi + yj + zk$ is an output quaternion, a purity function [30] can be used to measure the distance between this output and axis μ:

$$C = \frac{x + y + z}{3} - min(x, y, z) \tag{14}$$

4.3 A possible extension to the BT streaming client

It is well known that BT economy is mainly based on 'pure exchange'. This exchanging process can only happen between two peers that are both interested in each other. Therefore, before the BT peer achieve the steady state[Qiu], data exchange efficiency of the peer is vary limited; This is intuitive because the peer has few local pieces to share. Moreover, the discussion in section III also shows that the efficiency of the possible BT streaming peers is even scarce.

In order to address this problem and take advantage of the unused bandwidth at the beginning of the downloading, a visual channel mechanism is proposed based on the new bitfield organization method. The main idea of this approach is that, instead of sending local bitmap, a peer will forward bitmaps between a set of leecher pairs at the beginning of the downloading.

Following steps will be applied on the peers who require streaming service: 1) After start, connect to the tracker and request for neighbor information. 2) Collecting the bitmaps within the peer group. 3) Organize these bitmaps into pairs and forward bitmaps between each pair using local IP address. 4) Forwarding the piece request and reply data between leecher pairs using local IP address. Note that the preference of seeder in BT networks is independent of local piece information in nature, above steps are only designed to process leechers. Moreover, one optimistic unchoke is needed to trigger the data exchange.

The principle of this approach that, if we build a visual channel between a leecher pair through the local peer, the preference of these leechers will be independent of the local piece information. In this case, if we select leecher pairs carefully according to their bitmaps, the local peer may never be choked due to the remote piece preference. Moreover, the data exchange efficiency can also be improved. Fortunately, the quaternion based comparison method can help us with the peer selection.

After the process of the purity function Eq. 14. Following steps are applied: 1) We choose the peers which have the maximal value of the outputs(C) as the first leecher pair. 2) Delete values that include first leecher pair within all outputs. 3) Choose the maximal value in the remaining part as second leecher pair and so on. The whole process will stop if there is less or equal than one peer remaining.

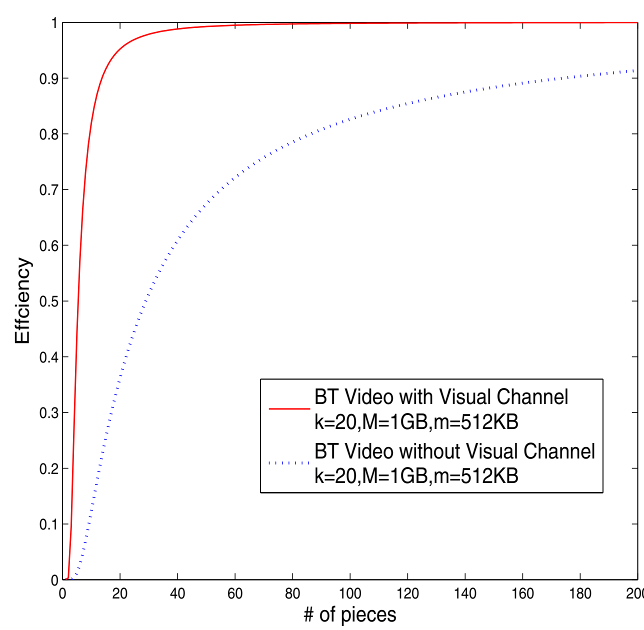

Fig. 21 Efficiency of visual channel based mechanism

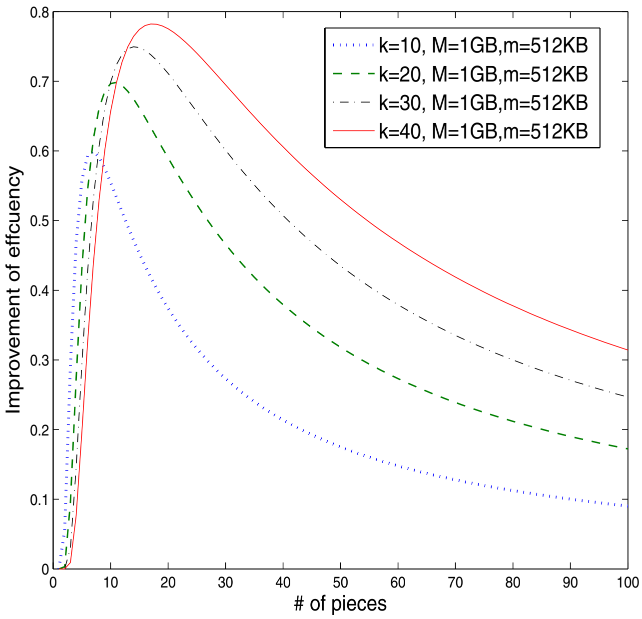

Fig. 22 Improvement of efficiency in different peer sets

5 Evaluation and discussion

According to the definition in the previous section, the main idea of our approach is to reduce the preference bias within a given peer set. Since local peers will only forward control and data message between leecher pairs, this solution is friendly to the existing BT protocol.

As we have already discussed in Section 3, the efficiency of BT video file swarm(shown in Eq. 1) is a very important metric. A local peer can hardly get all its neighbors' interests at the same time, especially at the beginning of the downloading. Based on our solution, the peer's preferences are no longer depending on local piece information. For a given peer, unless all the peers in its peer set are not interested with each other, our approach can always improve the efficiency of such a system. This improvement is shown in Fig. 21 using the assumption of a typical BT swarm (k = 20, M = 1 GB and m = 512 KB). Based on these parameters, We plot the data exchange efficiency of the system according to Eq. 1. Note that for the efficiency of the original video swarm, the x-axis indicates the number of pieces that the local peer has. On the other hand, for our virtual channel based solution, the x-axis indicates the average piece number of leecher pairs. We further present the improvement of efficiency with different peer set sizes in Fig. 22. We can find that the improvement of efficiency is highly related with the peer set size. This improvement will be limited if the local peer only have few neighbors. Fortunately, we have already indicated that the peer set size is acceptable in most BT video file swarms (in Section 3). Therefore, in most video swarms the improvement of our mechanism can be guaranteed. This figure also shows that, our solution is especially important at the beginning of the downloading. During this period (mostly before the local peer has less than 30 pieces), downloaders are very likely to be choked by other peers due to the preference bias. Therefore, make the preference independent of local information is a better choice. On the other hand, when the local peer has enough pieces of its own, this improvement will decrease smoothly as the abundance of local bitfield.

In order to validate the performance of quaternion based representation and the matching of leecher pairs, we built a simulator based on Quaternion Toolbox for Matlab [31] which is a proprietary software system for calculating with matrices of real and complex numbers. According to our pervious measurement results, we set the length of bitfields to 900 and fix the leecher number to 4 [7]. A large number of leechers can easily be computed by quaternion operations, but also can hardly provide intuitive rules in the table. Both inputs and outputs are shown in Table 4, where the rows in the table refer to the index of experiments and the columns refer to different bitfields (and the matching values of possible leecher pairs). In order to describe the number of pieces in each part, all these values are not normalized by M/m. The gray areas in the output table are the leecher pairs selected by our mechanism. We can learn that, the total number of local pieces are potentially included in the quaternion operations. The leechers are more likely to be matched with another who has similar capability. A poor leecher can hardly march to a rich one in our simulation. Moreover, we

Table 4 Inputs and outputs

IN	A	B	C	D
E_1	55,187,9	201,52,178	71,94,143	209,225,31
E_2	234,252,221	34,0,200	113,45,130	54,204,125
E_3	0,15,8	93,0,15	86,118,39	10,15,12
E_4	200,0,0	196,27,247	237,229,109	15,70,31
E_5	17,20,0	24,0,0	6,4,45	7,51,4

OUT	A-B	A-C	A-D	B-C	B-D	C-D
C_1	80.6	24.3	33.3	27.3	59	70
C_2	18.6	24.6	24.6	80.6	33	47
C_3	25	20.3	3	27.6	23	14
C_4	72.6	34.3	55.6	98.3	74.6	41.3
C_5	8	37.6	7.3	7.3	13.3	21.6

also observe that two kinds of peers always have relatively bad matching values: leechers who have very few pieces (peer A in E_3) and leechers who have almost a full copy of the content (peer A in E_2). This is intuitive because we are trying to find the long-term relationship between leechers. However, both are obviously not capable for that. On the other hand, if two leechers can meet each other's requirements, even poor leechers can have reasonable matching value in some cases. For example, peer A and C in E_5 are both poor leechers, but the matching value of A-C can also reach 37.6.

6 Conclusions and future research

In this paper, we studied the existing video file swarms in the BitTorrent system as regards the capability of providing streaming service. We for the first time examined the problem through a large scale Internet based measurement and especially focused on the video contents. Although current BT swarms may no be our best choice to provide streaming service, the strong requirement of Internet users motivates us to improve the capability of such a system. Using a hypercomplex based virtual channel mechanism, peers' preferences are no longer depending on the local pieces. It can help peers to make good use of existing peer information and obtain higher efficiency.

A distinguishing feature of our study in comparison to previous works is the focus on real-world measurement between different BT contents. Characteristics are discussed in swarm level rather than peer itself. The quaternion approach is also applied to solve such a problem for the first time. Meanwhile, one of its limitations is that the μ-axis is fixed during the whole process. This means all pieces are of the same importance for local peers. Since this assumption is not true in streaming applications, the dynamic adjustment of μ is necessary. For example, in order to give higher priority to pieces in the first part, μ-axis could be adjusted to close x-axis in $D(x, y, z)$. Obviously, the orientation of μ-axis may depend on the specific piece requirement of local peers. This issue is very important for streaming applications and we deserve them for future research.

References

1. AL-Hamra A, Legout A, Barakat C (2007) Understanding the properties of BitTorrent overlay. INRIA-00162088, version 1, Sophia Antipolis, France, 12 July 2007
2. Neglia G, Reina G, Zhang H, Towsley D, Venkataramani A, Danaher J (2007) Availability in bittorrent systems. In: Proc. IEEE INFOCOM, Anchorage, Alaska, USA, 6–12 May 2007
3. Yang X, de Veciana G (2004) Service capacity of peer to peer networks. In: Proc. IEEE INFOCOM, Hongkong, China, 7–11 March 2004
4. Liu J, Rao SG, Li B, Zhang H (2008) Opportunities and challenges of Peer-to-Peer internet video broadcast. Proc IEEE 96(1):11–24 (Special issue on recent advances in distributed multimedia communications)
5. Vlavianos A, Iliofotou M (2006) Faloutsos M BiToS: enhancing bittorrent for supporting streaming applications. In: Proc. IEEE INFOCOM, Barcelona, Spain, 23–29 April
6. Dana C, Li D, Harrison D, Chuah C (2005) Bass: Bittorrent assisted streaming system for video-on-demand. In: International workshop on multimedia signal processing (MMsP). IEEE Press
7. Zhang X, Liu J, Li B, Yum TSP (2005) CoolStreaming/DONet: a data-driven overlay network for Peer-to-Peer live media streaming. In: Proc. IEEE INFOCOM, Miami, FL, USA, 13–17 March 2005
8. Logout A, Liogkas EKN, Zhang L (2007) Clustering and sharing incentives in BitTorrent systems. In: Proc. ACM SIGMATERICS
9. Dale C, Liu J, Peters JG, Li B (2008) Evolution and enhancement of BitTorrent network topologies. In: Proc. IEEE IWQoS, Enschede, The Netherlands, 2–4 June 2008
10. Guo L, Chen S, Xiao Z, Tan E, Ding X, Zhang X (2004) Dissecting BitTorrent: five months in a torrent's life time. In: Proc. the 5th ACM SIGCOMM internet measurement workshop
11. Izal M, Urvoy-Keller EBG, Fellber P, Hamra AA, Garces-Erice L (2004) Measurement, analysis and modeling of BitTorrent-like systems. In: Proc. the 5th passive and active measurement workshop, Antibes Juan-les-Pins, France, 19–20 April 2004
12. Logout A, Urvoy-Keller, Michiardi P (2005) Understanding BitTorrent: an experimental perspective. INRIA, Tech, Rep. 00000156
13. Fessant FL, Kermarrec AM, Massouli L (2004) Clustering in Peer-to-Peer file sharing workloads. In: Proc. IPTPS, La Jolla, CA, USA, 26–27 February 2004
14. Guo L, Chen S, Xiao Z, Tan E, Ding X, Zhang X (2005) Measurements, analysis, and modeling of BitTorrent-like systems. In: Proc. ACM/USENIX IMC
15. Sen S, Wang J (2002) Analyzing Peer-to-Peer traffic across large networks. IEEE/ACM Trans Netw 12(2):219–232
16. Planetlab. Available at: http://www.planet-lab.org/. Accessed 3 Aug 2008
17. Ctorrent. Available at: http://ctorrent.sourceforge.net/. Accessed 3 Aug 2008
18. Tewari S, Kleinrock L (2007) Analytical model for BitTorrent-based live video streaming. In: Proc. IEEE CCNC, Las Vegas, Nevada, USA, 11–13 January 2007
19. Marciiak P, Liogkas N, Legout A, Kohler E (2008) Small is not always beautiful. IPTPS 2008, Tampa, FL, USA, 25–26 February 2008
20. Huang Y, Fu TZ, Chiu JD, Lui JCS, Huang C (2008) Challenges, design and analysis of a large-scale P2P-VoD system. In: Proc. ACM SIGCOMM, Seattle, WA, USA, 17–22 August
21. Hei X, Liang C, Liang J, Liu Y, Ross KW (2007) A measurement study of a large-scale P2P IPTV system. IEEE Trans Multimedia
22. Vu L, Gupta I, Liang J, Nahrstedt K (2007) Measurement and modeling a large-scale overlay for multimedia streaming. In: Proc. the 4th international conference on heterogeneous networking for quality, reliability, Vancouver, British Columbia, 14–17 August 2007

23. Qiu D, Srikant R (2004) Modeling and performance analysis of bit torrent-like Peer-to-Peer networks. In: Proc. ACM SIGCOMM
24. Hamilton WR (1866) Elements of quaternions, 3rd edn. Chelsea Pub Co. (March 1969). Longmans, Green and Co., London
25. Argand R (1874) Essai sur une manire de reprsenter les quantits imaginaires dans les constructions gomtriques. Gauthier-CVillars, pp 112 (Published by G. J. Hoel in 1874. First edition published Paris, 1806)
26. Coxeter HSM (1946) Quaternions and reflections. Am Math Mon 53:136–146
27. Cayley A (1845) On certain results relating to quaternions. Philos Mag 26(3):142
28. Astola J, Haavisto P, Neuvo Y (1990) Vector median filters. Proc IEEE 78(4):678–689
29. Sangwine SJ (1998) Color image edge detector based on quaternion convolution. Electron Lett 34(10):969–971
30. Pritchard AJ, Horne REN, Sangwine SJ (1995) Achieving brightness-incensitive measurements of color saturation for use in color object recongnition. In: Proc. the 2rd IMA confernce on image processing and its applications
31. Quaternion Toolbox for Matlab Available: http://sourceforge.net/projects/qtfm/. Accessed 3 Aug 2008

Haiyang Wang is currently 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.

Jiangchuan Liu (S'01-M'03-SM'08) 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.