# peer-to-peer and agent-based computing
BitTorrent

## Plan of lecture
- BitTorrent Background
  - What is BitTorrent?
  - BitTorrent Operation
  - The author and its history
- The Protocol
  - Terminology
  - Distributed Scenario
  - Structure of .torrent files
  - Protocol between peers and trackers
- BitTorrent Applications
  - Uses in Industry

## What is BitTorrent?
- A protocol for distributing files
- File searching based on the Web
  - content is identified by URL and is designed to integrate seamlessly with the Web (various new enhancements)
- Different from HTTP, though:
  - When multiple downloads of the same file happen concurrently, the downloaders upload to each other
  - Files are broken into pieces, pieces are exchanged between users until everyone has a complete set (TCP)
  - A tracker is used to update the network as more pieces are downloaded, which results in continuous "file swarming"
- Multiple applications, such as
  - File sharing
  - P2P TV
  - Content distribution
- "BitTorrent, plc." is now serious technology!
  - No illicit content
  - Solid/serious "business model" (i.e., £££, $$$, €€€, etc.)

## BitTorrent Operation

- To share a file or group of files, a peer first creates a small file called a “torrent” (e.g. MyFile.torrent).
  - This file contains metadata about the files to be shared and about the tracker, the computer that coordinates the file distribution.
- Peers that want to download the file must
  - Obtain a torrent file for it, and then
  - Connect to the specified tracker, which tells them from which other peers to download the pieces of the file.

## BitTorrent Operation (Cont’d)

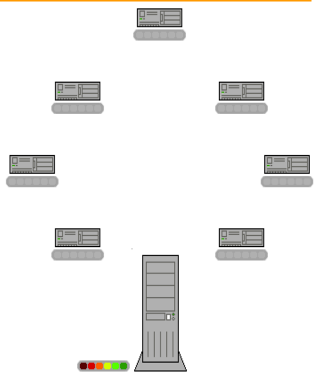
- Though it ultimately transfer files over a network, a BitTorrent download differs from a classic download in several fundamental ways:
  - BitTorrent makes many small data requests over different TCP connections to different machines, while classic downloading is typically made via a single TCP connection to a single machine.
  - BitTorrent downloads in a random or in a “rarest-first” approach that ensures high availability, while classic downloads are sequential.
- Taken together, these differences allow BitTorrent to
  - Achieve much lower cost to the content provider,
  - Much higher redundancy, and
  - Much greater resistance to abuse or to “flash crowds” than regular server software

## BitTorrent Operation (Cont’d)

- This protection comes at a cost:
  - Downloads can take time to rise to full speed because it may take time for enough peer connections to be established
  - It takes time for a node to receive sufficient data to become an effective uploader
  - A typical BitTorrent download will gradually rise to very high speeds, and then slowly fall back down toward the end of the download
- This contrasts with regular downloads (such as from an HTTP server, for example) that, while more vulnerable to overload and abuse, rises to full speed very quickly and maintains this speed throughout

## BitTorrent Operation (Cont'd)

- Coloured bars beneath all of 7 clients in the upper region above represent individual pieces of the file
- After the initial pieces transfer from the seed (large system at the bottom), the pieces are individually transferred from client to client
- Original seeder only needs to send out one copy of the file for all the clients to receive a copy



## Creating & publishing torrents

- Peer distributing a data file treats file as a number of identically sized pieces
  - Typically between 64 KB and 4 MB each.
- Peer creates a checksum for each piece, using the SHA1 hashing algorithm, and records it in the torrent file
  - Pieces with sizes greater than 512 KB will reduce the size of a torrent file for a very large payload, but may reduce efficiency of protocol
- When another peer later receives a particular piece, the checksum of the piece is compared to the recorded checksum to test that the piece is error-free
- Peers that provide a complete file are called seeders, and the peer providing the initial copy is called the initial seeder
- The exact information contained in the torrent file depends on the version of the BitTorrent protocol.
  - By convention, the name of a torrent file has the suffix .torrent.
  - Torrent files have an "announce" section, which specifies the URL of the tracker, and an "info" section, containing (suggested) names for the files, their lengths, the piece length used, and a SHA-1 hash code for each piece, all of which are used by clients to verify the integrity of the data they receive.

## Creating & publishing torrents (Cont'd)

- Torrent files are typically published on websites or elsewhere, and registered with a tracker.
- The tracker maintains lists of the clients currently participating in the torrent.
- Alternatively, in a trackerless system (decentralised tracking) every peer acts as a tracker
  - Azureus was the first BitTorrent client to implement such a system via a distributed hash table (DHT) method
  - An alternative and incompatible DHT system, known as Mainline DHT, was later developed and adopted by the BitTorrent (Mainline) and other clients.

## Downloading torrents and sharing files

- Users browse the web to find a torrent of interest, download it, and open it with a BitTorrent client.
- The client connects to the tracker(s) specified in the torrent file, from which it receives a list of peers currently transferring pieces of the file(s) specified in the torrent.
- The client connects to those peers to obtain the various pieces. If the swarm contains only the initial seeder, the client connects directly to it and begins to request pieces.
- Clients incorporate mechanisms to optimise their download and upload rates;
  - For example they download pieces in a random order to increase the opportunity to exchange data, which is only possible if two peers have different pieces of the file.

## Downloading torrents & sharing… (Cont'd)

- The effectiveness of this data exchange depends largely on the policies that clients use to determine to whom to send data
- Clients may prefer to send data to peers that send data back to them (a tit for tat scheme), which encourages fair trading.
- Strict policies often result in suboptimal situations, such as
  - when newly joined peers are unable to receive any data because they don't have any pieces yet to trade themselves
  - when two peers with a good connection between them do not exchange data simply because neither of them takes the initiative
- To counter these effects, the official BitTorrent client program uses a mechanism called "optimistic unchoking":
  - Client reserves a portion of its available bandwidth for sending pieces to random peers (not necessarily known good partners, so called preferred peers) in hopes of discovering even better partners and to ensure that newcomers get a chance to join the swarm

## Downloading Speeds

Download speeds depend on two factors:

- BitTorrent keeps track of how much you contribute to hosting files for the group
  - The more you share, the faster you download
- The more people trading a file, the more options for obtaining its pieces
  - So, unlike the old Napster, popularity doesn't bog down the process – it gives it a shot of adrenaline
  - Trackers also more dynamic than Napster servers - provide updates

## BitTorrent's Author: Bram Cohen

See his entry in Wikipedia:
http://en.wikipedia.org/wiki/Bram_Cohen

- Engineered large parts of MojoNation
  - Parts of it similar to BitTorrent
- In 2001
  - Authored BitTorrent protocol
  - Wrote first BitTorrent client (in Python)

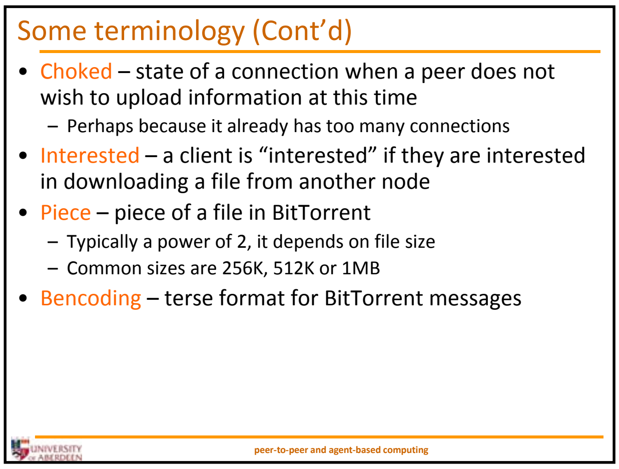
## BitTorrent's First "Public Appearance"

- Cohen unveiled his ideas at the 1st CodeCon
  - In 2002 (http://www.codecon.org/2002/)
  - A conference for hackers & technology enthusiasts
  - Co-organised by Bram and his roommate
  - Intended to be a low cost conference (i.e. < US$ 100)
  - Focus on developers doing presentations of working code
    - Rather than on companies with products to sell
  - Remains an event to find out new directions in software
  - BitTorrent continues to lay claim to the title of "most famous presentation"

CodeCon 2002

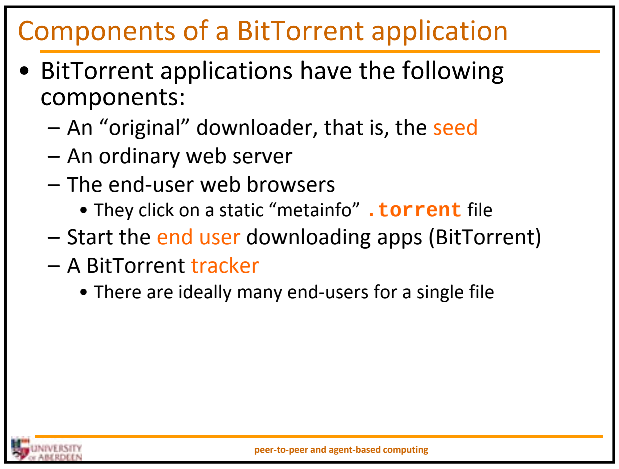
## Some terminology

- Torrent – metadata file containing information about file to be shared via BitTorrent
- Peer – a participant in the network
- Seed – peer that has complete copy of file
  - Who probably created the torrent
- Swarm – peers connected to a particular file
  - All peers of swarm interested in a particular file
- Tracker – server responsible for keeping track of people in a swarm



## Some terminology (Cont'd)

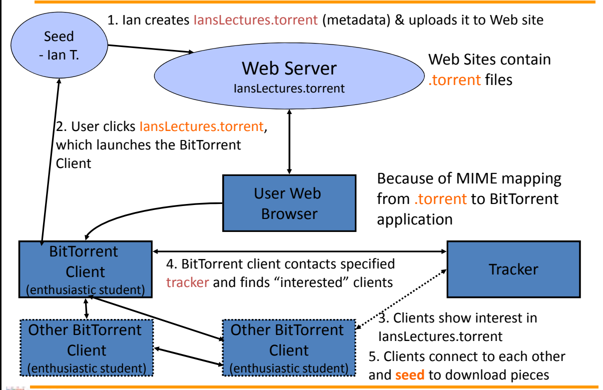
- Choked – state of a connection when a peer does not wish to upload information at this time
  – Perhaps because it already has too many connections
- Interested – a client is "interested" if they are interested in downloading a file from another node
- Piece – piece of a file in BitTorrent
  – Typically a power of 2, it depends on file size
  – Common sizes are 256K, 512K or 1MB
- Bencoding – terse format for BitTorrent messages



## Components of a BitTorrent application

- BitTorrent applications have the following components:
  – An "original" downloader, that is, the seed
  – An ordinary web server
  – The end-user web browsers
    - They click on a static "metainfo" `.torrent` file
  – Start the end user downloading apps (BitTorrent)
  – A BitTorrent tracker
    - There are ideally many end-users for a single file

## Example: a lecture


Seed
- Ian T.
1. Ian creates IansLectures.torrent (metadata) & uploads it to Web site
Web Server
IansLectures.torrent
Web Sites contain .torrent files
2. User clicks IansLectures.torrent, which launches the BitTorrent Client
User Web Browser
Because of MIME mapping from .torrent to BitTorrent application
BitTorrent Client (enthusiastic student)
4. BitTorrent client contacts specified tracker and finds "interested" clients
Tracker
Other BitTorrent Client (enthusiastic student)
Other BitTorrent Client (enthusiastic student)
3. Clients show interest in IansLectures.torrent
5. Clients connect to each other and seed to download pieces

## BitTorrent messages: Bencoding

- Bencoding is a way to specify and organise data in a terse format
- It supports 4 types:
  - Strings encoded as <string length>:<string data>
    - E.g., `4:spam` represents the string `spam` with 4 characters
  - Integers encoded as i<integer>e
    - E.g., `i3e` represents the integer `3`
  - Lists encoded as l<bencoded values>e
    - E.g., `l4:spam4:eggse` represents the list [`spam`, `eggs`]
  - Dictionaries encoded as

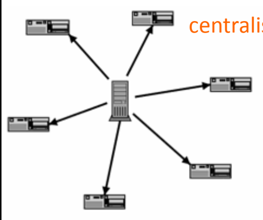
    d<bencoded string><bencoded element>e
    - Keys must be bencoded strings
    - E.g., `d4:spaml1:a1:bee` represents the dictionary `{spam => [a,b]}`

## .torrent files

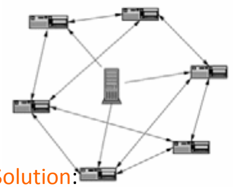
The content of a ".torrent" is a bencoded dictionary, containing:

- announce: The URL of the tracker (string)
  - Later versions have lists of trackers
- info: a dictionary that describes the file(s) of the torrent, containing:
  - Name – the name for the file
  - Piece length: number of bytes in each piece (integer)
  - Pieces: string consisting of the concatenation of all 20-byte SHA1 hash values, one per piece (byte string)
- Format changes if there's one file (as above) or many
  - Where there are files occurrences of the above information (piece length and pieces), and
  - Path is used to replace name for uniqueness

## BitTorrent Trackers



centralised: all clients go to one server

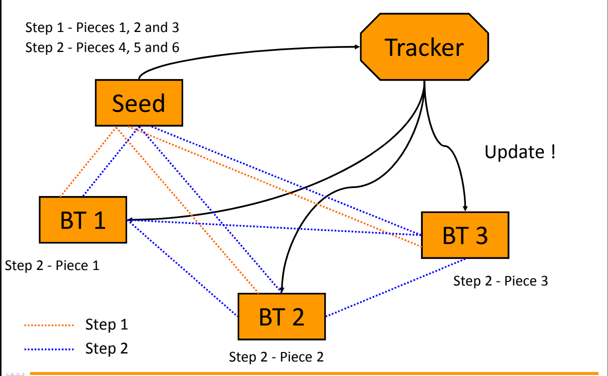


BitTorrent Solution:

customers help distribute content

- Their contribution grows at the same rate as their demand, creating limitless scalability for a fixed cost;
- The tracker maintains the process

## Tracker scenario



## Tracker GET request

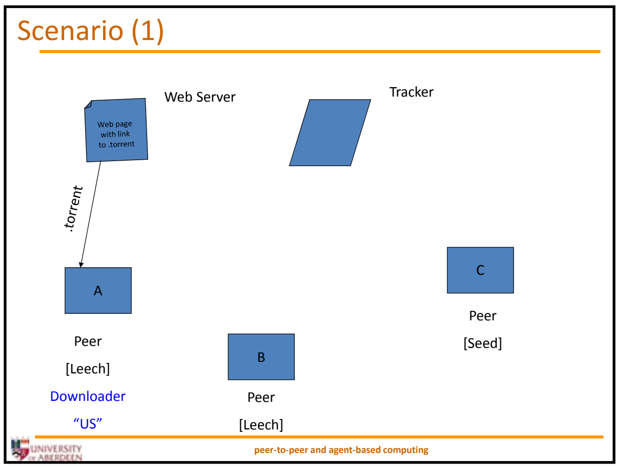
Peer → Tracker

- Info_hash – 20-byte SHA1 hash of the bencoded form of the info value from the metainfo file
- Peer_id – string of length 20 containing ID of downloader, generated at random at the start of a new download
- IP – IP (or DNS name) of peer
- Port – port number for the peer
  - Tries port 6881 and if that port is taken try 6882, then 6883, etc. and give up after 6889
- Uploaded – total amount uploaded so far
- Downloaded – total amount downloaded so far
- Left – number of bytes this peer still has to download
- Event – optional key which maps to started, completed, or stopped (or empty, which is the same as not being present)

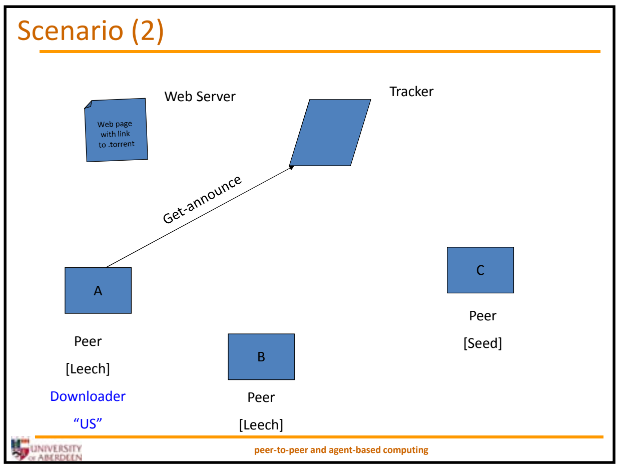
## Tracker response

Tracker → Peer

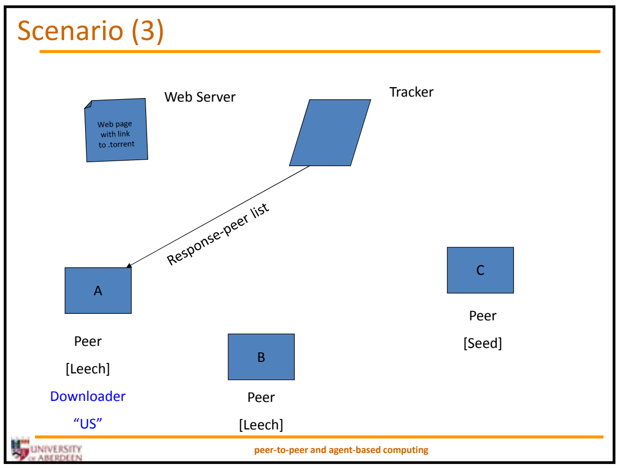
- Tracker responses are bencoded dictionaries
- If a tracker response has a key failure reason, then that maps to a human readable string which explains why the query failed, and no other keys are required
- Otherwise, it must have two keys:
  - Interval which maps to the number of seconds the downloader should wait between regular rerequests
  - Peers maps to a list of dictionaries corresponding to peers, each of which contains the keys peer id, ip, and port, which map to the peer's self-selected ID, IP address or DNS name as a string, and port number, respectively.



## Scenario (1)

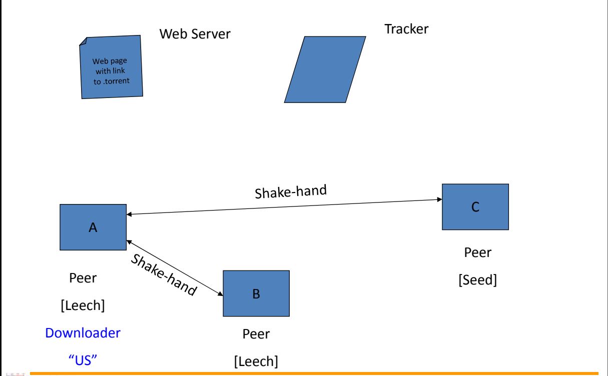


## Scenario (2)

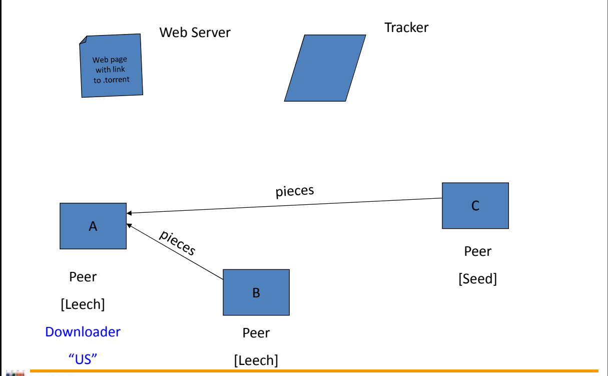


## Scenario (3)

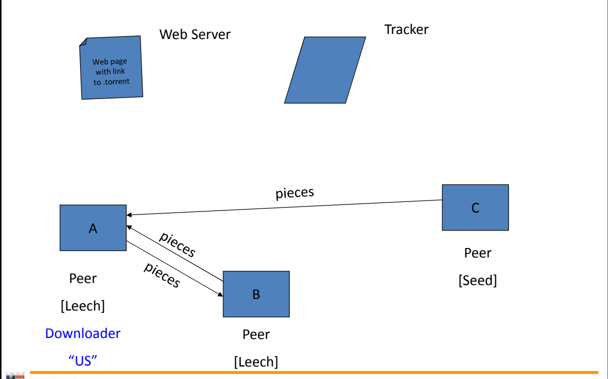
## Scenario (4)



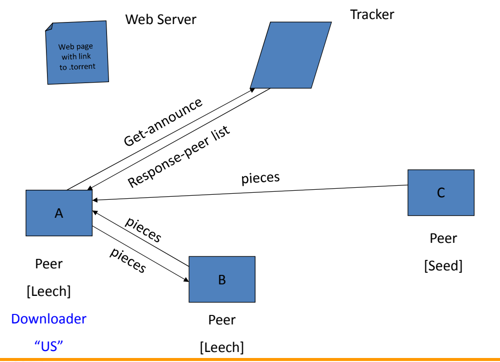
## Scenario (5)



## Scenario (6)



## Scenario (7)

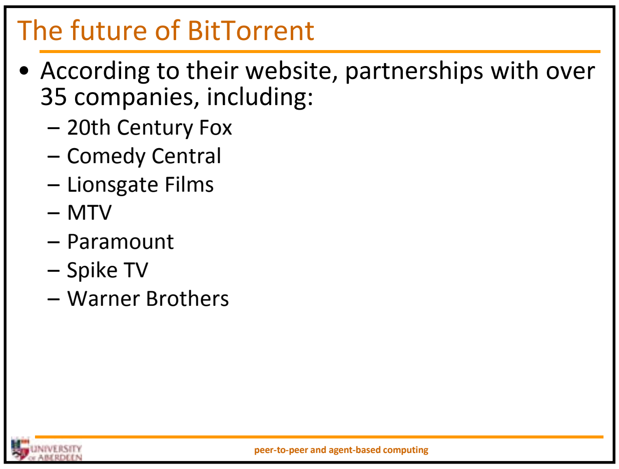


## Advantages

- Better bandwidth utilization
  - Unprecedented speeds (7 MB/s on the Internet)
- Limit "free-riding" (tit-for-tat)
- Limit "leech attack"
  - Coupling upload & download
- Spurious files not propagated
- Ability to resume a download
- Open source implementations!

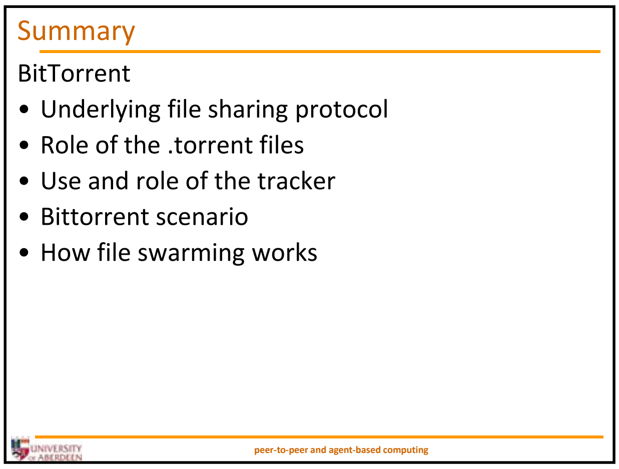
## Disadvantages

- Lack of anonymity
- Small files lead to latency and overheads
- Random list of peers (naïve)
- Scalability
  - Millions of peers – tracker behavior (uses 1/1000 of bandwidth)
  - Single point of failure – although there can be many trackers, there is only one tracker assigned to each torrent file
    - Difficult to load balance
- Robustness
  - System progress depends on altruistic nature of seeds (and peers)
  - Malicious attacks and leeches



## The future of BitTorrent

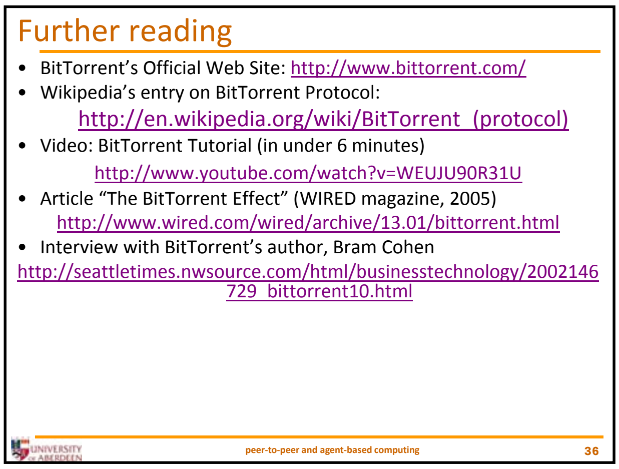
- According to their website, partnerships with over 35 companies, including:
  - 20th Century Fox
  - Comedy Central
  - Lionsgate Films
  - MTV
  - Paramount
  - Spike TV
  - Warner Brothers



## Summary

BitTorrent

- Underlying file sharing protocol
- Role of the .torrent files
- Use and role of the tracker
- Bittorrent scenario
- How file swarming works



## Further reading

- BitTorrent's Official Web Site: http://www.bittorrent.com/
- Wikipedia's entry on BitTorrent Protocol:
  http://en.wikipedia.org/wiki/BitTorrent_(protocol)
- Video: BitTorrent Tutorial (in under 6 minutes)
  http://www.youtube.com/watch?v=WEUJU90R31U
- Article "The BitTorrent Effect" (WIRED magazine, 2005)
  http://www.wired.com/wired/archive/13.01/bittorrent.html
- Interview with BitTorrent's author, Bram Cohen
  http://seattletimes.nwsource.com/html/businesstechnology/2002146729_bittorrent10.html