

CloudTorrent – Energy-Efficient BitTorrent Content Sharing for Mobile Devices via Cloud Services

Imre Kelényi

Budapest University of Technology and Economics
Budapest, Hungary
imre.kelenyi@aut.bme.hu

Jukka K. Nurminen

Nokia Research Center
Helsinki, Finland
jukka.k.nurminen@nokia.com

Abstract—BitTorrent-based file-sharing is already available for mobile phones; however, its energy profile makes it difficult to use it for transferring large amount of data. This paper analyzes an alternative cloud-based solution that uses a remote server to download content via BitTorrent and transfer it to the mobile device in a transparent and energy efficient way. The system is evaluated via measurements carried out on smartphones.

Keywords—mobile, energy-efficiency, cloud-computing, peer-to-peer, content-sharing, BitTorrent

I. INTRODUCTION

When mobile devices participate in content sharing communities we face at least two challenges. First, is the amount of data that the torrent download creates uneconomical? With the increasing adoption of flat rate tariffs on cellular network the cost no longer seems an obstacle. Moreover, many mobile devices support WiFi, which allows bypassing the cellular operator completely. The second, and more serious obstacle, is the energy consumption of the mobile device. The progress on battery technology is steady but slow so spending as little energy as possible on different operations is likely to remain an important design constraint for mobile solutions.

In our earlier work, we have analyzed the energy consumption of a BitTorrent client running on Symbian S60 devices [1] and devised energy-efficient extensions to the BitTorrent protocol [2]. The focus of this work is on the relationship of BitTorrent and the increasingly popular cloud computing. In particular, we investigate the case when the BitTorrent client is run on the cloud and mobile clients access it to download torrent content. The idea of running a BitTorrent client on a different computer and control it remotely is not completely new. uTorrent has a remote control API and some persons have been experimenting the use of torrent clients on Amazon EC2 [3]. These experiments show that the concept works. In this research, we have two main contributions: study how a torrent client in the cloud would work together with a mobile phone and perform systematic measurements on the behavior of such a concept especially from the power consumption point of view.

II. CLOUDTORRENT

The architecture of CloudTorrent consists of two main parts: a phone application communicating with the cloud and a server hosting the remote BitTorrent client. The phone

application is an extension of the SymTorrent open source BitTorrent client for Symbian S60 devices. It is able to control the remote BitTorrent server, indicate download progress, and transfer the downloaded files from the server to the mobile device. All communication with the server is carried out via a single HTTP connection.

On the server side, we use uTorrent, which is a popular free PC BitTorrent client with most of its functions available via an HTTP-based API. Since the uTorrent API does not support file downloading, we also run a separate web server that is used to transfer the downloaded files to the mobile devices.

Currently the CloudTorrent content download happens in two steps. First the server side uses the BitTorrent protocol to download the content to the CloudTorrent server. Once the torrent download is completed, an HTTP download transfers the content to the phone. From the user experience point of view, CloudTorrent operates like a standard BitTorrent client. After initial configuration of the remote server access, the phone application works in a transparent way hiding the two-phase transfer from the user.

III. MEASUREMENTS AND RESULTS

In order to evaluate the proposed cloud-based solution, we compared its performance with SymTorrent. We performed energy and transfer speed measurements with Nokia N82 phones connected to the Internet via 3G. The server hosting the BitTorrent client and the web server was an Amazon EC2 instance with at least 10 Mb/s uplink capacity. In the test case, a 25 MB size torrent was downloaded to the phone using the two different methods.

TABLE I. MEASUREMENT RESULTS OF DOWNLOADING A 25 MB TORRENT

Method	Energy (J)	Download time (s)	Average download speed (kB/s)
SymTorrent	672	465	58
CloudTorrent	248	189	189

The measurements results are depicted in Table I. The download time is the total time from the invocation of the download to the time the full content has arrived to the mobile. The average download speed, on the other hand, focuses only on the speed experienced by the mobile device: in the CloudTorrent case the HTTP file transfer from the server to

mobile and in the SymTorrent case the aggregate download speed from different peers.

CloudTorrent outperformed SymTorrent both in energy consumption and in download time. The difference in energy consumption can mainly be attributed to difference in download speeds; CloudTorrent was able to reach much higher transfer speed than SymTorrent. This is consistent with the earlier observations that the higher the bit rate the lower the energy cost per bit [2]. In the CloudTorrent case, the server isolated the mobile client from the limitations and variability of torrent download and provided a fast, dedicated connection to the mobile. SymTorrent, on the other hand, received data from several peers and suffered directly from the bandwidth limitations of the peers, Internet bottlenecks, and competition between multiple downloaders.

Moreover, since the BitTorrent client in the cloud is able to serve its peers with high upload speeds, the tit-for-tat mechanism increases the download speed of the torrent. If we compare the torrent download times only (and exclude the time transferring the file from the server to the mobile) we notice that CloudTorrent server was able to download the torrent 88% faster than SymTorrent.

Further details of the energy consumption and download speed as a function of time can be observed in Figures 1 and 2. For SymTorrent the download speed was fluctuating between 40 and 80 kBytes/s during the download time. The power consumption was almost constant with a 1.5W average.

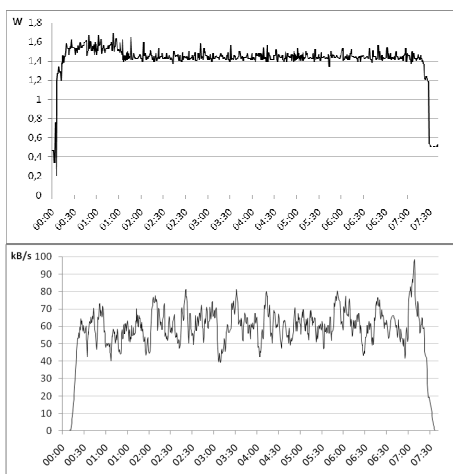


Figure 1. SymTorrent energy consumption and download speed

In the case of CloudTorrent, the transfer session started with a low power phase with some high-power spikes. During this period the server is downloading the torrent and a phone is idle waiting for the content to be ready. The spike at 30s arises because the mobile polls the status of the server via HTTP. These queries involve only a few bytes of data transfer, which is not even visible in the transfer speed graph, but the power consumption remains at a high level for around 10 seconds. Our assumption is that the long delay is due to the 3G timer settings that control the power save mode activation in the 3G network.

This indicates that handling the torrent progress indication in a smart way on the phone side can have major importance for energy-efficiency. For very large content, e.g. movies, it does not make sense to poll the progress of the torrent too frequently as each poll consumes some energy. However, in the case of smaller content, where the download times are rather short, the user may be eager to follow the progress. An adaptive mechanism based on the size of the content might be the proper solution.

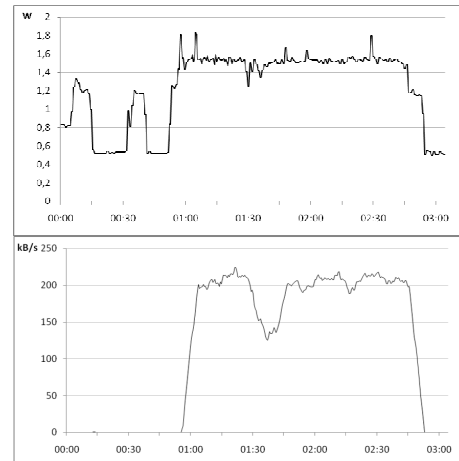


Figure 2. CloudTorrent energy consumption and download speed

In this study, we have assumed that the content is moved as a file to the mobile device for later consumption. However, other alternatives are also possible; the user could e.g. stream the content to the mobile from the CloudTorrent server.

IV. CONCLUSIONS

The results in this paper have shown that moving the torrent client functionality to the cloud makes a lot of sense for the mobile device: energy consumption is reduced, traffic is reduced and access to the content on the cloud server can be done in multiple ways such as HTTP or streaming.

At first glance, using servers to download torrent content may seem contrary to the P2P ideology of distributing tasks among the client devices. In our case, the benefits for the mobile user are promising. However, the feasibility of servers in the cloud would require further investigation both from the network architecture and business side. Another question for further investigation is comparing the cases when the torrent servers are running in the cloud, like in this study, or at devices owned by the user e.g. on home computers.

REFERENCES

- [1] J. K. Nurminen and J. Nöyränen, "Energy-Consumption in Mobile Peer-to-Peer – Quantitative Results from File Sharing", 5th IEEE Consumer Communications & Networking Conference (CCNC) 2008
- [2] I. Kelényi, J. K. Nurminen, "Bursty Content Sharing Mechanism for Energy-Limited Mobile Devices," in Proc. of 4-th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks (PM2HW2N 2009), Tenerife, Spain, 2009
- [3] How to Use Amazon EC2 for Bittorrent, available at: <http://negatendo.net/blog/2009/01/17/howto-use-amazon-ec2-for-bittorrent/>