Peer-to-Peer Networking

Case Study: BitTorrent

Lecture Content

- BitTorrent (protocol version 1.0)
  - File transfer protocol
  - Novel techniques for distributing content
- P2P system studied further
  - The original design
  - Later enchantments are not discussed here

BitTorrent (BT)

- BT is a file transfer protocol for content distribution
  - Protocol specifications v1.0 studied here
- A centralised P2P system (compare to Napster)
  - A tracker server managing users' downloads
- BitTorrent concentrates on efficient file transfer
- Searching content is not provided by the protocol specification, but out-of-band methods
- Addresses the free riding problem in P2P file sharing

BT Terminology (1/3)

- Torrent
  - Set of peers cooperating to download the same content using BT protocol
- Tracker
  - Centralised server keeping track of current participants. Does not involve to data transfers, but collect statistics
- Pieces and blocks
  - File is cut into fixed size pieces (typically 256 KB) and pieces are further cut into blocks (transfer unit, 16 KB)

BT Terminology (2/3)

- Metainfo file, or .torrent file
  - Contains information about the file, its length, name and the address and port of the tracker
  - Hashes for pieces of files for verification
- Interested and Choked
  - A is marked as *interested* in peer B when B has pieces that A wants, and vice versa. Also, A is *choked*, when B decides not to upload data when A is interested. When B is willing to upload again, A is *unchoked*.

BT Terminology (3/3)

- Peer set or a swarm
  - The group of machines that are collectively connected for a particular file, i.e. a list of open TCP connections
- Leecher and seed
  - Leecher is a peer that is still downloading the pieces. Seed is a peer holding the complete file for uploading
- Choking algorithms
  - Strategy to select which peers to choke and which to unchoke – *tit-for-tat algorithm*

Publishing Content

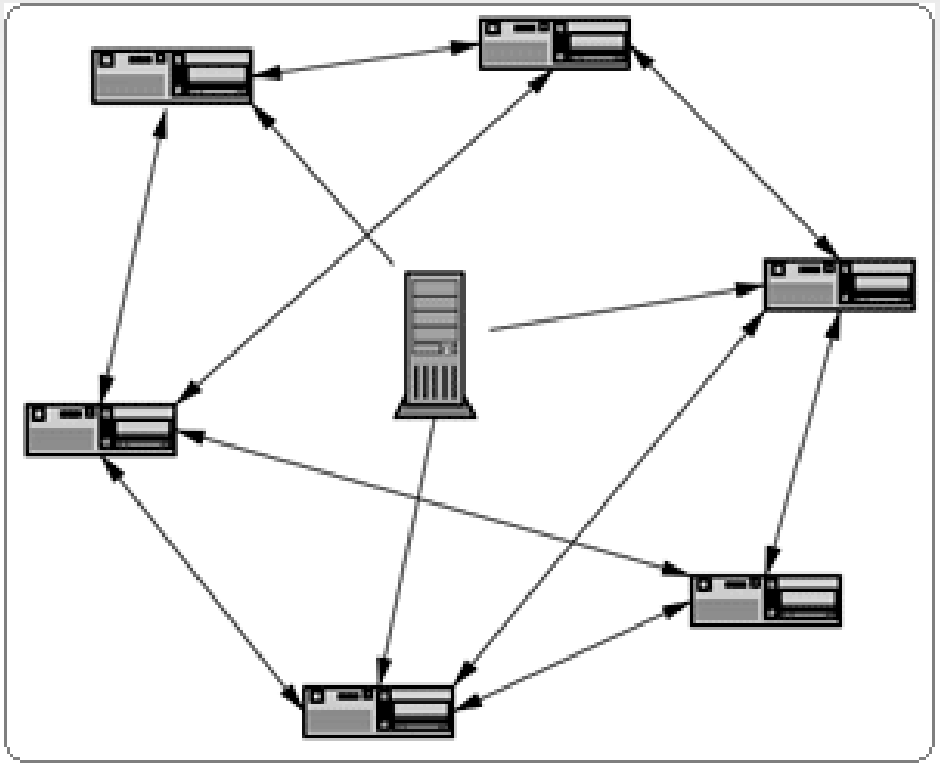
- A static file with the extension *.torrent* put on a web server
  - How to contact tracker and needed metadata for the file
- Trackers responsible for helping peers to download
  - Very simple negotiation protocol on top of HTTP
  - Peer asks information about other peers for a file
  - Tracker replies with a list of other peers (e.g. 50 peers)
  - Initial seed (a complete copy) must be available

Peer Distribution

- Peers maintain a list of pieces they are holding
  - Exchange and update the lists after the handshake
- Continuously download blocks from connected peers which have wanted pieces and allow download
  - Two or more blocks assembled to a piece and verified
- As transferring continues, peers have more and more pieces and can provide more aggregate upload capacity

BitTorrent Deployment

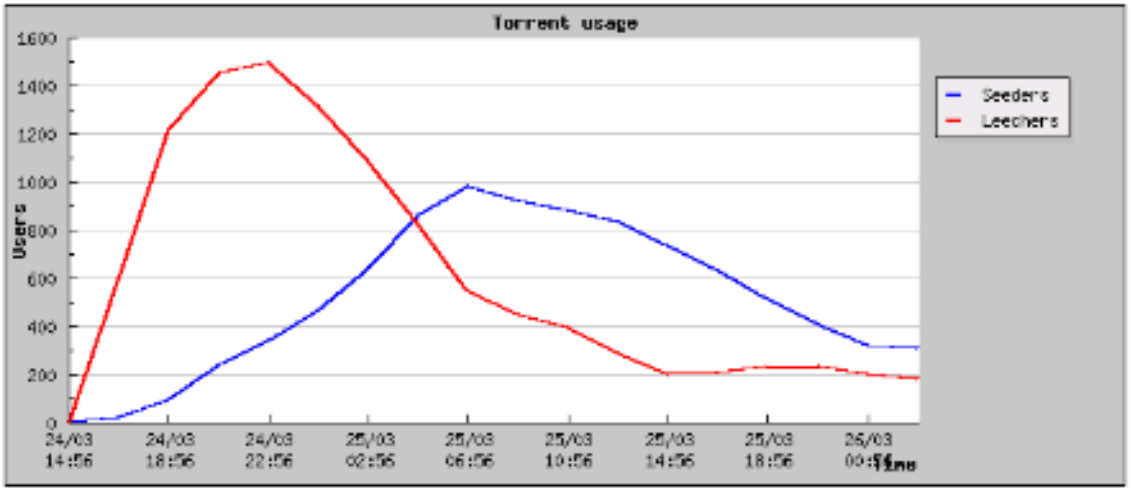
- The tracker is on the centre
- Peers asks about other peers involved to the distribution of file stated in .torrent file
- Peers with common interest form a swarm, or a peer set
- Transfer happens directly between peers according to downloading strategies (discussed later)



[http://www.bittorrent.org/introduction.html]

BitTorrent Deployment

- In a typical deployment, the number of leechers increases very rapidly after the file is made available. The peak of leechers passes as they complete and leave. It eventually peaks and then falls off at a roughly exponential rate.
- The number of seeders increases slowly, peaks some time after the number of leechers does, then also falls off exponentially. The exponential falloff of both reflects the rate of new downloaders joining after the initial rush is over.



[Bram Cohen, Incentives Build Robustness in BitTorrent]

Pipelining

- When data is being transferred, downloaders should keep several piece requests queued up at once in order to get good TCP performance
- On the other side, requests which can't be written out to the TCP buffer immediately should be queued up in memory rather than kept in an application-level network buffer, so they can all be thrown out when a choke happens.

Piece Selection Policies

- Performance depends highly on how the pieces are distributed
  - End up with pieces which are currently on offer?
  - Initial seeder leaves and no one fetched the end part?
- Strict policy
  - Once a block has been requested, all block from that piece must be fetched before requesting from another piece
  - Getting complete pieces as quickly as possible

Ensuring piece availability

- Rarest first: select piece which peers have least
  - Makes sure that peers have pieces which all of their peers want, so uploading can be done when wanted
  - Avoiding the risk that initial seed is removed before other seed are available and rare pieces are not found
- Random first piece
  - Exception to the rarest first when download started
  - Pick the first randomly to get it faster than a rare piece

Finishing the Transfer

- Requested piece from a peer with slow transfer rate
  - Potentially delays the finishing of download
- Endgame mode: Finish download quickly
  - Send requests for blocks of the final piece to all peers
  - Cancels sent for arrived blocks to prevent wasting too much bandwidth on redundant sends
  - In practise, endgame period is very short, not wasting too much bandwidth, and file is finished quickly

Tit-for-Tat Algorithm

- Peers try to download whoever they can and decide which peers upload to via tit-for-tat algorithm
  - To cooperate, peers upload, and not to cooperate, they "choke" peers
  - "If peers stop cooperating, I will not cooperate and if peer cooperates, I will cooperate"
  - Always start in cooperating mode
- Ensuring that peers need to upload to in order to be able to download: both receive benefit

Choking algorithm

- No central resource allocation
  - Each peer is responsible for maximising its own download rate
- Why choke peers?
  - TCP congestion control behaves very poorly when sending over many connections at once
  - Choking lets each peer use a tit-for-tat algorithm to ensure that they get a consistent download rate

Choking algorithm

- Always unchoke a fixed number of peers (e.g. 4)
- How to select peers to unchoke in order to maximise transfer performance?
  - Based strictly on current download rate
  - For meaningful rate, use a rolling average (20 seconds)
- Avoid “fibrillation”
  - Choking and unchoking quickly wastes resources
  - Recalculate new unchokes at 10 seconds intervals

Optimistic Unchoke

- How to discover if currently unused connections are better than the ones being used?
- At any one time there is a single peer which is unchoked regardless of it's upload rate
- Which peer is optimistically unchoked rotates every 30 seconds
- New connections are three times as likely to start as the current optimistic unchoke as anywhere else in the rotation

Anti-snubbing

- Occasionally a peer will be choked by all peers which it was formerly downloading from
  - Peer will usually continue to get poor download rates until the optimistic unchoke finds better peers
- When over a minute goes by without getting a single piece from a particular peer, assume being “snubbed”
  - Don't upload to that peer except as an optimistic unchoke
- Frequently results in more than one concurrent optimistic unchoke: download rates recover quickly

Upload Only (Seed)

- Once peer has completed downloading, unchoke selection is based on the upload rates
  - Utilising all available upload capacity
  - Preferring peers no one else seems to be uploading at the moment
- For efficient and fair BT usage, participants are often advised to keep it running awhile after finishing download

BitTorrent Conlusions

- Proved to work quite well
  - Very popular, claimed to account majority of the current P2P traffic and a considerable part of Internet traffic
  - End of 2004 BT accounting 30% of all Internet traffic according to Cache Logic (http://www.cachelogic.com/)
- Demands that all users participate in the P2P spirit
  - Tit-for-tat strategy favours cooperating peers
- Centralising involves only the tracker
  - Anyone can set up their own

BitTorrent in Practise

- Does not provide any search mechanism for content
  - Have to find the .torrent file
  - Many web pages provide indexes and search engines
- Peers often have little incentive to stay seeders
  - Torrent swarms gradually die out
- Typical BitTorrent download will gradually ramp up to very high speeds and then slowly ramp back down toward the end of the download

Lecture Considerations

- What makes BitTorrent such a successful network?
- BitTorrent and content
  - Trackers don't host files, but still enable the downloads
    - The infamous Pirate Bay BitTorrent tracker site
  - On the other hand, often used for distribute open source, like Linux ISO images (can be multiple CDs)