# Analyzing Content Life Cycle in Mobile Content Sharing Environment


Kristóf Csorba
Department of Automation and Applied Informatics
Budapest University of Technology and Economics
Budapest, Hungary
kristof.csorba@aut.bme.hu

Péter Ekler
Department of Automation and Applied Informatics
Budapest University of Technology and Economics
Budapest, Hungary
peter.ekler@aut.bme.hu

Imre Kelényi
Department of Automation and Applied Informatics
Budapest University of Technology and Economics
Budapest, Hungary
imre.kelenyi@aut.bme.hu



***Abstract*— In this paper we analyze the lifecycle of content shared via the BitTorrent network by focusing on torrents retrieved by mobile phone clients using the MobTorrent application. MobTorrent is a full-feature BitTorrent client for feature phones. It has a profiling mechanism which was used to collect anonymous usage statistics over the last two years. Based on these statistics, we analyze how mobile BitTorrent clients are being used. We also investigate the successfulness of single client sessions by performing additional measurements of peer connection success ratios and download statistics. This research can be considered as a pioneer work in the field of mobile content sharing solutions, as MobTorrent is one of the first and most popular BitTorrent client for features phones.**



***peer-to-peer, BitTorrent, mobile phones, DHT, measurements***


## I. Introduction

The main objective of this research is to evaluate the behavior of mobile devices in content sharing. One of the most efficient content sharing solutions nowadays is the peer-to-peer (P2P) based BitTorrent technology. Involving mobile phones into such large networks is challenging if we consider the limitations of mobile phones. Furthermore, we aim to support as many types of mobile devices as possible, even with different platforms and limited resources.

The ever increasing capabilities of mobile phones allow new range of applications implemented on them, including P2P clients. Efficient and scalable content distribution is a key value of P2P technologies; bringing that technology to the media player devices and mobile phones seems attractive. In that way the multimedia content would be found, accessed and played with the same device. Using a PC as an intermediary would not be needed. Nowadays mobile phones have two major categories. Smartphones can be characterized as small computers with advanced hardware and software capabilities, while feature phones are simpler mobile devices with some multimedia features. In this research we mainly focus on feature phones in order to support a wide range of devices.

Peer-to-peer solutions differ from general applications, their common characteristics are larger network intensity and distributed resource usage. Therefore we investigated BitTorrent technology and feature phones from the perspective of network handling, processing power and file handling. Following we describe our experiences and measurement results.

The result of this research can also be considered as a proof of concept that mobile phones even with limited resources are able to participate in large peer-to-peer networks like BiTorrent.

The rest of the paper is organized as follows. In Section II, we summarize the research related to BitTorrent and its applicability on feature phones. Section III discusses mobile BitTorrent clients in general. Our new results, including the analysis of the gathered data and its discussion, is presented in Section IV. Finally, Section V concludes our work and outline some ideas for future work.

## II. Related Work

One of the most popular P2P protocol is the BitTorrent. Despite its popularity, the actual behavior of these systems over prolonged periods of time is still poorly understood. Pouwelse et al. [1] presented a detailed measurement study over a period of eight months of BitTorrent. They presented measurement results of the popularity and the availability of BitTorrent, of its download performance, of the content lifetime, and of the structure of the community responsible for verifying uploaded content. The results were that the system is quite popular, but the number of active users in the system is strongly influenced by the availability of the central components. They also found that 90% of the peers experienced speeds were below 65 kB/sec. From the lifetime point of view, they showed that only 9,219 out of

53,883 peers (17 %) have an uptime longer than one hour after they have finished downloading. For 10 hours this number has decreased to only 1,649 peers (3.1 %), and for 100 hours to a mere 183 peers (0.34 %).

While it is well-known that BitTorrent is vulnerable to selfish behavior, Locher et al. [2] demonstrated that even entire files can be downloaded without reciprocating at all in BitTorrent. To this end, they presented BitThief, a free-riding client that never contributes any real data. They showed that simple tricks suffice in order to achieve high download rates, even in the absence of seeders (peers who only share the content). They also illustrated how peers in a swarm react to various sophisticated attacks.

Wang et al. [3] investigate issues related to the development of wireless P2P games with J2ME. Their work is mainly focused on the Bluetooth communication protocol and APIs but they briefly touch some other issues as well.

JXME [4], the JXTA [5] Java Micro Edition, provides a JXTA compatible platform on resource constrained devices. JXTA is a technology to create peer-to-peer applications based on Java. JXME has been an influential platform that has been used by a number of other researchers. Nützel and Kubek [6] discuss the development of a mobile extension to the HotPotato music distribution system using JXME. Bisignano et al. [7] have analyzed the use of JXME on handheld devices in the MANET context. Andersen and Torabi [8] propose a framework that is able to optimally choose an implementation matching the needs of an application. They demonstrate their system with a simple chat application running on JXME.

In the last two decades enormous efforts have been devoted to developing wireless communication technologies. Once affordable only to specific niche markets, these wireless communications are rapidly becoming everyones mainstream source of connectivity. In [9], the authors are introducing different concepts of cognitive and cooperative networks. Peer-to-peer networks also belong to this category. The authors are introducing a P2P based information retrieval system.

The first experimental steps towards bringing peer-to-peer technology to mobile phones have already been taken with the implementations of popular content sharing protocols, Gnutella and BitTorrent, for mobile phones [10]. The applications, Symella [11], SymTorrent [12] and MobTorrent [13], are available in source code. However, these solutions were implemented on the Symbian platform which limits their use to a subset of high end mobile devices and they do not consider simple mobile phones.

A key difference between previous research and our work is that our goal is to involve feature phones in the BitTorrent network, and analyze their behavior in real operational environment.

## III. MOBILE BITTORRENT CLIENTS

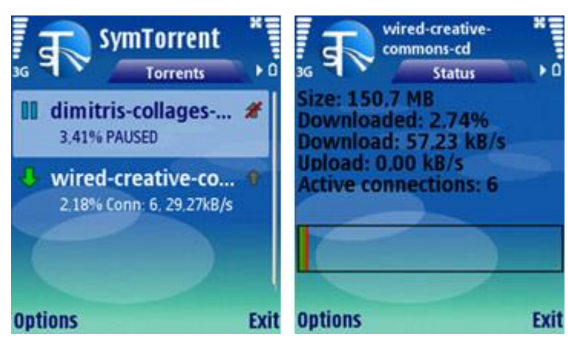
Nowadays mobile phones are also used as portable music players. The consumption of video and other multimedia content besides music on mobile devices is also increasing. The increasing capabilities of high-end mobile phones and also mainstream devices allow new range of applications implemented on them. An important set of applications are peer-to-peer (P2P) applications.

Efficient and scalable distribution of multimedia content is a key value of P2P technologies; bringing that technology to the player devices seems attractive. In that way the multimedia content would be found, accessed and played with the same device. Using PC as an intermediary would not be needed.

The essence of BitTorrent [14] is that content (one or more files) is downloaded in multiple pieces. The different pieces can be downloaded from different peers (if possible) in parallel which makes the aggregate download speed much higher than downloading from a single bandwidth limited peer only. Downloadable content is described by a torrent file which contains the address of the tracker and the hash values of the content pieces. The tracker maintains a list of peers that are working on the download (or share) of the same content. MobTorrent can open and interpret the torrent file. After that it connects and registers to the tracker, which sends back addresses of several peers that are able to serve the content. Then it sends piece request messages to these peers which respond by starting to upload the content to us.

Our first mobile BitTorrent client was SymTorrent (Figure 1.), that allows download, upload and it also contains a Tracker and a TorrentMaker. This way the users are able to share their own content on mobile phones. These applications are available in source code at the Budapest University of Technology and Economics. However, this application was implemented on the Symbian platform for S60 devices, which limits their use to a subset of high-end mobile devices.

Figure 1. SymTorrent




After this implementation we started to work on bringing the popular BitTorrent P2P solution also to low-end mobile phones and that way the system can be used by a wide area of people. The target platform was the Nokia Series 40 platform (S40). We created our own Java Micro Edition (Java ME) based BitTorrent application, called MobTorrent

(Figure 2.). MobTorrent was implemented by porting the SymTorrent Symbian application to Java ME platform. Currently MobTorrent supports only content downloading. Implementation of upload functionality is on-going.

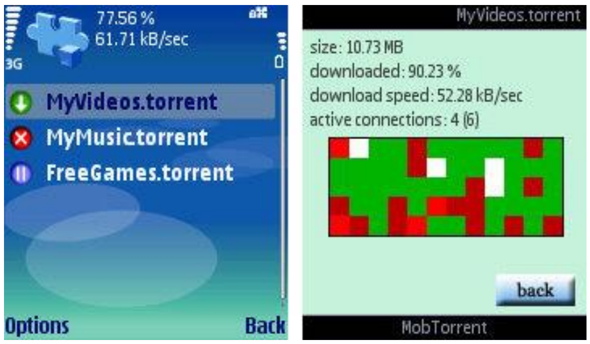



Figure 2. MobTorrent

In [15] we described MobTorrent in detail. The paper investigates deeply the abilities of the mainstream phones, explains the problems while bringing BitTorrent technology to low-end phones and also introduces the measurement results of the application (download speed and memory usage).

## IV. Analyzing Content Life Cycle and User Behaviour

MobTorrent sends anonymous usage statistics to a profiling server, if the function is enabled by the user. We managed to collect a dataset suitable for statistical investigations. It should be noted, that these results are based on data collected specifically for torrents downloaded by mobile clients: torrents containing data of multiple 100 MB are less likely to appear in the mobile environment, as in the BitTorrent traffic of desktop computers.

The raw dataset we use for our investigations consists of two data tables: one for individual torrents and one for BitTorrent client sessions. In the following, we will analyze these two sets of data from various aspects to get a better understanding on the torrents downloaded by mobile BitTorrent clients.

### A. *Evaluation of torrent data*

Torrent data contain information about the retrieval of a single torrent. The first few records of the data table are shown in Table 1. The meanings of the columns are the following:

- *ID*: It is only a unique identifier of the record without further meaning.
- *InfoHash*: identifies the torrent, as the hash value of the torrent file.
- *Success* (also *S* in the following): *Success* describes the accessibility of the peers: after retrieving the list of possible peers from the tracker or the DHT, *success* is the number of peers the client could successfully connect to.
- *Failed* (also *F* in the following): Similarly to *success*, *failed* is the number of peers the client could not connect to, after it retrieved the list of peers from the DHT.

In the whole data set, a total number of records (torrent download cases) is 25038.

By observing S and F, one of the most important measures from the user's point of view is the rate of peers the client could connect to (Success Rate, SR): SR = S/(S+F).

The histogram of the Success Rate *hist(S/(S+F))* is shown in Fig. 3. It has to be noted, that there were many cases with S=0, and thus, zero Success Rate. These 3880 cases have been removed to avoid its domination on the histogram in the form of a huge peak at 0.

As the figure shows, the most common success rates are between 0.2 and 0.4. The peaks at 1, 0.5, 0.33, 0.25 and 0.2 should be noted: as a significant ratio of the torrents had a low number of possible peers, these success rate values are more common than the other ones, as these can be achieved with one successfully connected peer (*S=1*) and a few unconnectable ones (low *F*).

TABLE I. First few records of registered torrent data

| ID | InfoHash | success | failed |
|---|---|---|---|
| 1 | F1A40B5254F029ACDCBF6BD8956A6194EA56189E | 101 | 215 |
| 2 | EC958201DD8878AD96F8348E6455303F75428559 | 30 | 77 |
| 3 | 56F613D4218D46EF36C56189E60D1479EB7A9C66 | 14 | 36 |
| 4 | 336C4D5F4A87E14F7BEDA454E626691CB3A0E26F | 262 | 389 |
| 5 | 472C912266612DAC218E2B787B54234DB34FED2B | 29 | 98 |

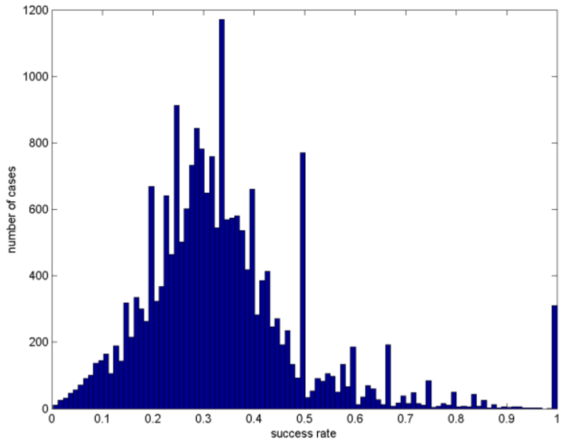



Figure 3. Histogram os success rate, which is the rate of successful peer connections among all peers retrieved for the torrent.

The previous observation about the torrents with low peer numbers leads to the next question: how many peers are there for an average torrent retrieved by mobile clients? The histogram of the total number of peers *hist(S+F)* is presented in Fig. 4. As the number of peers varies on a very large scale, the horizontal axis is chosen to have logarithmic scale, so that the details around lower values remain visible. The maximal value is 551 cases with *S+F=7*. The histogram shows that the most typical peer numbers are around the intervall 2—20.

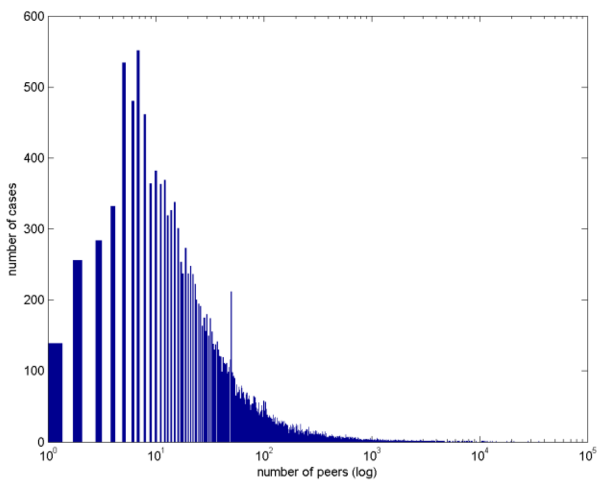



Figure 4. Histogram of the number of peers retrieved for a torrent. The maximal value is at 551 for S+F=7.

The relationship between the successful and failed peer connections is visualized in the two-dimensional space of *S* and *F*. This is shown in Fig. 5, and to better highlight the details, Fig. 6 presents the same data in logarithmic scales for both S and F.

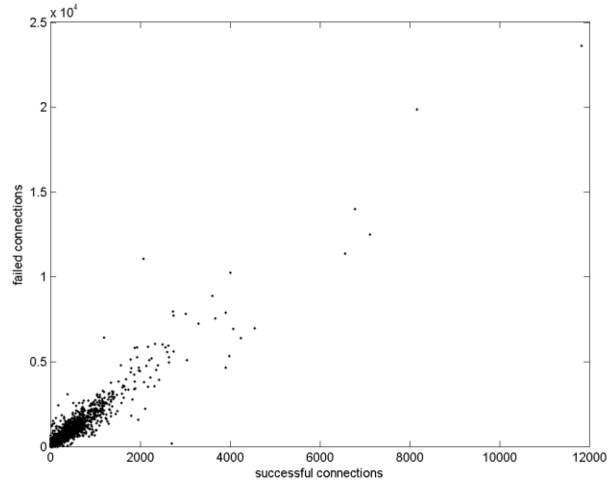



Figure 5. All cases (torrent downloads) plotted in the space of successful (horizontal axis) and failed connections (vertical axis).

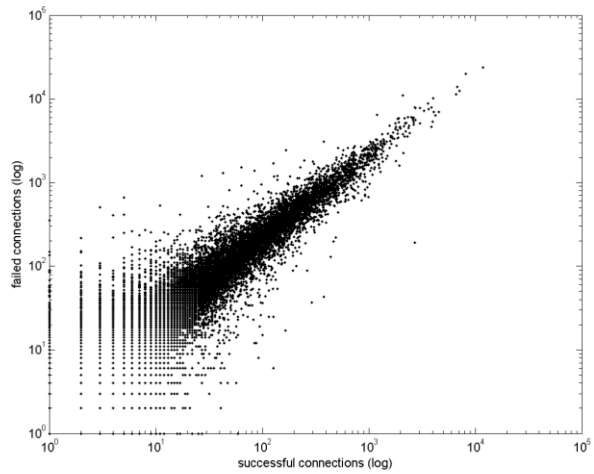



Figure 6. All cases plotted in the space of successful and failed connections. Both axes are shown with logarithmic scale to improve visibility of details in small value regions.

Both figures confirm the rough linear relationship between the successful and failed connection numbers for higher peer numbers, and a bias towards more failed connections in the area of lower peer numbers.

### B. *Evaluation of client session data*

Client session data contain information about the downloads between the start end close of MobTorrent. Each record consists of the following values:

- *ID*: only the record identifier, does not have further meaning
- *created*: the date and time of the entry

- *completed* (*C* in the following): the number of torrents completely downloaded
- *unfinished* (*U* in the following): the number of torrents which have started downloading, but the download was not completed before closing the program.

The first few records of the data set are presented in Table 2 for example. Altogether, 22088 records were collected.

TABLE II. FIRST FEW RECORDS OF REGISTERED CLIENT SESSION DATA

| ID | created | completed | unfinished |
|---|---|---|---|
| 1 | 2009.03.30 11:52 | 29 | 99 |
| 2 | 2009.03.30 13:57 | 5 | 58 |
| 3 | 2009.03.30 15:37 | 0 | 1 |
| 4 | 2009.03.30 17:34 | 2 | 3 |
| 5 | 2009.03.30 19:38 | 0 | 2 |

Considering the situation, where one wants to download a specific file to the mobile device and then one closes MobTorrent, the first value we investigated was the number of sessions with one single completed download, and the number of sessions with one single unfinished downloads:

- Total number of registered sessions: 22088
- Number of sessions with only one completed download: 212
- Number of sessions with only one unfinished download: 7677

The latter value seems to be very high, but we have to take into account, that an unfinished download may be resumed in a later session, so these 7677 sessions were not necessarily useless ones.

To compare the number of complete and unfinished downloads, Fig. 7 presents the sessions in the space of *C* and *U*. Although there ware sessions with much higher *C* and/or *U* values, they have been omitted as outlier cases to improve the figure quality. (There number was too low for statistical conclusions anyway.)

The figure suggests that usually, there is a higher number of unfinished downloads, than completed ones. One possible interpretation is that the users start multiple downloads for the same content, and as soon as the first one is completed, the remaining ones are cancelled.

The completion rate defined as CR=C/(C+R) is an important measure for MobTorrent usage. There are many sessions with no completed downloads (*C=0*): 18609 sessions, which is 84.25% of the total number of registered sessions. This caused by the high number of interrupted, and later resumed sessions. Fig. 8 presents the histogram of the completion rate *hist(CR)*. To improve the visibility of important values, the mentioned *C=0* cases were omitted to remove the dominating peak at *CR=0*, and only the remaining 3479 sessions were taken into account. The peaks at values frequently achieved with low torrent numbers (0.5, 0.33, 0.25, 0.2 etc.) are clearly visible in this histogram as well.

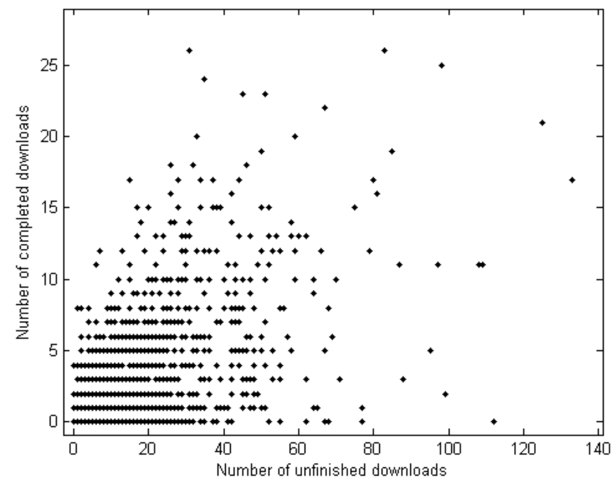



Figure 7. All registered client sessions in the space of the number of unfinished (horizontal axis) and completed downloads (vertical axis).

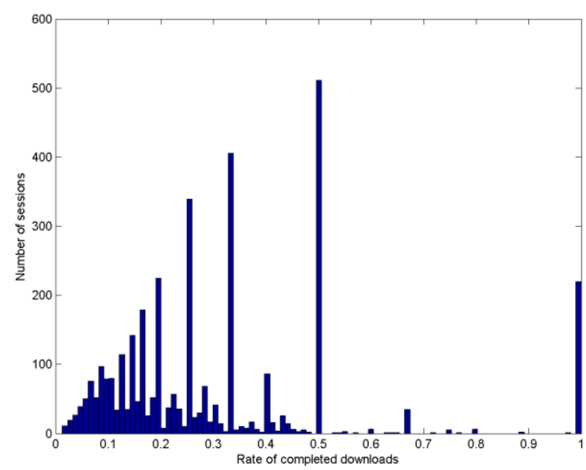



Figure 8. Histogram of the rate of completed downloads. Peaks are introduced by values achievable with low torrent numbers.

As the data set contains a timestamp for every record, client session related information can also be observed as a function of time: number of sessions, C, U and completion rate are presented in the following for 1 month intervals. The values are cumulated for one month intervals to suppress noise.

Fig. 9 presents the number of sessions per month, for every month since the start of MobTorrent in $30^{th}$ of march, 2009. The snapshot of the data set used for these evaluations was saved on the $28^{th}$ of march, 2011. The monthly session numbers have a near linear increasing tendency and doubles in one year. This clearly shows that MobTorrent and using BitTorrent clients on mobile devices becoming more and more popular.

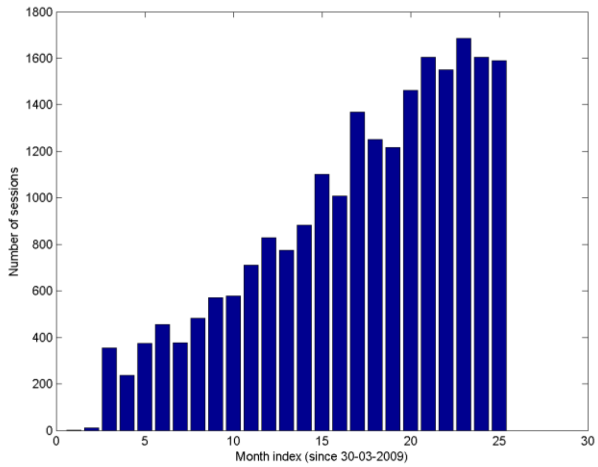



Figure 9. Monthly number of sessions since the start of MobTorrent.

Fig. 10 presents the cumulated number of completed and unfinished downloads for the same 1 month intervals, and the completion ratio is illustrated in Fig. 11.

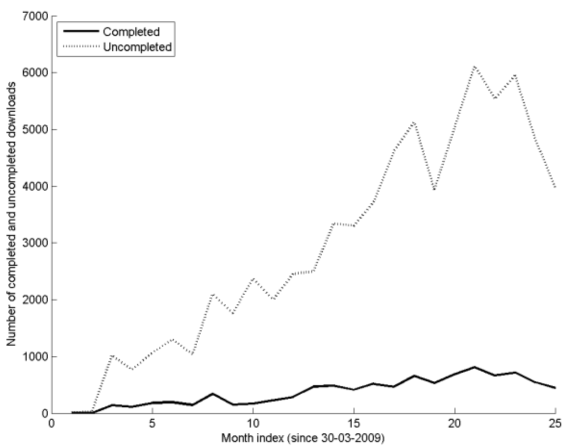



Figure 10. Monthly number of completed and unfinished downloads.

As time goes by, the number of completed and unfinished downloads increases together with the number of sessions, but the completion rate does not seem to have a significant tendency: it seems to remain between 0.08 and 0.14 most of the time. (The low ratio is significantly influenced by the interrupted, and later resumed downloads, which appear in the form of many unfinished and one completed download in this data set.) An interesting pattern is a drop in completion rate after 6-7 month intervals, which was especially significant in the early months of MobTorrent's life. The reason behind this pattern is still subject of further investigation.

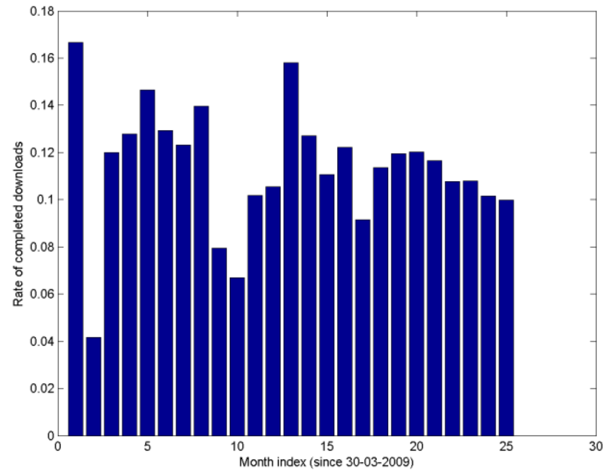



Figure 11. Ratio of completed downloads for the individual months since the start of MobTorrent.

## V. CONCLUSION AND FUTURE WORK

In this paper, we have presented the evaluation of the statistics collected by MobTorrent, a BitTorrent client for feature phones. The data collected over the period of more than two years reflects how mobile BitTorrent is becoming more and more popular and what are the general usage patterns in the mobile environment. Our future plans for research include extending the profiler with additional measurements, such as session time and transfers speeds, and analyzing the actual torrents based on their infohash.


## ACKNOWLEDGMENT

This work is connected to the scientific program of the "Development of quality-oriented and cooperative R+D+I strategy and functional model at BUTE" project. This project is supported by the New Hungary Development Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).


## REFERENCES


[1] I. Kelényi, B. Molnár, Symella 1.40. 2007.

[2] J. Pouwelse, P. Garbacki, D. Epema, H. Sips, "The BitTorrent p2p file-sharing system: Measurements and analysis", IPTPS'05. *4th Int. Workshop on Peer-To-Peer Systems 2006*, Ithaca, New York, USA

[3] T. Locher, P. Moor, S. Schmid, R. Wattenhofer, "Free Riding in BitTorrent is Cheap", *In HotNets*, 2006

[4] A. Inge Wang, M. Sars Norum, C. Wolf Lund, "Issues related to Development of Wireless Peer-to-Peer Games in J2ME", *Telecommunications 2006*. AICT-ICIW '06. *Int. Conf. on Internet and Web Applications and Services/Advanced Int. Conf. on*, vol., no., 19-25 Feb. 2006, pp. 115-115

[5] JXTA Java Micro Edition overview, Dec. 18, 2007. [Online]. Available: https://jxta-jxme.dev.java.net/

[6] JXTA description, Dec. 27, 2007. [Online]. Available: https://jxta.dev.java.net/

[7] J. Nutzel, M. Kubek, "A Mobile Peer-to-Peer Application for Distributed Recommendation and Re-Sale of Music", AXMEDIS '06. *Automated Production of Cross Media Content for Multi-Channel*

*Distribution, 2006. Second Int. Conf. on*, vol., no., Dec. 2006, pp.93-98

[8] M. Bisignano, G. Di Modica, O. Tomarchio, "A JXTA compliant framework for mobile handheld devices in ad-hoc networks", *Computers and Communications, 2005*. ISCC 2005. *Proceedings. 10th IEEE Symposium on*, vol., no., June 2005, pp. 582-587, 27-30

[9] A. M. Andersen, T. Torabi, "A holistic framework for mobile environments", *Software Engineering Conf.*, 2006. Australian, vol., no., April 2006, pp. 7 pp.-, 18-21

[10] B. Forstner, G. Csúcs, I. Kelényi, H. Charaf, "Peer-to-Peer Information Retrieval Based on Fields of Interest", *In Cognitive Wireless Networks*; F. H. P. Fitzek, M. D. Katz; ISBN: 978-1-4020-5978-0. Springer, 2007

[11] I. Kelényi, G. Csúcs, B. Forstner, H. Charaf, "Peer-to-Peer File Sharing for Mobile Devices", *In Mobile Phone Programming: Application to Wireless Networks*; F. Fitzek, F. Reichert Eds.; ISBN: 978-1-4020-5968-1. Springer, 2007

[12] B. Molnár, B. Forstner, I. Kelényi, "Symella 1.40", *Budapest University of Technology and Economics*. Apr. 11, 2007. [Online]. Available: http://amorg.aut.bme.hu/projects/symella

[13] I. Kelényi, "SymTorrent 1.50", *Budapest University of Technology and Economics*. Apr. 11, 2011. [Online]. Available: http://amorg.aut.bme.hu/projects/symtorrent

[14] P. Ekler, "MobTorrent 1.1", *Budapest University of Technology and Economics*. Apr. 11, 2011. [Online]. Available: http://amorg.aut.bme.hu/projects/mobtorrent

[15] BitTorrent specification, Apr. 11, 2011. [Online]. Available: http://wiki.theory.org/BitTorrentSpecification

[16] Péter Ekler, Jukka K. Nurminen, Attila Kiss. "Experiences of implementing BitTorrent on Java ME platform", *5th IEEE Consumer Communications & Networking Conference (CCNC 2008)*, Las Vegas, 2008.