OPTIMIZATION QUESTIONS IN BITTORRENT COMMUNITIES

Elvira Antal

Department of Natural Sciences and Engineering, Faculty of Mechanical Engineering and Automation, Kecskemét College, Hungary

E-mail: antal.elvira@gamf.kefo.hu

Abstract

A BitTorrent community is a peer-to-peer (P2P) computer network. All users have restricted upload and download bandwidths, and they have different seed libraries, while various objectives could emerge in the network. The behavior of the peers is determined by the BitTorrent protocol which was designed originally for file exchange. However, other applications are also possible as BitTorrent Assisted Streaming Systems or BitTorrent Sync, for example.

This paper summarizes the main concepts of studying BitTorrent communities. After a brief introduction to BitTorrent networks and their applications, we will talk about the possibility of modeling P2P networks by simulators. On the other hand, it is possible to determine the "ideal" behavior of computer networks by solving optimization problems of the underlying flow networks.

Keywords:
optimization, BitTorrent, bandwidth allocation problem

1. Introduction

The main idea of the BitTorrent protocol [7] is to increase efficiency of file-sharing on the web by distributing the load of the original uploader among all the downloaders. Furthermore, distributed networks are known for excellent robustness compared to traditional centralized hierarchies.

Since Bram Cohen released the first version in 2001, various file-sharing networks were based on this protocol, and BitTorrent became one of the most popular services of the Internet. In November 2004, BitTorrent was generating 35% of total internet traffic, and in 2013, it was responsible for the usage of 3.35% worldwide bandwidth, what is more than half of the total bandwidth dedicated to file sharing [19]. Computer scientists' interest for analyzing this networks have been emerged together with the popularity of BitTorrent.

In consideration the wide scope of the TEAM conference, the present paper attempts to give only a brief introduction into the main concepts of studying such P2P networks. This approach hopefully helps to find connections to other conference participants' present and future works.

2. The BitTorrent protocol

The BitTorrent Protocol Specification [7] was released in 2008, and there were only three minor clarifications in the text until now. The reason is that the specification is not very strict, it contains a lot of recommendations for implementation details, as adaptable package sizes, and so on. Therefore the specification acts only as a guideline for the programmers of various BitTorrent clients, however, it tends to include de-facto rules too.

Figure 1 summarizes the process of file distribution with BitTorrent.

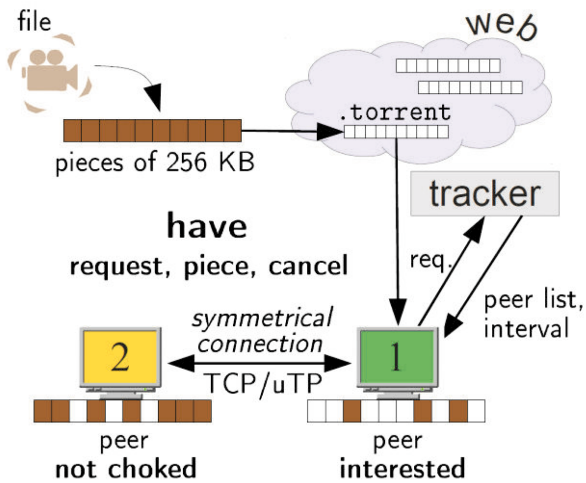

Figure 1. File distribution with BitTorrent

If someone would like to share a file, for example, his/her favorite home video, then he or she needs to create a metainfo file with `.torrent` extension. His/her BitTorrent client automatically splits the original file into small pieces, in most implementations the piece length is set to 256 KB. As the metainfo file does not contain any sensitive information, the original uploader can make it available on the web. The `.torrent` file contains the URL of the tracker and some technical information about the pieces. The tracker is a central computer which coordinates the users, but it does not store any valuable content. For private file sharing, however, it is possible, that the original uploader is the tracker at the same time.

Suppose that User 1 wants to download the mentioned video, and it sends a request to the tracker. Its ID will be included in the peer list of the

swarm (the users who share the same content), and the response contains a compact representation of the peer list. Then User 1 establish symmetrical connections with other peers over TCP or uTP in order to exchange pieces of the file. They communicate with short messages. If one peer is interested in a given piece, and another is not choked, then data transfer takes place. When a peer get a non-corrupt piece it have to announce with a `have` message to all of his peers. This simple strategy lead to less than 0.1% bandwidth overhead [6].
Every peer asks for the pieces in random order, which increases the robustness of the system.
A small example for a swarm of three users can be seen in Figure 2.

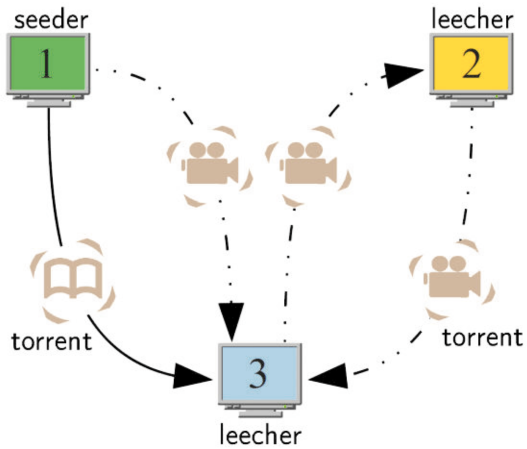

Figure 2. Swarm of three users

A peer who downloads and uploads in the same time is called a leecher. Seeder is the name of a peer, who completed the downloading and still uploads. Torrent refers to a shared file. The efficiency of BitTorrent shines up in big data distribution, however, there are no restrictions for the minimal size of the torrents.
It is polite to act as a seeder for a while [6]. Some communities require registration, and actuate accounting mechanisms in order to preserve free riding. This leads to lower resource contention than of the case of open BitTorrent sites [5].
Peers usually communicate with a small part of their swarms simultaneously. BitTorrent utilize a variant of tit-for-tat for choosing four peers to cooperate with in the same time, and they keep a free slot for optimistic unchoking (optimistic disconnection) [6, 7]. It means that the peers response on their four quickest connections, and they choose also one random peer for uploading to in the next ten seconds. Experiments of Lehman et al. shows, that optimistic disconnection is beneficial to swarm performance, as it helps to reach lower average download times. It increases the usage of total upload capacity, while results in more stable topologies, and therefore helps to reduce the negative effects of flash crowds [13].

To imagine the life circle of torrents, let us refer to the work of Izal et al. [12]. They studied a popular torrent, the 1.77 GB Linux Redhat 9 distribution in 2004, by analyzing the tracker logs for a five-month period. They found, that more than the quarter of all participants (51,000 of 180,000) connected to the swarm in the first five days (this phenomenon is called flash crowd). Average seeders remained connected for six and a half hours after finishing downloading. The seeders contributed about two times the uploading activity of the leechers. Their presence were the most dominant in the first five day (proportion of seeders and peers was over 40%), which can be explained by two factors: first, the seeders behaved altruistic; and second, the system prefers elder peers due to the tit-for-tat mechanism, therefore supports creating new seeds instead of threatening the security of downloading by dividing the resources evenly between users. Izal et al. diagnosed, that 81% of the download sessions were never finished, however, only 21.5% of the total data transfer belonged to these incomplete sessions.

3. Who is using BitTorrent?

Cuevas et al. introduce a socio-economic point of view [9] by discussing the motivation of the most intensive uploaders of the BitTorrent communities. They collected and analyzed data from Mininova and Pirate Bay, the most popular BitTorrent portals at the time of their measurements. They found, that the top 3% of the publishers, i.e. 100 users, contributed around 40% of the total published materials, while they were downloading almost nothing. Three groups of these top publishers can be distinguished: the altruistic, the profit-driven, and the malicious users. It seems that 45% of the top 100 publishers use a large number of usernames simultaneously to inject fake content to the system. They focus on video and software uploading. Cuevas et al. suggest that antipiracy agencies and malicious users can belong to this group. The former usually publishes fake versions of recent movies, while the latter is interested in sharing malware. Distressingly large amount of usernames (25%) and torrents (30%) seems to be fake. Fortunately, these contents are relatively unpopular, but still produce 25% of total downloads. Fake publishers seed for raised time and they run parallel seeds for up to 15 threads instead of the average 1. On the other hand, 48% of the non-fake top publishers seems to be also profit-driven, as they promote their web sites or private BitTorrent trackers in the uploaded contents. The remaining part of the top publishers is probably altruistic. They typically enclose very extensive descriptions of their torrents and ask for help of other users by seeding their contents.
On the other hand, the study of Cox et al. [8] focuses on the person behind the computer. They

analyzed the dataset of the Helsinki Institute for Information Technology, originated from an online survey during 2007 with 6103 answers, to find out the socio-economic background and attitudes of users of file sharing services toward legal and illegal file sharing. Cox et al. found that education and income have an inverted U shaped relationship with the probability of taking part in file sharing. Four groups can be distinguished: non-participants, leechers, seeders, and first-seeds (the first uploaders of a special content). The average age of the respondents were 33.34, 27.31, 25.20, and 23.97 years in the different groups, respectively. The social influence seems to be the main factor for involvement in file-sharing, as the participation of family members, friends, and colleges was usually related to bigger chances of participation of the respondents. Other main motivation was the prospect of financial savings. Finally, significant proportion of the first-seeders expected little chance for being caught participating in illegal file sharing, while saw itself as the group of masked philanthropists acting against incorrect legal regulations.

4. BitTorrent applications: storing and streaming

To the top of classic file-sharing, there are some further applications of BitTorrent growing in popularity.
BitTorrent Sync (BTSync, BitSync or BSync) is a new rival of the popular cloud file synchronization services, such as Dropbox, Microsoft OneDrive, and Google Drive. The BitTorrent Inc. released the private alpha in April 2013, and so far the only related academic publication is the paper of Farina et al. [10] which discusses the digital forensic possibilities in connection to BitSync. The main key and the main question of this application is privacy: can BitSync guarantee more security in comparison to the centralized cloud file synchronization services?
Media streaming is another interesting challenge for BitTorrent, and the two main field, as live streaming and video-on-demand, requires different approaches. Instead of the earlier concepts as IP level multicast and other tree-based systems at the application layer, BitTorrent offers a mesh-based architecture which is extremely robust against peer churn [14, 16]. High quality video-on-demand is more challenging than live streaming, as in the former case all the peers need different video chunks in the same time because of the asynchronous operation. Wu et al. [20] reveal that it is impossible to achieve maximum throughput of the BitTorrent network (optimal performance of the system) and maximal fairness (individual optimum) in the same time.

5. Optimal behavior, fairness conceptions

To determine optimal behavior of BitTorrent, it is essential to find an appropriate mathematical model of the system. Figure 3 contains a straightforward graph representation of the illustrative example of Figure 2.

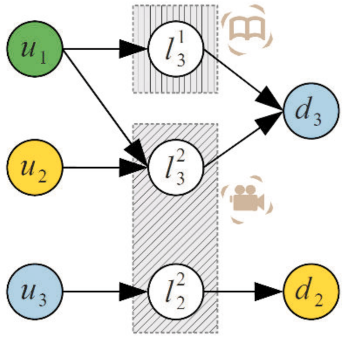

Figure 3. Bipartite graph representation of the earlier swarm

The colors refers to identical users with same color in Figure 2. The graph is bipartite, the upload and download activity of user i are represented by the nodes u_i and d_i accordingly and l_i^t represents the leeching session of user i in torrent t. With addition of a synthetic source and sink, we can get a traditional network model (Figure 4) of the given swarm.

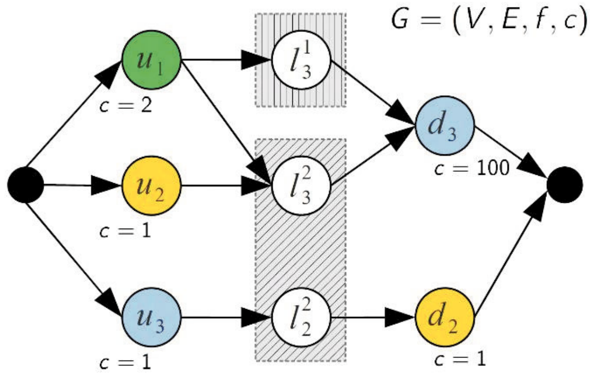

Figure 4. Network of the swarm

The edges of the graph represent the participation of the users in the leeching sessions as uploader or downloader. The flow function f assigns different amount of data exchange for the channels, while the capacity function c vindicates upload and download bandwidth constraints of the peers. This graph model was introduced by Capotă et al. [5] for examining the difference of achieved BitTorrent performance from the exact theoretical optimum.
Different objectives can be set for a P2P network as BitTorrent, as throughput maximization, max-min fairness, proportional fairness, and so on. Uchida and Kurose [18] generalized all the above mentioned popular objectives as weighted

α-proportional fairness. Consider the following optimization problem for bandwidth allocation:

$$\max \sum_{i=1}^{n} \mathrm{w}_i \, \Phi_\alpha(x_i),$$

where $\boldsymbol{x} = \{x_1, \dots, x_n\} \in \mathbb{C}$ is a feasible solution, and

$$\Phi_\alpha(x) = \begin{cases} \log x, & \text{if } \alpha = 1, \\ \dfrac{x^{1-\alpha}}{1-\alpha}, & \text{if } \alpha \geq 0,\ \alpha \neq 1. \end{cases}$$

This objective is identical to the maximum throughput for $\alpha = 0$, to the proportional fairness for $\alpha = 1$, to the potential delay minimization for $\alpha = 2$, and to the max-min fairness for $\alpha \to \infty$. Figure 5 shows the maximal throughput [5, 18] or maximal flow allocation for the earlier example.

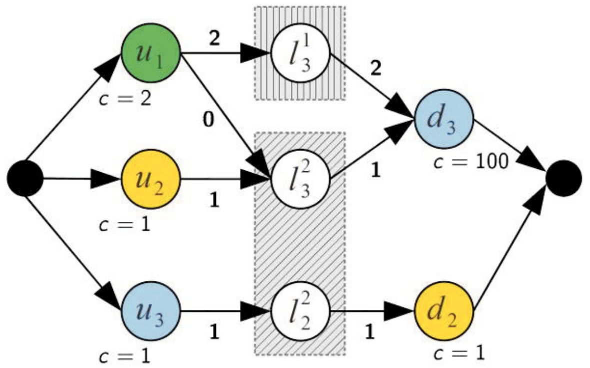

Figure 5. Maximum throughput

For comparison, the max-min fair allocation [5, 15] is presented in Figure 6.

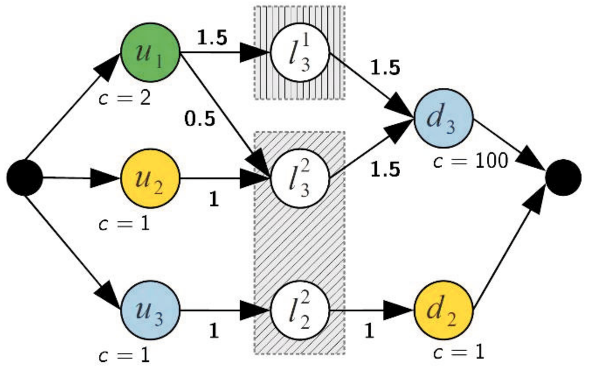

Figure 6. Max-min fair allocation

The max-min fair allocation, introduced by Bertsekas and Gallager [2], tends to equalize the download performance as far as possible, so it can be useful for media streaming, for example. In this special case, the max-min fair allocation guaranties maximal throughput also $(2 + 1 + 1 = 1.5 \cdot 2 + 1 \text{ data unit})$, but it is not a necessity. Radunović and Le Boudec [17] recommended a unified watershed-like algorithm for the computation of max-min fair allocation. However, this leads to a mixed-integer nonlinear optimization problem [4] in every iteration for the above mentioned graph model of BitTorrent swarms. Asadpour and Saberi [1] discuss the approximation possibilities for the allocation problem of indivisible goods (where every flow needs to be a non-negative integer: $f(x) \in \mathbb{N}\ \forall x \in E$).

6. Performance analysis with simulation, measurement, and modeling

Different approaches are presented in the literature for studying BitTorrent networks. First of all, it is possible to emulate the behavior of a P2P system and study the impact of various mechanisms (e.g. optimistic disconnecting in [13]) due to experiments [3, 5]. These works can build upon available P2P simulators, for example PeerSym or GPS [21]. The simulations can be used for analyzing different scenarios and proposed modifications of the BitTorrent protocol due to the comparison of different performance indicators as total upload bandwidth of the system and Jain's fairness rate [20]. Measuring the same indicators in real BitTorrent networks [11] is more challenging, but it is the main key to get acquainted with the behavior of our working BitTorrent communities. On the other side, simulated or measured data could serve system analysts building realistic topological models of BitTorrent systems [22], comparing observed performance to the theoretic optimum [5], or getting inspiration for further theoretic findings.

7. Summary

This paper attempted to catch attention of wide range of TEAM participants (researchers, teachers, and college students) by surveying the main concepts and research questions connected to BitTorrent. The author's research concentrates on the rewriting possibilities of the mathematical models of the bandwidth allocation problem, and on heuristic approaches to determine good approximations for computing the max-min fair allocation of real BitTorrent systems.

Acknowledgement

This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013).

References

[1] A. Asadpour, A. Saberi, "An approximation algorithm for max-min fair allocation of indivisible goods," *SIAM J. Comput.*, vol. 39, no. 7, pp. 2970-2989, 2010.
[2] D.P. Bertsekas, R.G. Gallager, *Data Networks*, 2nd ed., Prentice Hall, 1992.
[3] A.R. Bharambe. C. Herley, V.N. Padmanabhan "Analyzing and Improving BitTorrent Performance," *Technical Report MSR-TR-2005-03*, Microsoft Research, 2005.

[4] P. Bonami, M. Kilinç, J. Linderoth, "Algorithms and software for convex mixed integer nonlinear programs," in *The IMA Volumes in Mathematics and its Applications*, vol. 154, Springer, 2012, pp. 1-39.

[5] M. Capotă, N. Andrade, T. Vinkó, F. Santos, J. Pouwelse, D. Epema, "Inter-swarm resource allocation in BitTorrent communities," in *Proceedings of IEEE International Conference on Peer-to-Peer Computing (P2P 2011)*, 2011, pp. 300-309.

[6] B. Cohen, "Incentives build robustness in BitTorrent," in *Workshop on Economics of Peer-to-Peer systems*, vol. 6, 2003, pp. 68-72.

[7] B. Cohen, "The BitTorrent Protocol Specification," *BitTorrent.org [Online].* Available: http://bittorrent.org/beps/bep_0003.html [Accessed: 19-Aug-2014]

[8] J. Cox, A. Collins, S. Drinkwater, "Seeders, leechers and social norms: Evidence from the market for illicit digital downloading," *Information Economics and Policy*, vol. 22, pp. 299-305, 2010.

[9] R. Cuevas, M. Kryczka, A. Cuevas, S. Kaune, C. Guerrero, R. Rejaie, "Is content publishing in BitTorrent altruistic or profit-driven?," in *Proceedings of the 6th International COnference (Co-NEXT '10)*, Article No. 11, 2010.

[10] J. Farina, M. Scanlon, M-T. Kechadi, "BitTorrent Sync: First Impressions and Digital Forensic Implications," *Digital Investigation*, vol. 11, pp. S77-S86, 2014.

[11] K.P. Gummadi, R.J. Dunn, S. Saroiu, S.D. Gribble, H.M. Levy, J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in *Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP '03)*, 2003, pp. 314-329.

[12] M. Izal, G. Urvoy-Keller, E.W. Biersack, P.A. Felber, A. Al Hamra, L. Garcés-Erice, "Dissecting BitTorrent: Five Months in a Torrent's Lifetime," *LNCS*, vol. 3015, pp. 1-11, 2004.

[13] M.B. Lehmann, L.F. Müller, R.S. Antunes, M.P. Barcellos, "Disconnecting to Connect: understanding Optimistic Disconnection in BitTorrent," in *Proceedings of IEEE International Conference on Peer-to-Peer Computing (P2P 2012)*, 2012, pp. 138-148.

[14] Y. Liu, Y. Guo, C. Liang, "A survey on peer-to-peer video streaming systems," *Peer-to-Peer Netw. Appl.*, vol. 1, pp. 18-28, 2008.

[15] D. Nace, L.N. Doan, O. Klopfenstein, A. Bashllari, "Max-min fairness in multi-commodity flows," *Computers & Operations Research*, vol. 35, pp. 557-573, 2008.

[16] N. Parvez, C. Williamson, A. Mahanti, N. Carlsson, "Analysis of bittorrent-like protocols for on-demand stored media streaming," in *Proceedings of the 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS '08)*, 2008, pp. 301-312.

[17] B. Radunović, J.Y. Le Boudec, "A unified framework for max-min and min-max fairness with applications," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1073-1083, 2007.

[18] M. Uchida, J. Kurose "An information-theoretic characterization of weighted α-proportional fairness in network resource allocation," *Information Sciences*, vol. 181, pp. 4009-4023, 2011.

[19] Wikipedia, "BitTorrent," *Wikipedia, the free encyclopedia [Online].* Available: http://en.wikipedia.org/wiki/BitTorrent [Accessed: 31-Aug-2014]

[20] D. Wu, Y. Liang, J. He, X. Hei, "Balancing Performance and Fairness in P2P Live Video Systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, iss. 6, pp. 1029-1039, 2013.

[21] W. Yang, N. Abu-Ghazaleh, "GPS: a general peer-to-peer simulator and its use for modeling BitTorrent," in *Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '05)*, 2005, pp. 425-432.

[22] L. Zhong, "Topological model and analysis of the P2P BitTorrent protocol," *Int. J. System Control and Information Processing*, vol. 1, no. 1, pp. 54-70, 2012.