Performance Analysis of BitTorrent and its Impact on Real-time Video Applications

Amuda James Abu and Steven Gordon

Sirindhorn International Institute of Technology, Thammasat University, Pathumthani 12000, Thailand
james@ict.siit.tu.ac.th, steve@siit.tu.ac.th

Abstract. BitTorrent and similar peer-to-peer file sharing applications can represent a large portion of network traffic. Despite the advantages for BitTorrent users, it can unfairly consume access link bandwidth from other user(s) and applications. It can also rapidly fill up buffers at access routers. We have used a detailed model of the BitTorrent protocol to analyze its performance and impact on real-time video traffic. We have shown that increasing the number of BitTorrent clients and/or upload connections can cause a decrease in download rate due to delayed TCP acknowledgements. We also show the effect of access router buffer size on performance: too small reduces BitTorrent's upload rate, too large increases video jitter and delay.

Keywords: BitTorrent, TCP, Peer-to-Peer File Sharing, Real-Time Video

1 Introduction

BitTorrent [1] is a popular protocol for peer-to-peer (P2P) exchange of data, such as file sharing. The BitTorrent protocol allows a client to download portions of a file from different remote hosts, thereby avoiding dependence on a single server and potentially decreasing the total download time. To ensure there are sufficient remote hosts to download from, BitTorrent requires a downloading client to also upload files.

The popularity and efficiency of P2P file sharing have resulted in performance problems for end-users and Internet Service Providers (ISPs). Estimates of the portion of all traffic contributed by P2P file sharing range from 40% to 70% [2, 3]. This presents problems for end-users because of the way in which the protocols interact with other applications, such as increasing the delay experienced when web browsing. In addition, the large amount of data uploaded by end-users presents challenges for ISP networks, traditionally engineered for a high download/upload ratio.

In this paper we analyze the performance of BitTorrent and investigate its impact on interactive video traffic in an ISP network. BitTorrent aims to maximize the download rate for the end-user. However BitTorrent implements a tit-for-tat strategy to ensure sufficient amount of data is uploaded so that there are enough remote hosts to download from. Therefore, maximizing the upload rate is also important for BitTorrent. For interactive video traffic, delay and jitter should be minimized, while a small number of packet drops can be tolerated.

When there are many end-users in an ISP network, the access router that the users connect to, and in particular the uplink from the access router to the next router, may become the bottleneck in the network [4, 5]. BitTorrent, which uploads a large amount of data using multiple TCP connections, may utilize a large portion of that link (and access router buffers), resulting in unacceptable delays for other applications (and non-BitTorrent users). This is particularly detrimental to real-time voice and video applications. The IETF, through the Low Extra Delay Background Transport (LEDBAT) Working Group [6], have also identified these problems for end-users and ISPs, and have recently begun analyzing the issues involved. The results in this paper are a step towards understanding the performance of BitTorrent and its impact on video applications, especially when BitTorrent users are sharing the same access link in an ISP network with a real-time video user. We show the relationship between BitTorrent clients, upload connections, and access router buffer size on delay, upload/download rates and packet drops for BitTorrent and video applications.

This paper is structured as follows: In Section 2, the BitTorrent performance issues are explained. Our assumed scenario is described in Section 3, and the analysis methodology and results are presented in Section 4 Related work is discussed in Section 5, and finally conclusions and future work are given in Section 6.

2 Performance Issues with BitTorrent and other Applications

BitTorrent is a protocol primarily used for P2P file sharing. In BitTorrent a file is referred to as a *torrent*. The set of *peers* downloading and/or uploading a torrent is called a *swarm*. A peer within a swarm that has fully downloaded the torrent and makes it available to others is a *seed*, while those yet to download the entire torrent are *leeches.* Consider a peer that wants to download a torrent. We will refer to it as the *local peer* and others as *remote peers*. Using a *tracker* server, the local peer discovers a list of remote peers in the swarm, selects N remote peers and establishes TCP connections with each. Then the peers use the *Peer Exchange Protocol* to exchange pieces (i.e. upload and download the file). The strategy for selecting peers to exchange pieces with is important for maximizing the download rate, as well as uploading content for other peers to access. Of the N remotes peers that the local peer is connected to, it will choose U unchoked peers to exchange pieces with. The local peer chooses these unchoked peers from those that it has the best download rates from, and the peers are interested in exchanging pieces with the local peer. Regular updates of choked/unchoked peers are performed, as well as occasional optimistic unchoking in the hope of maximizing the download rate. Further details can be found in the official [1] and unofficial [7] BitTorrent specifications.

Although using multiple TCP connections can increase reliability and distribute traffic load to multiple peers, it can have impact on how link bandwidth is shared among applications and users. In ideal conditions TCP will share the bandwidth of a link equally among the connections. Application X with M TCP connections sharing a link with an application Y with one TCP connection will obtain M times the bandwidth as application Y. Considering an end-user running multiple applications (e.g. BitTorrent, web browsing, file download), the user may experience unfairness in the link bandwidth allocation for each of the applications because of the different number of TCP connections established by each of the applications (e.g. BitTorrent resulting in excessive web response times). However, the end-user has control over this unfairness – the user can manually choose the applications to run, or configure their host to give priority to desired applications. Now consider multiple end-users connecting to an ISP access router as shown in Fig. 1a. If the access router uplink (to the next router) is the bottleneck in the path, then unfairness may arise, this time outside the control of the end-user. Because the access link is a bottleneck, P2P client is capable of rapidly filling up the queue of the access router. Therefore, packets from the video source experience longer queue delay at the access router and they are possibly dropped when the queue is full, especially when the access router uses a drop-tail queue discipline. Large delay and high packet drops can be detrimental to real-time video applications.

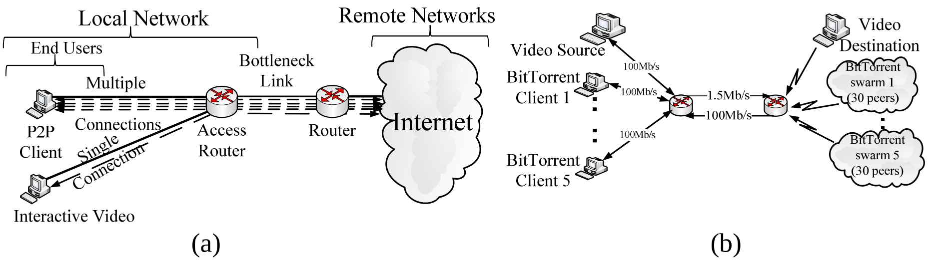


Fig. 1. Showing (a) the sharing of access link among end-users running different applications with different number of connections, and (b) the simulation network topology

Although unfairness between any application with multiple TCP connections and application with single TCP connection may arise (e.g. web browsers use multiple connections), the issue is especially relevant for BitTorrent (and other P2P protocols) for the following reasons:

1. BitTorrent uploads a large amount of data unlike other applications, with the exception of interactive voice/video applications which typically use UDP.
2. BitTorrent local peer may change TCP connections to remote peers on a regular basis (depending on the choking algorithm and availability of peers). This results in changes in TCP parameters (e.g. window sizes) and performance when compared to an application that always uses the same multiple connections for the entire duration of the data transfer.
3. The traffic profile of BitTorrent (data packet sizes, frequency and size of control packets being sent) differs from other applications.

Therefore analysis of the performance of BitTorrent is needed in the access network of an ISP network from the end-user perspective, especially in the presence of other applications used by other end-users in the same access network. We consider the impact of the access router capabilities on BitTorrent under different loads, as well as the interactions with video traffic.

3 Scenario Description and Simulation Setup

We assume an ISP network with *L* local hosts (i.e. customers) all with dedicated links to the ISP access router. The uplink from the access router to the next router is the bottleneck link in the path for end users. This scenario arises when there are a large number of local hosts, each with reasonable uplink speeds, but the uplink speed from access router is insufficient for all hosts uploading at the same time. Beyond the ISP network are *R* remote hosts. To ensure a very high percentage of peer-to-peer traffic in the access network of the ISP network, we assume *L-1* local hosts are running BitTorrent while one local host is running an interactive video application.

The network simulator ns2 [8], with a BitTorrent patch [9], is used to analyze the performance of BitTorrent and its impact on interactive video application using the topology shown in Fig. 1b. The capacity of the access router uplink to the next router is set to 1.5Mb/s while the downlink is set to 100Mb/s. We use these values because we are interested in making the uplink a bottleneck. Other links are set to 100Mb/s in order to congest the access network especially from the local end-users perspective. The delays of all links randomly (uniformly distributed) range from 1-50ms. All routers use drop-tail queue discipline. The default parameter values for the network and applications are shown in Table 1. Queue size of the access router is set high because we want to avoid early packet drop at the access router.

Table 1. Default parameter values

Parameter	Value	Parameter	Value
Local BitTorrent clients	1	*BitTorrent Application*	
Remote peers per swarm	30 per local client	File size	100 MB
Access router queue size	300 pkts	Unchoked connections	4
Link MTU	1500 Bytes	Unchoke interval	10 sec
Video Application		Piece size	256 KB
Data rate	750 kb/s	Block size	16 KB
Packet size	500 B		

TCP New Reno is the transport protocol used by BitTorrent. One of the remote peers is the initial file seed, and each peer remains in the swarm until all other peers have finished the download. This topology (similar to [4]) is chosen to allow the local peer to select from sufficient remote peers and to generate significant traffic on the local link. The video traffic is constant bit rate using UDP. The data rate and packet size are chosen to reflect a good quality video conversation over the 1.5Mb/s uplink.

Two different simulation configurations were carried out: Firstly, BitTorrent traffic with no video session and secondly, BitTorrent traffic with 1 video session in the network. Numerous statistics were collected from the simulations. For brevity, we present the following in this paper: aggregate Uploading and Downloading rate of all local BitTorrent peers; Packet Delay and Packet Drops at access router queue; Inter-arrival time for receiving video client (i.e. an indicator of video jitter). All statistics shown are the average of 10 simulations with different random seeds in each simulation configuration.

4 Analysis and Results

4.1. Number of Local BitTorrent Clients

First we consider the impact of varying the number of local BitTorrent clients from 0 up to 5. Results for selected performance metrics with and without the video session are shown in Fig. 2. Fig. 2a shows the aggregate upload rate – the sum of the upload rates for all local peers. With one local peer in the absence of video traffic in the network, the rate is approximately 900kb/s, whereas with two local peers the rate per peer is approximately 575kb/s (a total of 1150kb/s). The aggregate upload rate reaches 1200kb/s. This is limited by the access link rate of 1.5Mb/s (and packet and TCP overheads). However, with a single BitTorrent client, the upload rate is not limited by the access link capacity, but rather by the demand for pieces from the remote peers. In the presence of video traffic, the aggregate BitTorrent upload rate is reduced as the TCP connections across the access router uplink must now compete with the UDP traffic from the video source.

The increasing upload rate of local BitTorrent clients is responsible for an increasing queue delay of packets at the access router. The results in Fig. 2b show that the video packet queue delay at the access router is 0ms with no BitTorrent traffic in the network, but rises sharply as BitTorrent traffic is introduced. Large delay is undesirable for any interactive video application. Large queue delay of BitTorrent packets can affect BitTorrent download rate as TCP ACK packets are delayed.

Arriving packets are frequently dropped when the queue of the access router is quickly filled to its limit (Fig. 2c). BitTorrent packets with 1 video session experience the highest drop rate due to the increasing queue delay and the unavailability of the entire bandwidth of the access router uplink. The reduced drop rate of BitTorrent packets with no video session is due to the availability of the full capacity of the access router uplink. When we have only video traffic present in the network (i.e. 0 BitTorrent clients), no packet is dropped. However, as we introduce BitTorrent traffic, video packets are dropped at the access router. High video packet drop rate is undesirable for the real-time video application. However, BitTorrent packets dropped are re-transmitted by TCP, leading to higher load on the access router.

Recall that the downlink rates are effectively unlimited compared to the upload link from access router. The aggregate download rate of local BitTorrent client(s), with and without video session in the network, increases with increasing number of local BitTorrent clients (Fig. 2d). With one local peer the download rate is approximately 2000kb/s, and with two 4000kb/s (2000kb/s per peer). As the local peers are part of independent swarms, it could be expected to see each peer maintaining 2000kb/s download rate. However, this is not the case with 3 or more peers (e.g. with 5 peers less than 8000kb/s). This can be explained by the delayed TCP acknowledgement packets in the queue of the access router. The sending rate of a remote peer (and hence downloading rate of local peer) is limited by the rate at which the remote peer receives TCP ACKs from the local peer.

The video source generates a packet every 5.33ms. Performance is degraded for the interactive video application when the video receiver inter-arrival time increases compared to 5.33ms as the number of local peers increases as shown in Fig. 2e. This is due to the variation of queue delay of video packets at the access router.

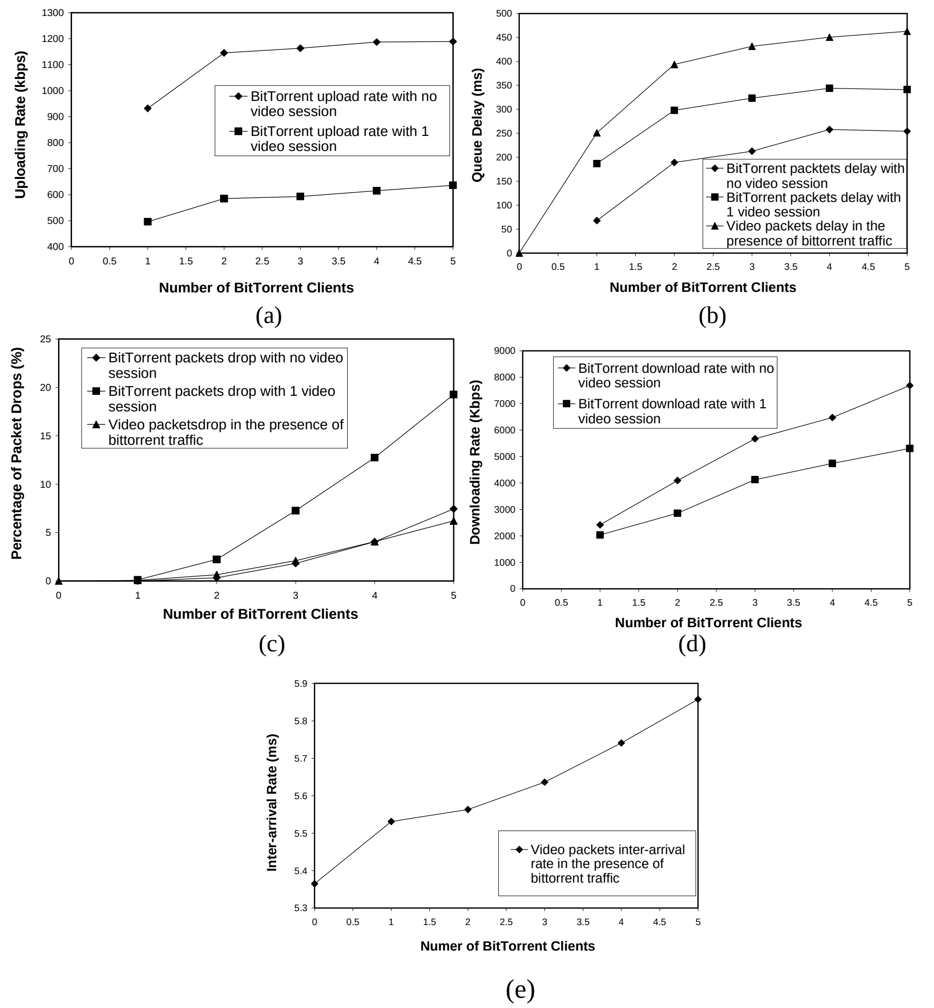


Fig. 2. Effects of varying the number of local BitTorrent clients on: (a) BitTorrent upload rate, (b) queue delay of BitTorrent and video packets, (c) percentage of BitTorrent and video packets drops, (d) BitTorrent download rate; with and without video session in the network, and (e) Inter-arrival rate of video packets at the remote host (receiver).

4.2 Number of Unchoked Remote Peers

Now we consider the impact of varying the number of upload connections (unchoked remote peers) from 4 to 20 with a single local BitTorrent client. As shown in Fig. 3a the uploading rate of the local peer (with 1 and zero video session) increases as a result of an increase in the number of multiple TCP connections used by the local peer to upload data to remote peers. An increased number of upload connections of the local peer implies an increased number of remote peers to download from at the same time. As a result, large portion of the file will have been downloaded within a short period of time which can also be uploaded to other interested remote peers.

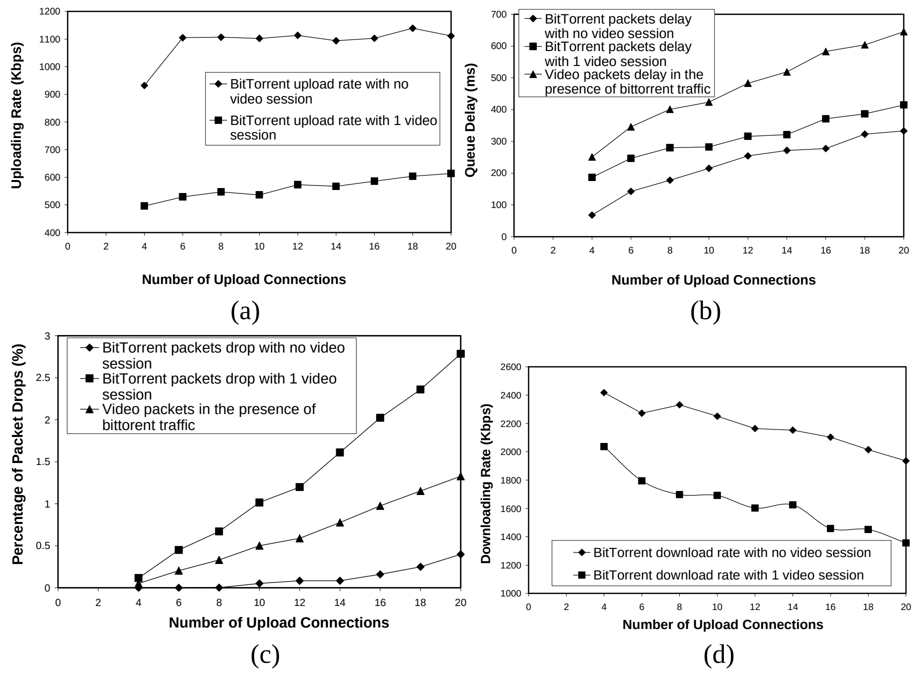


Fig. 3. Effects of varying the number of upload/unchoked connections of a single local BitTorrent client on: (a) BitTorrent upload rate, (b) queue delay of BitTorrent and video packets, (c) percentage of BitTorrent and video packets drops, and (d) BitTorrent download rate; with and without video session in the network.

The increasing upload rate is responsible for an increasing queue delay of packets at the access router (see Fig. 3b), with video packets experiencing longer delay which cannot be tolerated by real-time video applications, while coexisting with BitTorrent traffic in the network.

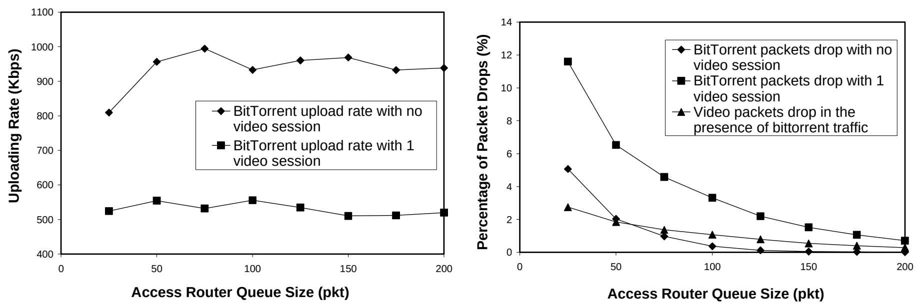
As the queue of the access router grows, it will be eventually filled up to the size limit. Therefore, arriving packets are dropped (Fig. 3c). This is responsible for the increasing percentage of packet drops at the access router. BitTorrent packets dropped are retransmitted by TCP. However, as video packets dropped become large, noticeable portion of the video packets become unavailable at the receiver.

The decreasing download rate for the local BitTorrent client (note that this is for a single client, compared to Fig. 3d which shows the aggregate for all clients) is due to the delayed TCP ACKs in the queue of the access router. 400ms and 300ms queue delays of BitTorrent packets (ACKs) are responsible for the minimum download rates of 1400kb/s and 2000kb/s with and without video session respectively with 20 unchoked connections. This is because the sending rates of remote peers depend on the rates at which they receive ACK packets from the local peer.

4.3 Access Router Queue Size

Finally we obtain results when we vary the queue size of the access router from 25 to 200 packets with an increment of 25. Fig. 4 shows the effects of increasing the access router's queue size on the upload rate of a local peer as shown in Fig. 4a, and the percentage of BitTorrent and video packet drops as shown in Fig. 4b.

With no video traffic and queue size of 25 packets, the upload rate of BitTorrent is poor as the number of packets dropped is large as shown in Fig. 4b, leading to TCP retransmissions with each TCP connection, initiated by BitTorrent, halving its congestion window in response to packet loss, and low uploading rate. However, as queue size increases, the upload rate becomes better as less packets are dropped as shown in Fig. 4b and less retransmission. With video traffic, the Bittorrent upload rate becomes poorer as more packets are dropped. Large video packet drops can be unfriendly for the interactive video application. Low latency tolerance applications such as real-time video perform undesirably with increasing queue size (delay) in the presence of BitTorrent traffic.



(a) (b)

Fig. 4. Effects of access router queue size on: (a) BitTorrent upload rate and (b) percentage of BitTorrent and video packets drops; with and without video session in the network.

5 Related Work

Researchers have studied the traffic characteristics of actual P2P file sharing systems, mostly via trace analysis and experimentation, observing factors such as load generated on networks, distribution of pieces among peers, and distribution and activity of peers (e.g. [10, 11, 12]). Such studies provide high-level statistics on BitTorrent traffic across one or more swarms, but do not give insight into the impact of parameters on individual users and ISP networks.

The issues of multiple connections (i.e. parallel downloads) have been studied in the context of P2P file sharing [13] and other applications (e.g. web browsing [14, 15]). Although many issues are similar for non-P2P applications, our analysis is significant because it uses a detailed model of BitTorrent, including aspects of switching between TCP connections, as well as BitTorrent traffic.

An evaluation of the effects of P2P traffic on UDP, in particular voice, is given in [16]. The analysis focuses on wireless access networks, and presents a comparison of voice/P2P performance when QoS mechanisms are used. Although similar results with P2P affecting UDP are seen as in our paper, the effects of multiple connections and access router properties are not considered. It should also be noted that QoS control is not always possible in ISP networks.

ACCM [17] proposes a modified congestion control algorithm to improve fairness between P2P file sharing and other TCP applications. The results show considerable promise for ACCM, but focus only on TCP interactions. The authors are yet to consider real-time voice or video (UDP) traffic.

The IETF LEDBAT Working Group [6] has begun to review the impacts of P2P file sharing on ISPs and end-users. A qualitative analysis of the advantages and disadvantages of multiple TCP connections has been initiated, as well as discussion of transport protocols and congestion control mechanisms suitable for BitTorrent-like applications that improve fairness for other applications. Two promising techniques are Friendly P2P and Ledbat (from BitTorrent). Friendly P2P [4] is a proposed application-level modification of P2P protocols to provide improved fairness between P2P, FTP, and voice applications. Simulation analysis has shown fairness can be improved in the presence of FTP and UDP applications when a single P2P client with multiple connections is operating. Factors such as multiple clients, different number of connections have not been analyzed.

Ledbat [18] is a congestion control scheme used with some BitTorrent/uTorrent applications. The approach is for the local peer to measure delay, and reduce its sending rate before the access router buffer is full, allowing other applications to obtain a fair share of the access router uplink. Although used in real BitTorrent networks, no results or analysis has been reported.

6 Conclusions

In this paper we have analyzed the performance of the BitTorrent protocol and its impact on interactive video application. The analysis is significant as it is one of the first to model the detailed behavior of the BitTorrent protocol, and considers the effects of the access router capabilities, as well as the number of connecting clients in the access network of an ISP network. We have shown that the number of unchoked connections can contribute more to queuing delay than the number of BitTorrent clients. In addition, although upload rates can be increased, because of delayed TCP acknowledgements, the overall download rate reduces. This leads to an important design tradeoff when selecting the number of connections. Finally, the access router queue size has significant impact on application performance: too small can greatly reduce BitTorrent upload rate due to packet drops which leads to halving the congestion window of each TCP connections, and subsequently retransmission; too large will increase delay and jitter for video applications which can jeopardize the performance of such a real-time interactive application. An ISP may not be able to control the number of unchoked connections used by an end-user running BitTorrent application but it can control the number of end-users that mostly use BitTorrent application. Furthermore, an ISP can control the queue delay of packets at the access router by using routers in the access network whose queue sizes are optimal (i.e. not too large because of large delay and not too small because of high percentage of packet drops). As future work we will compare different congestion control mechanisms, both transport and application level, that can deliver fair and efficient access to all applications and users.

7 References

1. BitTorrent protocol specification, http://www.bittorrent.org/beps/bep_0003.html
2. Ipoque Internet Study 2008/2009, http://www.ipoque.com
3. The Youtube Effect, http://arstechnica.com/old/content/2007/06/the-youtube-effect-http-traffic-now-eclipses-p2p.ars
4. Y. Liu, H. Wang, Y. Lin, S. Cheng, and G. Simon: Friendly P2P: Application-level congestion control for peer-to-peer applications. In IEEE GLOBECOM, pp. 1-5 (2008)
5. A. Akella, S. Seshan, and A. Shaikh: An empirical evaluation of wide-area Internet bottlenecks. In Proc. ACM Conf. Internet Measurement, pp. 101-104 (2003)
6. IETF Low Extra Delay Background Transport Working Group, http://www.ietf.org/dyn/wg/charter/ledbat-charter.html
7. Unofficial BitTorrent specification, http://wiki.theory.org/BitTorrentSpecification
8. ns2, Network Simulator, http://www.isi.edu/nsnam
9. K. Eger, T. Hosfeld, A. Binzenhofer, and G. Kunzmann: Efficient simulation of large-scale P2P networks: Packet-level vs flow-level simulations. In Proc. Workshop Use of P2P, Grid and Agents for Development of Content Networks, pp. 9-16 (2007).
10. L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang: A performance study of BitTorrent-like peer-to-peer systems. In IEEE Journal on Selected Areas in Communications, 25(1), pp. 155-169 (2007).
11. R. Bharambe, C. Herley, and V. N. Padmanabhan: Analyzing and improving a BitTorrent networks performance mechanisms. In Proc. INFOCOM, pp. 1-12 (2006).
12. M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. A. Hamra, and L. Garc'es-Erice: Dissecting BitTorrent: Five months in a torrent's lifetime. In Proc. Passive & Active Measurement Workshop (2004).
13. S. G. M. Koo, C. Rosenberg, and D. Xu: Analysis of parallel downloading for large file distribution. In Proc. IEEE Intl. Workshop Future Trends in Distributed Computing Systems, pp. 128-135 (2003)
14. C. Gkantsidis, M. Ammar, and E. Zegura: On the effect of large-scale deployment of parallel downloading. In Proc. IEEE Workshop Internet Applications, pp. 79-89 (2003).
15. T. Loukopouos and I. Ahmad: Optimizing download time of embedded multimedia objects for web browsing. In IEEE Trans. Parallel Dist. Sys., 15(10), pp. 934-945 (2004).
16. D. Astuti and M. Kojo: Evaluating the behaviour of peer-to-peer IP traffic in wireless access networks. In Proc. Intl. Network Conf. (2005).
17. W. Li, S. Chen, Y. Liu and X. Li: Aggregate congestion control for peer-to-peer file sharing applications. In Proc. Intl. Conf. Software Engineering, AI, Networking and Parallel/Distributed Computing, pp. 700-705 (2008).
18. S. Shalunov: LEDBAT Congestion Control Algorithm. IETF Internet Draft, draft-shalunov-ledbat-congestion-00.txt, March 2009 (work in progress).