Revenue Maximisation in Peer-to-Peer Networks

Kolja Eger and Ulrich Killat

Department of Communication Networks
Hamburg University of Technology (TUHH)
Denickestr. 17, D-21073 Hamburg, Germany
{eger, killat}@tu-harburg.de

Abstract: BitTorrent is a peer-to-peer (p2p) protocol for file-sharing. Its improvement over other file-sharing protocols is its tit-for-tat strategy to decide to whom a peer should upload. This means that an upload to a peer depends on the download from that peer.
This paper investigates the BitTorrent upload algorithm and proposes a new algorithm which is based on a distributed optimisation problem where each peer maximises its own revenue. First simulation results show that the proposed algorithm outperforms the BitTorrent choking algorithm with respect to efficiency and fairness.

1 Introduction

Peer-to-peer (p2p) networks gained in popularity in recent years. In particular file-sharing applications are used in these overlay networks. The advantages are obvious. While in the server/client architecture the total load must be carried by the server(s), it is distributed across the users in a p2p network. Namely, each peer acts as a client and a server at the same time. But measurements show that the lack of incentives induces free-riding, where most of the users do not contribute to the network [AH00].
The BitTorrent protocol tries to overcome this with two ideas. Firstly, a file is fragmented into a number of smaller pieces, called chunks. When a peer completes the download of a chunk, it can already upload it to other peers and does not have to complete the whole file to share something. Secondly, each peer controls to whom it uploads data. This is called unchoking in BitTorrent. A peer uploads to the four peers with the highest upload rates to him measured every 10 seconds. When a peer has completed the download of the file, unchoking is based on the download rate of the connected peers rather than the upload rate. An exception is run every 30s, where a peer is unchoked independent of its rate. This is called an optimistic unchoke. Details about the BitTorrent protocol and architecture can be found in [Co03].
Measurement results indicate that the fairness and efficiency issue is not fully solved in a BitTorrent network. [IUKB$^+$04] observed high variability in the download rates of peers and that 81% of the peers did not complete their download at all. This motivates us to propose a new unchoking algorithm which is described in Section 2. Section 3 presents

first simulation results and compares both algorithms. The last section concludes the paper.

2 Revenue Maximisation Algorithm

The proposed algorithm is based on congestion pricing for IP networks [KMT98]. In that context it proved to be a robust and fair decentralised algorithm.
It is assumed that each peer behaves self-interested and maximises its own revenue. This is modeled with the help of a concave, non-decreasing utility function $U_p(x_{qp})$, which depends on the rate x_{qp} from peer q to peer p . Like in [KMT98] we choose $U_p(x_{qp}) = w_p \log x_{qp}$, where w_p is the willingness to share of peer p in this context. This utility function ensures a weighted proportional fair allocation. The optimisation problem of peer q is then

$$\begin{aligned} \text{maximise} \quad & \textstyle\sum_{p\in\mathcal{PS}} w_p \log x_{qp} && (1) \\ \text{subject to} \quad & \mathbf{A}\mathbf{x}_q \leq C_q^{\mathrm{up}} && (2) \\ \text{over} \quad & \mathbf{x}_q > 0, && (3) \end{aligned}$$

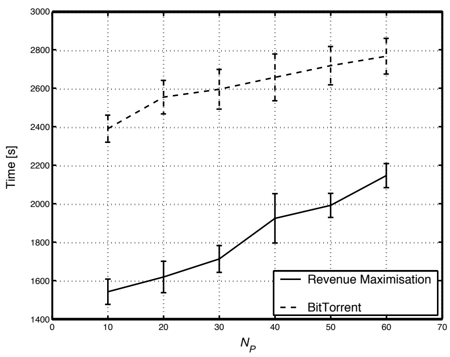
where $\mathcal{PS}$ is the set of peers with which peer q is connected. $\mathbf{A}$ is a 0-1 vector. It is 1, if peer p is in the peer set of q and is interested in a chunk of peer q and 0 otherwise. C_q^{up} is the upload capacity of q. The Lagrangian of this problem is

$$L(x,z;\lambda) = \sum_{p\in\mathcal{PS}} w_p \log x_{qp} + \lambda(\mathbf{C}_q^{\mathrm{up}} - \mathbf{A}\mathbf{x}_q - z), \tag{4}$$

where λ is the Lagrange multiplier and $z \geq 0$ a slack variable. To solve the problem numerically we use the Gradient Projection Algorithm [Lu03] and compute rates with

$$x_{qp}(t+1) = \left[x_{qp}(t) + \gamma \frac{\partial L}{\partial x_{qp}}(\mathbf{x}_q)\right]^+, \tag{5}$$

where γ is a positive stepsize.
We interpret the willingness to share of a peer as its upload rate measured in an interval of time and thus set $w_p = x_{pq}$ in (1). This allows the control of the upload rate for active connections where rates are non-zero, i.e. $x_{pq} > 0$ and $x_{qp} > 0$. To open new connections we propose a probing algorithm which is called when the peer is not satisfied with its download rate. It is willing to open a new connection if its download performance did not increase in the last time interval and $\sum_{p\in\mathcal{PS}} x_{pq} < \alpha C_q^{\mathrm{up}}$, where α is a satisfaction factor. We choose α near to 1 since a peer expects from a fair p2p network to receive approximately as much as it uploads. Each new connection is assigned a small portion of bandwidth at the beginning. This should avoid rapid rate fluctuations for the other connections.
Using the download rate as the willingness to share requires to formulate an exception in the presented algorithm for peers, which have completed the download. The optimal behaviour of these peers would be to maximise social welfare, which can be interpreted as



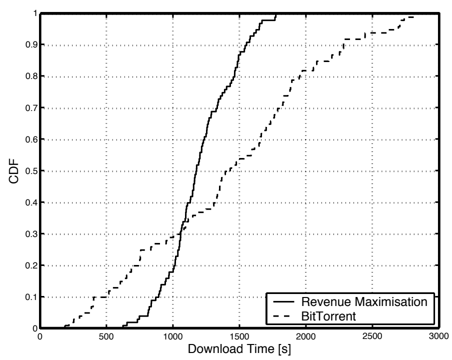


Figure 1: Overall download time for a flash crowd with N_P peers

Figure 2: CDF of the download times with Poisson Arrivals ($\lambda = 100$)

the overall upload rate of the whole p2p network. We try to approach such a behaviour by uploading to the peers which have the fewest number of chunks. This scheme allows new peers to complete the first chunk and actively participate in the network afterwards.

3 Simulation Results

We simulated BitTorrent-like networks with the packet simulator ns2 and compared both choking algorithms. Thereby, it is assumed that the upload capacities of the peers are the bottlenecks in the network and download capacities are set 8 times higher. We justify this with the fact that Internet connections are often asymmetric, e.g. ADSL, and users limit their upload rate for p2p applications. For the here presented simulations peers' upload capacities are set to $C^{up} = 10\text{KB/s}$. Furthermore no packet losses are considered. The end-to-end delay between peers are determined randomly between 25-100ms. The download of a 10MB file is simulated, where each chunk is of size 256KB. At the beginning of each simulation only one peer holds the complete file.

In the first simulation a number of peers enter the network at the same time. This is called a flash crowd. No peer leaves during or after download. The time until all peers have finished their download is measured. Figure 1 shows the overall download time for different number of peers and the 95% confidence intervals. The overall download time for the peers using the revenue maximisation algorithm is smaller than with the original BitTorrent algorithm.

In the second simulation peers enter the network as a Poisson arrival process with mean λ and leave the network exactly after the download is completed. The first 110 peers finishing their download are logged, whereas the results of the first ten are ignored to neglect start-up behaviour. Table 1 shows the mean download time, standard deviation and average number of peers in the network. Also in this simulation the performance with the revenue maximisation algorithm is superior to the BitTorrent algorithm. Mean download times and standard deviations are smaller resulting in a smaller average number of peers

$1/\lambda$ [s/Arrivals]	BitTorrent			Revenue Maximisation		
	t_{mean}[s]	σ[s]	N_P^{ave}	t_{mean}[s]	σ[s]	N_P^{ave}
50	1626	1047	33	1301	360	24
100	1407	674	15	1199	251	12

Table 1: Mean download time, standard deviation and average number of peers for different Poisson arrival rates λ

in the system.
Figure 2 depicts the cumulative distribution function (CDF) of the download times for $\lambda = 100$ for the two choking algorithms. The download times for peers using the BitTorrent algorithm vary strongly, whereas the CDF of the revenue maximisation algorithm rises more steeply suggesting better fairness.

4 Conclusion and Future Work

The presented simulation results indicate that the proposed revenue maximisation algorithm outperforms the original BitTorrent choking algorithm with respect to efficiency and fairness. These properties are of great importance. Overall and individual download time depend directly on the efficiency of the network. Further on fairness must be maintained in a p2p network to give users an incentive to contribute to the network.
Future work comprises extensive simulations for both algorithms especially with larger number of peers and different upload capacities. At the moment simulation time restricts simulations to small file sizes. We aim at improving our simulator to handle larger file sizes which are closer to the file sizes which are normally shared in BitTorrent networks.

References

[AH00] Adar, E. and Huberman, B. A.: Free riding on Gnutella. *First Monday*. September 2000.

[Co03] Cohen, B.: Incentives build robustness in BitTorrent. In: *Workshop on Economics of Peer-to-Peer Systems*. Berkeley, CA. June 2003.

[IUKB+04] Izal, M., Urvoy-Keller, G., Biersack, E. W., Felber, P., Hamra, A. A., and Garcés-Erice, L.: Dissecting BitTorrent: Five months in a torrent's lifetime. In: *Passive and Active Measurements*. pp. 1–11. April 2004.

[KMT98] Kelly, F., Maulloo, A., and Tan, D.: Rate control in communication networks: Shadow prices, proportional fairness and stability. In: *Journal of the Operational Research Society*. volume 49. pp. 237–252. March 1998.

[Lu03] Luenberger, D.: *Linear and Nonlinear Programming*. Kluwer Academic Publishers. Second. 2003.