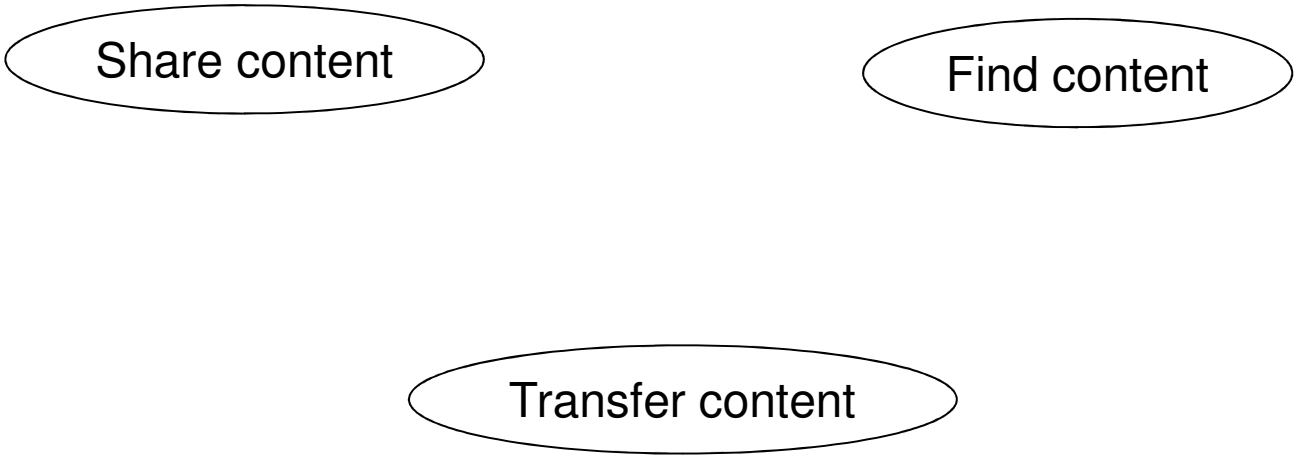
Aalto University

BitTorrent and CoolStreaming

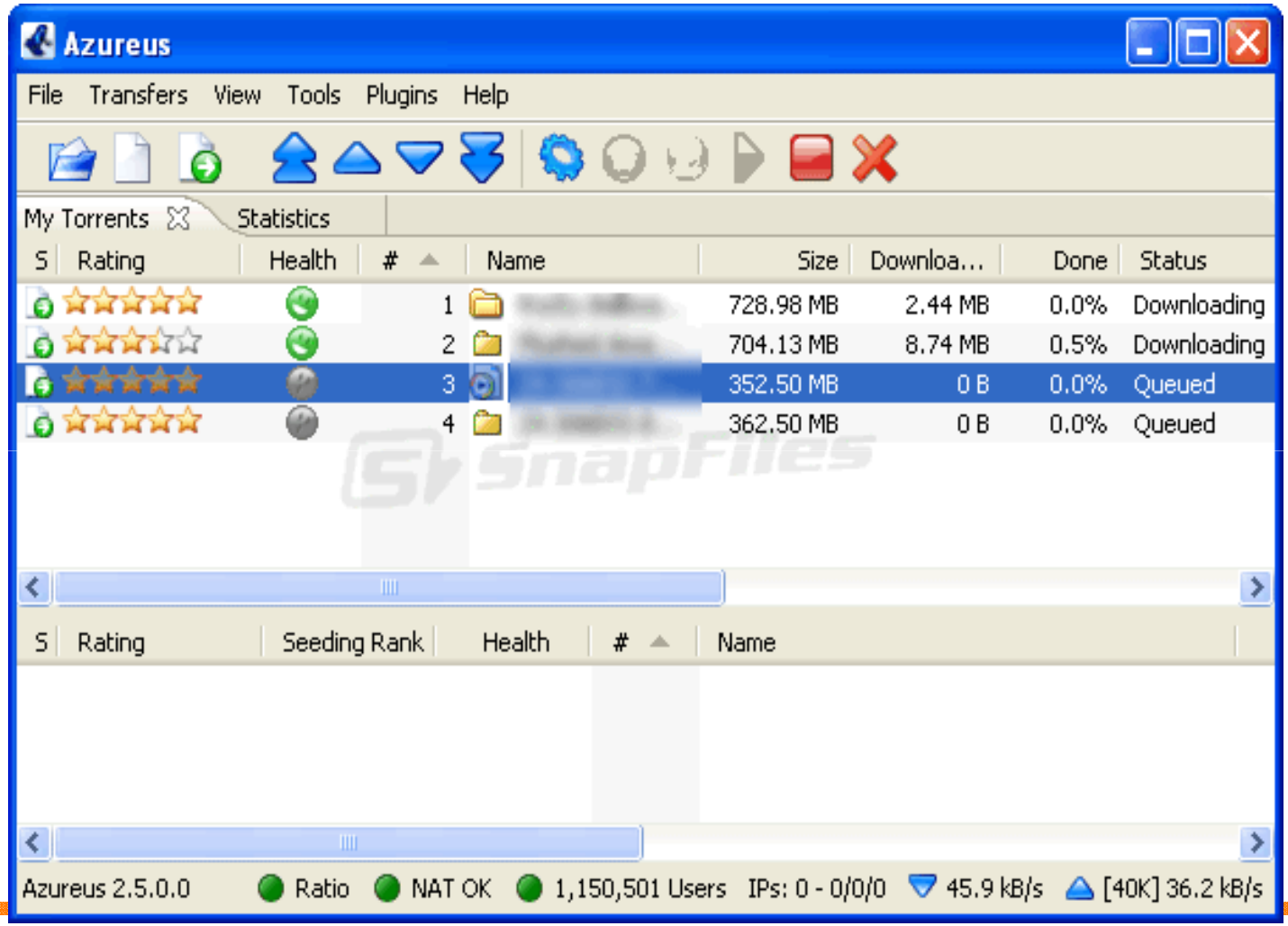
Zhonghong Ou
Post-doc resercher
Data Communications Software (DCS) Lab,
Department of Computer Science and Engineering,
Aalto University

Slides are mostly from Prof. Jukka K. Nurminen, Aalto University

Steps of content sharing



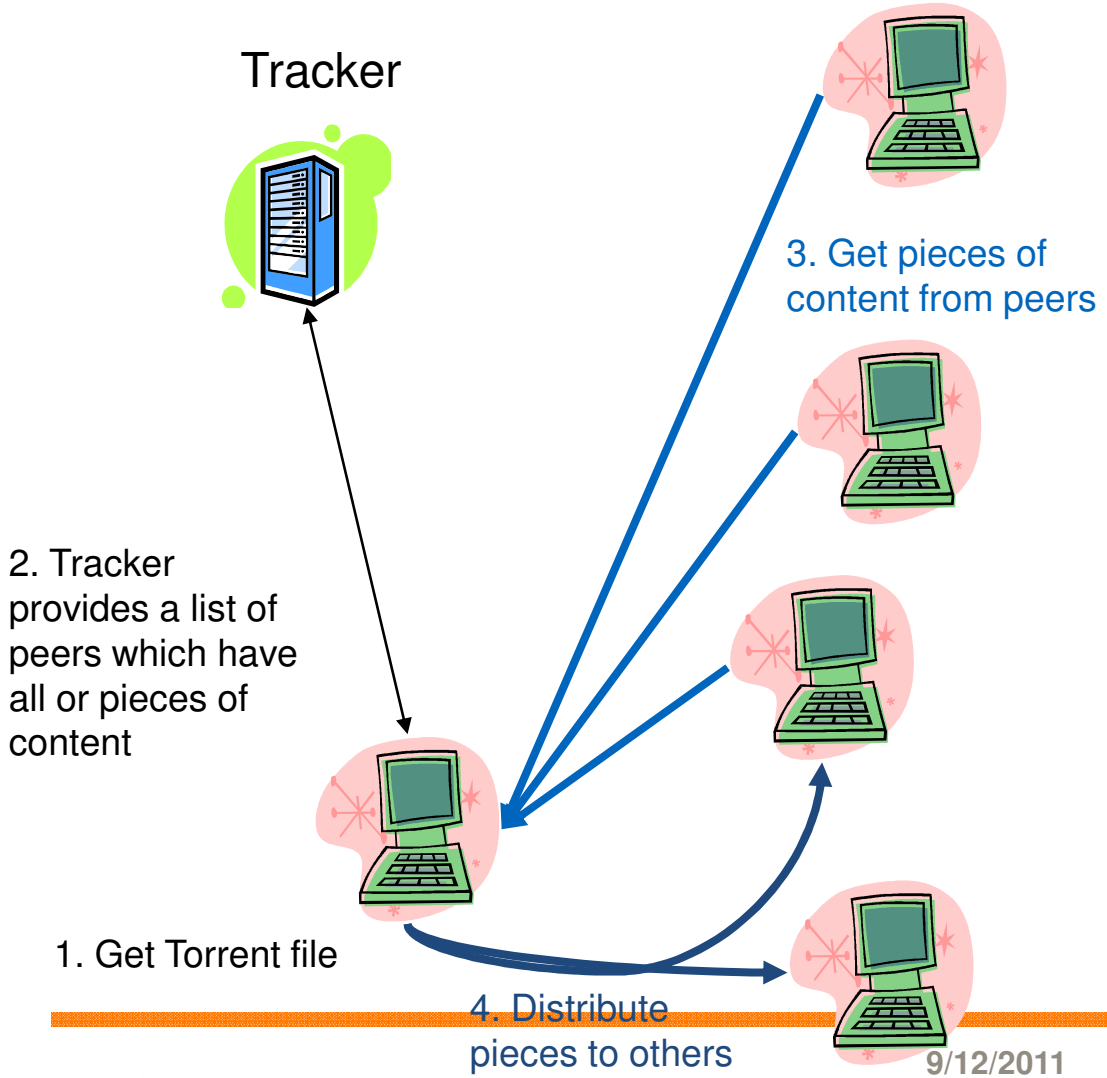
Azureus BitTorrent client



BitTorrent – content downloading

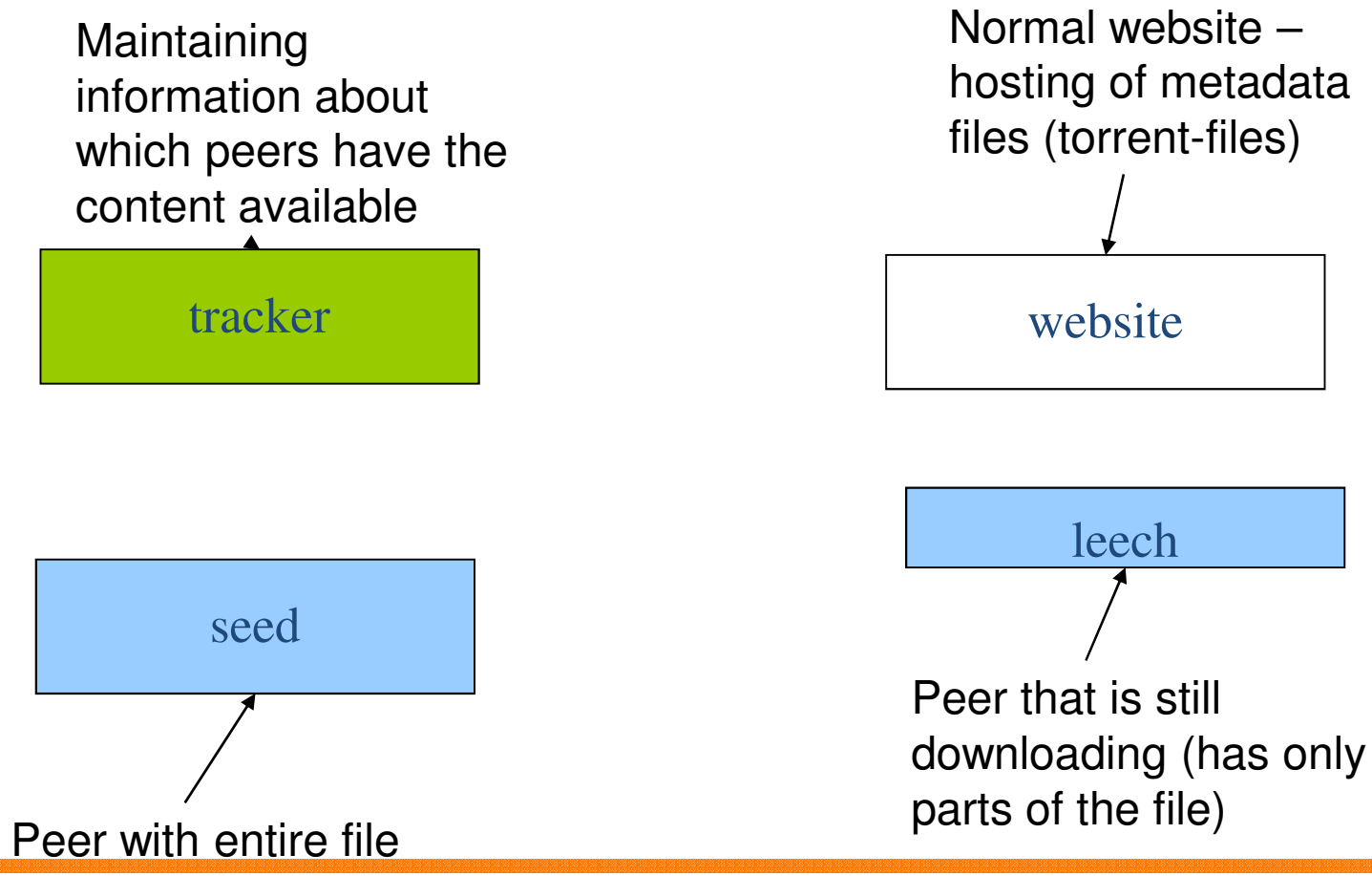
- Efficient content distribution
- Bram Cohen, 2001
- Key idea: you can receive faster than what your peer is able to send
  - Peer serving multiple users
  - Asynchronous connections
  - E2E speed of Internet
- File divided into pieces, recipient receives pieces from multiple peers
- Each recipient supplies pieces of the data to newer recipients

BitTorrent, 2002



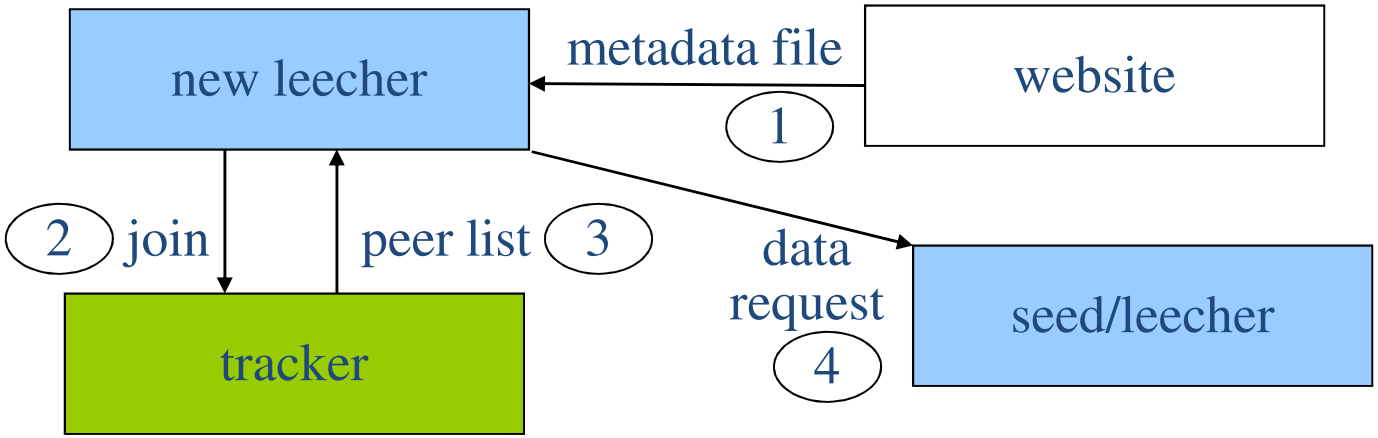
- Most popular P2P technoloy
- Targets efficient content download
- Upload capacity is the bottleneck
  - Sharers < Loaders
  - Unsymmetric link speeds
- Especially for large files (video)
- The more popular content, the better it works
- Open protocol, many open source clients, commercial services

BitTorrent – components



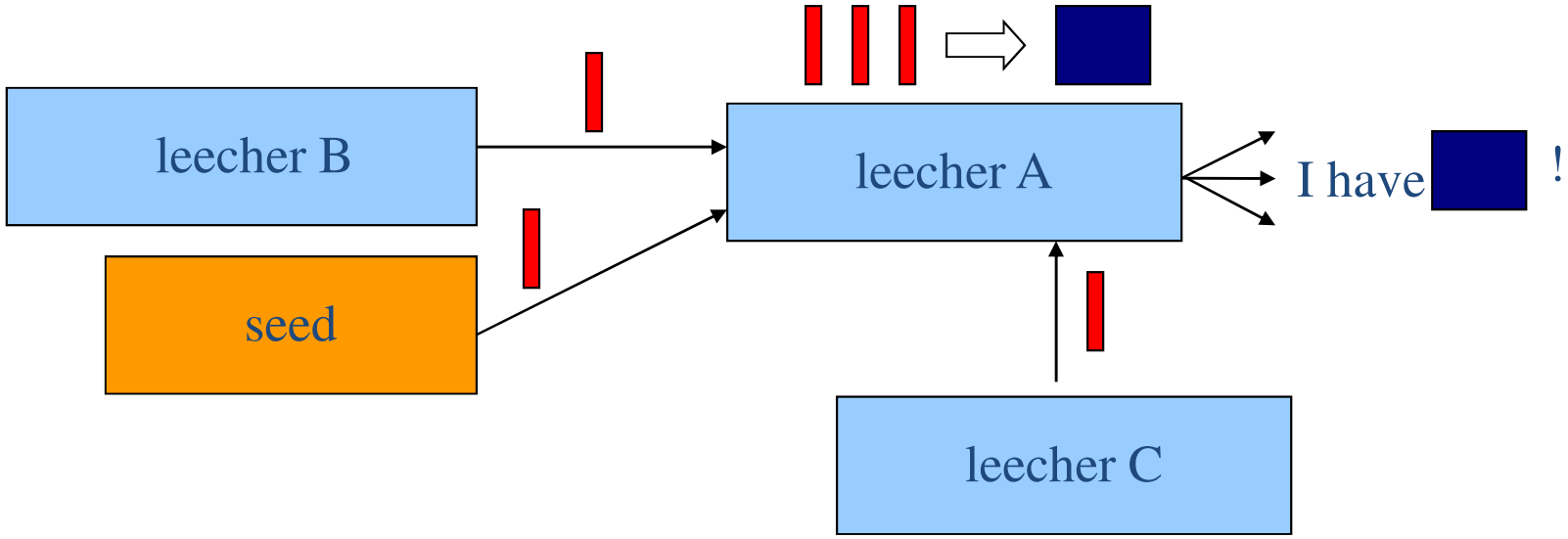
Adapted from Nikitas Liogkas, Robert Nelson, Eddie Kohler, Lixia Zhang, "Exploiting BitTorrent For Fun," University of California, Los Angeles

BitTorrent – joining a torrent



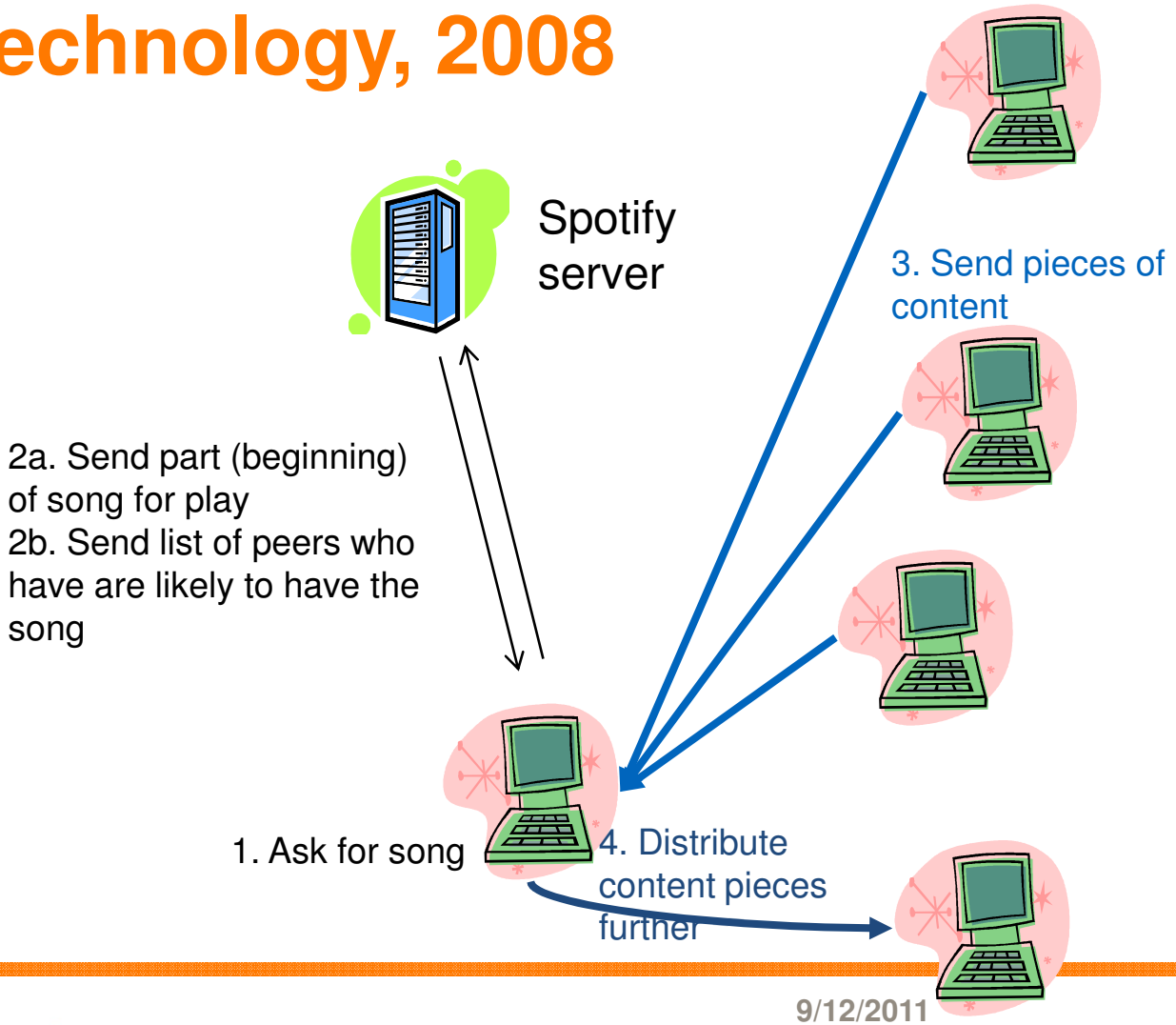
1. obtain the *metadata file (.torrent -file)*
2. contact the *tracker*
3. obtain a *peer list* (contains seeds & leechers)
4. contact peers from that list for data

BitTorrent – exchanging data



- Download sub-pieces *in parallel*
- Verify *pieces* using hashes
- *Advertise* received pieces to the entire peer list
- Look for the *rarest* pieces

Spotify music streaming – hybrid technology, 2008



BitTorrent Summary

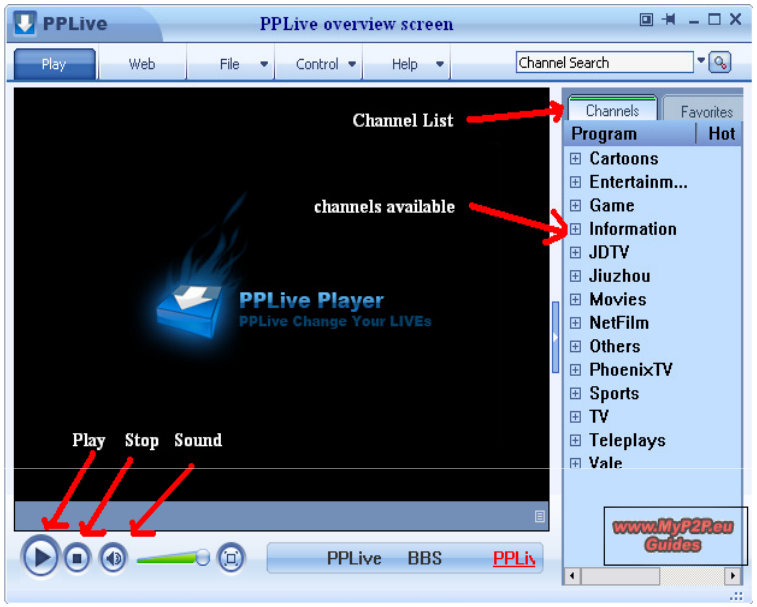
- Benefits
  - reduced cost and burden on any given individual source
  - much higher redundancy
  - greater resistance to abuse or "flash crowds“
  - less dependence on the original distributor
- Disadvantages
  - Slow start and finish
    - downloads take time to rise to full speed because peer connections take time to establish
    - Special end game algorithms
  - Full content has to be downloaded before playing can start (in most cases)
  - Central tracker can be a bottleneck
    - Distributed trackers based on DHT
- Applications
  - Legal video distribution (e.g. BitTorrent, Vuze)
  - Illegal video distribution (e.g. PirateBay)
  - Distribution of patches (e.g. Wow, Linux distros)

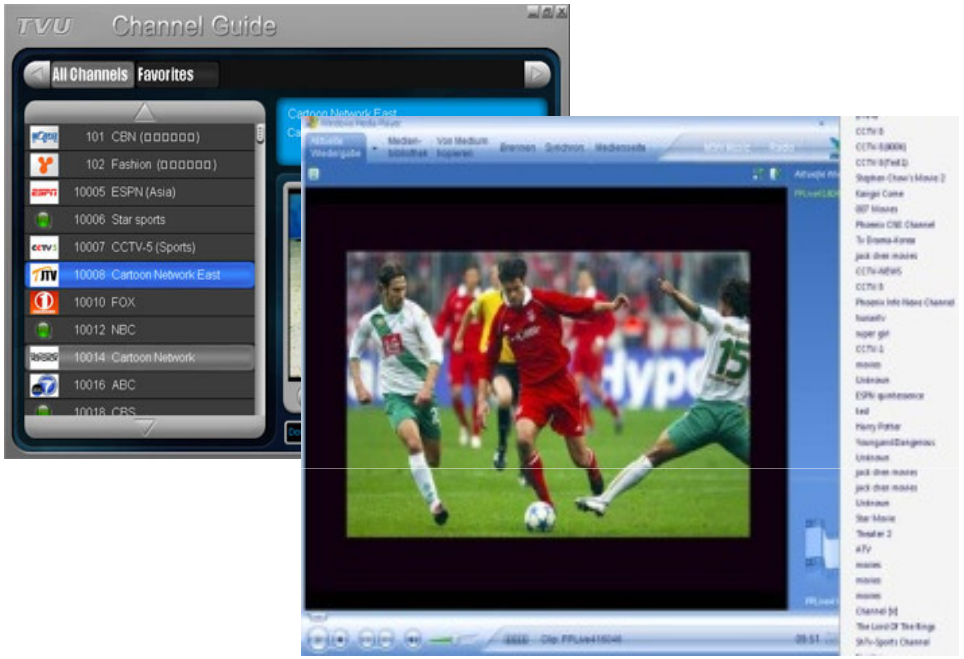
P2P Live Streaming

- “TV over the Internet”

PPLive, PPS, TVU, …

Source: http://www.synacast.com/en/





PPLive

- Founded in 2004, the first online video service provider in China.
- The largest aggregator of China TV programs with over 120 TV stations, thousands of TV shows and programs.
- Has more than 200 million user installations and its active monthly user base (as of Dec 2010) is 104 million, i.e, PPLive has a 43% penetration of Chinese internet users.
- Average viewing time per person per day has reach over 2 hours and 30 minutes.

http://cool.pptv.com/

Traditional stream delivery models

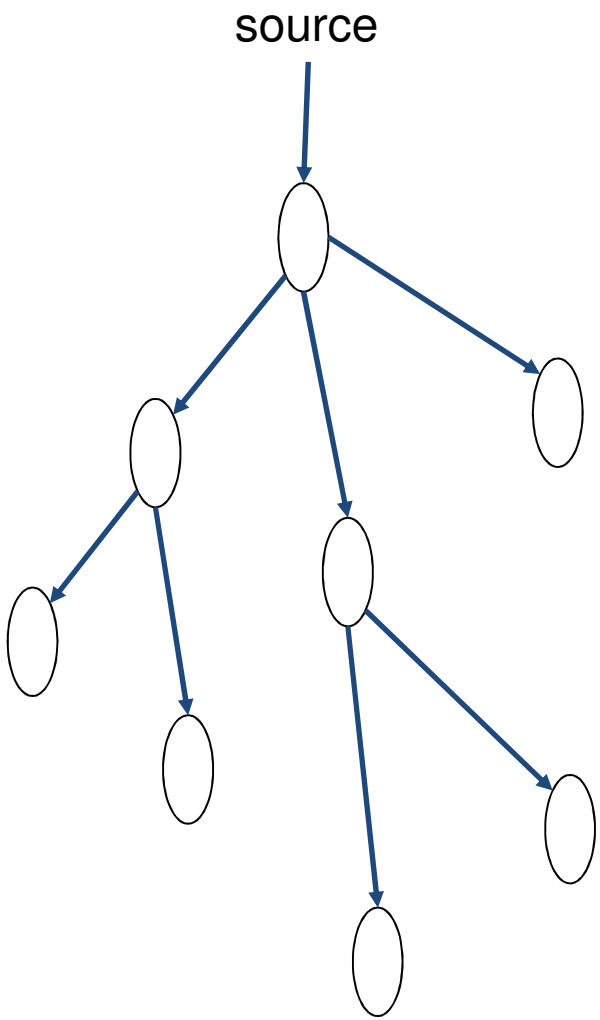
- Server
  - Widely used, simple and easy
  - Free Internet radios, YouTube, Liveleak.com, Google video, …
  - Allows using standard clients (browser)
  - Limited server output capacity / stream quality; expensive to scale
- Server grid
  - Content delivery network
  - Expensive to scale
- IP multicast / LAN multicast
  - The “ideal” model proposed for 20+ years
  - Not available in large scale Internet
    - Technical + non-technical constraints
  - Perhaps possible in local environments

P2P streaming (“peercasting”)

- Each receiver of the stream forwards it to other receivers
- Promises
  - No servers required
  - “Infinite” scalability
- Challenges
  - Churn: peers constantly join and leave the network
  - Limited peer capabilities: asymmetric data connections
  - Limited peer visibility: NAT, firewall
  - Optimal use of network resources

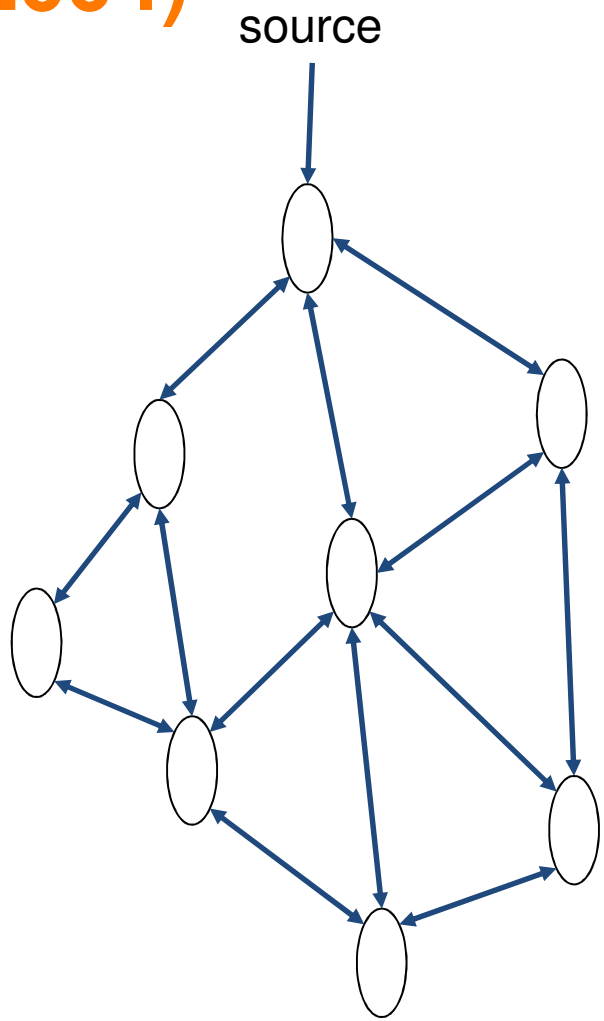
Multicast tree (ca. 2002)

- First practical approach
  - End-System Multicast II
  - Open source solutions (peercast, freecast)
  - Over 20 well-known variants
- Peers form a tree topology
  - Own tree for each data stream
  - Forward stream down the tree
- Works in practice
  - Scales 10…100…1000? users
- Problems
  - Large output bandwidth required
  - Tree optimization
  - Tree repair due to churn
  - Less than half of peers can contribute



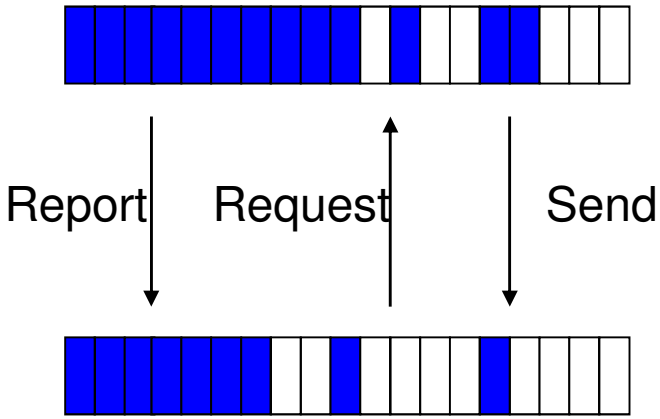
Data-driven overlay (ca. 2004)

- The mainstream practical approach
  - Active area for current research
  - Coolstreaming (2004),
    Chainsaw (2005),
    GridMedia (2006),
    PRIME (2006),
    HotStreaming (2007)
- BitTorrent for streams
  - Chunk stream in small pieces
  - Distribute pieces in a swarm
- Works well in practice
  - Most large-scale solutions
  - Coolstreaming, PPLive, Roxbeam, Sopcast …
  - Scales to 10k … 100k … 1M?



Basic data-driven overlay approach

- Coolstreaming/DONet (2004), Chainsaw (2005)
- Topology creation: gossiping protocol (SCAMP)
  - Peers maintain random partial view of the network
  - Peers select random *partners*
  - No centralized tracker
- Swarming: sliding buffer of pieces
  - Reports pieces it has to its partners
  - Partners request for pieces they don't have
- Design problems
  - Whom to select as partner?
  - When and from whom to request a piece?
  - Overhead vs. latency?



Main challenges of data-driven approach

- Open research questions
  - Based on real-life experiences with Coolstreaming and 80k users
  - Affect negatively to end-user experience
- Dealing with flash crowd
  - How to cope if number of users increases from 1k to 100k in 10 minutes?
  - We don't have infrastructure to support new users
  - Joining takes a long time
  - > 25% of new users must re-try joining
- Dealing with 50% of users that don't contribute
  - Due to asymmetric connection, firewall, NAT, …
  - Where to get the missing output capacity?

Hybrid technology

- The best known technology for commercial large-scale streaming
  - Streaming to 100k … 1M users
  - Proposed practical solution to problems of data-driven overlay
- A combination of P2P and server grid
  - Use P2P distribution in stable conditions
  - Use powerful servers to fill in missing output capacity
  - Servers support newcomers
  - Servers support users behind asymmetric connections

Contact Information

- Course web page:
  - https://noppa.aalto.fi/noppa/kurssi/t-110.5150/etusivu
- Contact email:
  - zhonghong.ou@aalto.fi
- Office hour:
  - Fri 10-11 room A 109
- Qustions & Suggestions?