# Incentives Build Robustness in BitTorrent by Bram Cohen

Proseminar Algorithmen für Rechnernetze

Jascha Epperlein

SS 2012

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

# Motivation

UNI FREIBURG

- BitTorrent is one of the best-known protocols for filesharing
- Simple yet effective
- One of the few filesharing protocols being used for legal distributions
- This presentation will show you how BitTorrent works and its benefits

# Introduction of BitTorrent

UNI FREIBURG

- Developed by Bram Cohen for etree
- Tracker-based distributed download system
- With BitTorrent it is possible to distribute large data to a great number of peers
- Differences between BitTorrent and p2p networks like gnutella

# Structure of the presentation

- The presentation has 4 parts:
- 1. What BitTorrent does
- 2. Technical Framework of BitTorrent
- 3. Choking in BitTorrent
- 4. What has changed since the paper was published?

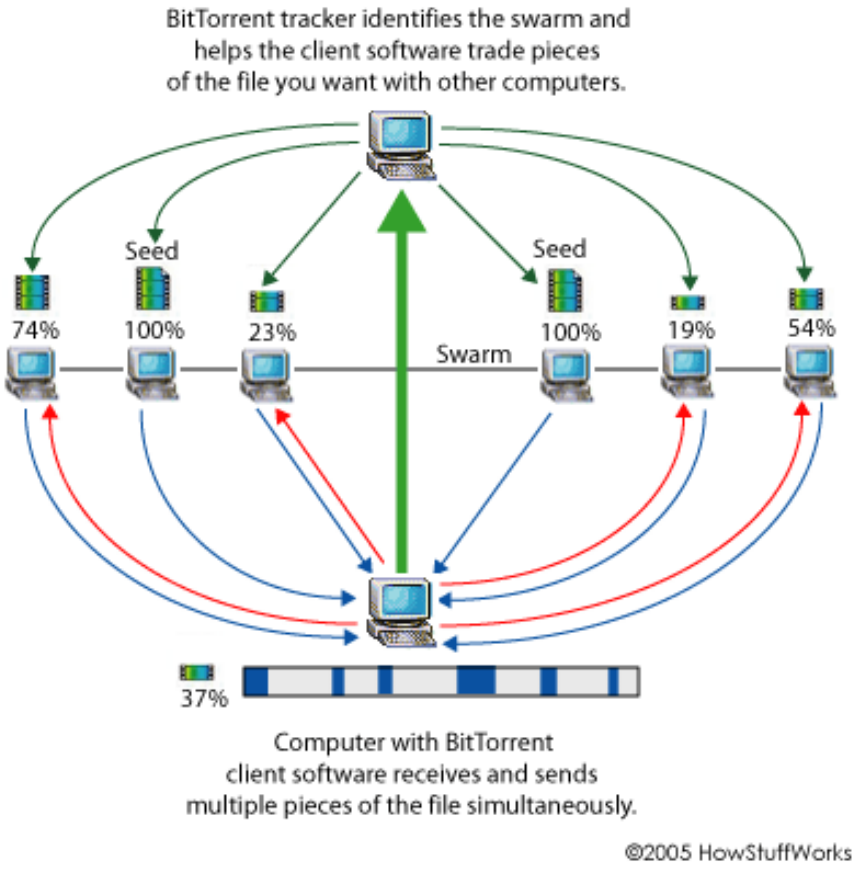
# 1. What BitTorrent does

UNI FREIBURG

- Makes it possible to distribute large data
- Upload cost is distributed to downloaders
- Has a simple interface since it first was developed
- Who uses BitTorrent → Publisher decides

# 2. Technical Framework of BitTorrent

- Publishing Content
- Peer Distribution
- Pipelining
- Piece Selection




Grafik: http://computer.howstuffworks.com/bittorrent2.htm

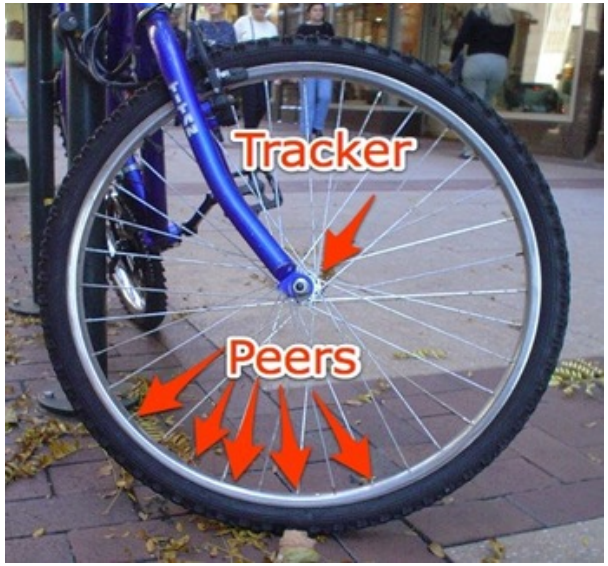
# Publishing Content

UNI FREIBURG

- Publisher decides to use BitTorrent
- Creates static .torrent file and uploads it to HTTP server
- .torrent includes all important information about the file
- Trackers help peers find each other

# Peer Distribution

- The standard Tracker returns a random list of peers
- Hash value for every chunk
- Peers announce which chunk they have




Left: Tracker identifies the swarm.
Right: Peers exchange chunks without touching the tracker.

Grafiken: http://omgeureka.blogspot.de/2011/03/what-is-bittorrent-basic-guide-to.html

# Pipelining

UNI FREIBURG

- Data is transferred over TCP
- Breaking chunks into sub-pieces
- Several requests pending at once

# Piece Selection I

- Important for good performance
- Strict priority policy
- Rarest first policy
- Reduces risk of losing particular chunks

# Piece Selection II

- At the beginning of a Download: random first piece
- Important to get a complete piece quickly
- Endgame mode

# 3. Choking

- The theoretical idea behind choking
- BitTorrent‘s Choking Algorithm
- Optimistic Unchoking
- Anti-snubbing
- Upload only

# The Theoretical Idea behind Choking

- Each peer is responsible for maximizing its download rates
- Pareto efficiency
- Prisoner‘s Dilemma → „Tit for tat“

# BitTorrent‘s Choking Algorithm

UNI FREIBURG

- BitTorrent unchokes a fixed number of peers
- Decision on unchoking based on download rate → 20 second average
- Recalculating Chokes every 10 seconds

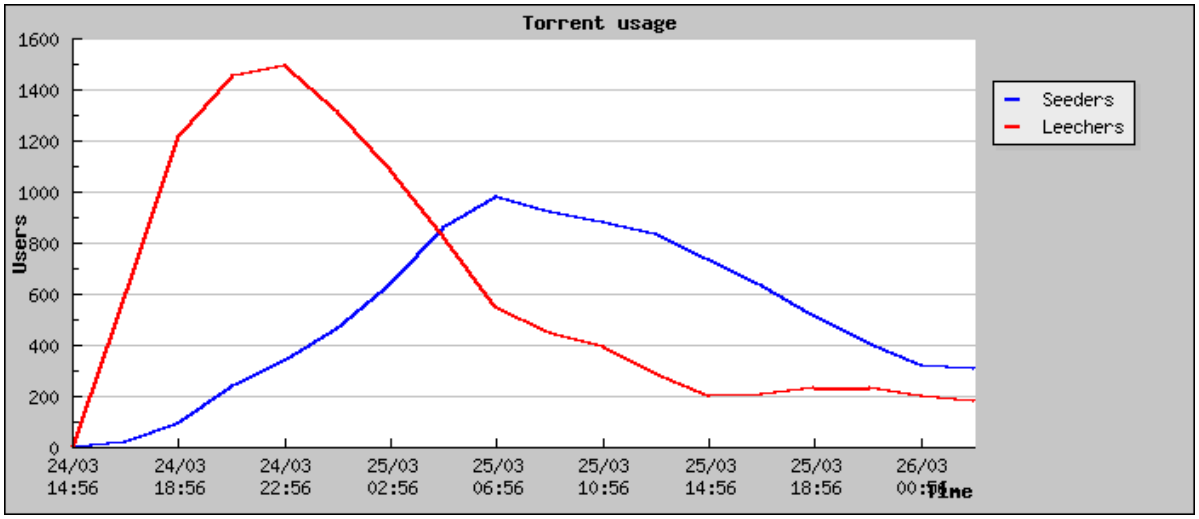
# Optimistic unchoking/Anti-snubbing

- Detecting if there are better connections available than the ones being used
- Occasionally a peer will be choked by all peers it was downloading from
- When snubbed peer stops uploading → leads to several concurrent optimistic unchokes

# Upload Only

- Which peer to upload to when download is finished?

→ Uploading to the peer to which I get the best upload rate




Grafik: Incentives Build Robustness in BitTorrent [1]

# 4. What has Changend with BT?

UNI FREIBURG

- Many different BitTorrent clients which help the user downloading/uploading files
- Large sites with vast amount of files for sharing via BitTorrent
- Trackerless BitTorrent
- BitTorrent is the protocol which brings forth the most web-traffic europewide

# End of presentation

UNI FREIBURG

Thank you very much for your attention

# References


- Cohen, Bram (2003). Incentives Build Robustness in BitTorrent. In Workshop on Economics of Peer-to-Peer systems, volume 6, pages 68-72, 2003.
- Data about BitTorrent‘s usage: http://www.slideshare.net/ipoque/ipoque-internetstudy0809
- http://www.bittorrent.com
- Axelrod, R. (1980). Effective choice in the prisoner's dilemma. Journal of Conflict Resolution, 24(1), 3-25. Sage Publications.