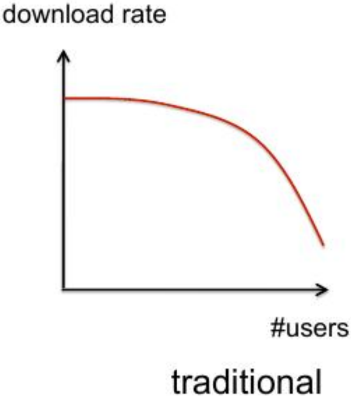
CS711: Introduction to Game Theory and Mechanism Design

Teacher: Swaprava Nath

P2P file sharing
slides adapted from CS186 Harvard

Peer to Peer



Desired Properties and Terminology

- Scalability
- Failure resilience

Terminology:

- **Protocol:** messages that can be sent, actions that can be taken over the network
- **Client:** a particular process for sending messages, taking actions
- **Reference client:** particular implementation
- **Peer**

Early P2P Technologies

Napster (1999 - 2001)

- Centralized database
- Users download music from each other

Early P2P Technologies

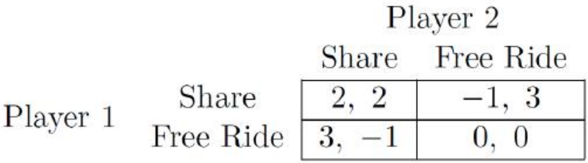
Napster (1999 - 2001)

- Centralized database
- Users download music from each other

Gnutella (2000 -)

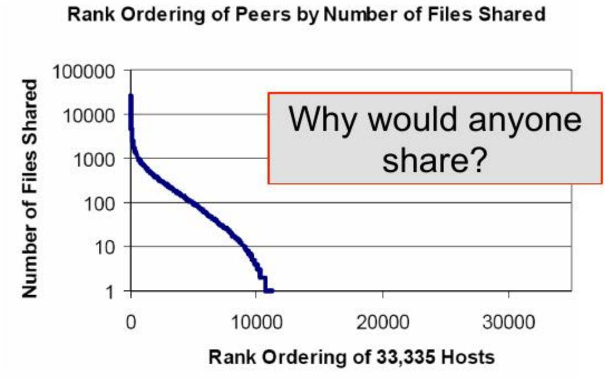
- Get list of IP addresses of peers from set of known peers (no server)
- To get a file: Query message broadcast by peer A to known peers
- Query response: sent by B if B has the desired file (routed back to requestor)
- A can then download directly from B

The File Sharing Game



		Player 2	
		Share	Free Ride
Player 1	Share	2, 2	−1, 3
	Free Ride	3, −1	0, 0

The File Sharing Game (Contd.)



Why would anyone share?

Image courtesy: Adar and Huberman (2000)

Incentives for Client Developers

- Client developers can ensure file sharing
- But competition among the developers

Incentives for Client Developers

- Client developers can ensure file sharing
- But competition among the developers

- 85% peers free-riding by 2005; Gnutella less than 1% of ww P2P traffic by 2013
- Few other P2P systems met the same fate

New Protocol

BitTorrent (2001 -)

- Approx 85% of P2P traffic in US
- File sharing
- Also used for S/W distribution (e.g., Linux)

New Protocol

BitTorrent (2001 -)

- Approx 85% of P2P traffic in US
- File sharing
- Also used for S/W distribution (e.g., Linux)

Key innovations

- Break file into pieces: A repeated game!
- "If you let me download, I'll reciprocate."

BitTorrent Schematic

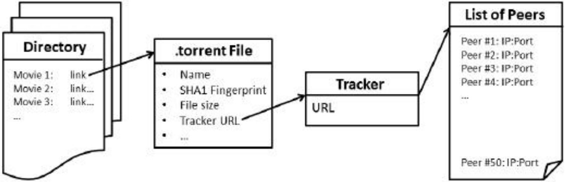


Figure 5.4.: Starting a download process in the BitTorrent protocol: 1) A user goes to a searchable directory to find a link to a .torrent file corresponding to the desired content; 2) the .torrent file contains metadata about the content, in particular the URL of a tracker; 3) the tracker provides a list of peers participating in the swarm for the content (i.e., their IP address and port); 4) the user's BitTorrent client can now contact all these peers and download content.

Image courtesy: Parkes and Seuken (2017)

BitTorrent Optimistic Unchoking Algorithm

Tracker is a centralized entity that controls the traffic, tracks the connection between peers and their speed of upload, download etc.

BitTorrent Optimistic Unchoking Algorithm

Tracker is a centralized entity that controls the traffic, tracks the connection between peers and their speed of upload, download etc.

Reference Client Protocol:

- Set a threshold r of uploading speed (typically the third maximum speed in the recent past)

BitTorrent Optimistic Unchoking Algorithm

Tracker is a centralized entity that controls the traffic, tracks the connection between peers and their speed of upload, download etc.

Reference Client Protocol:

- Set a threshold r of uploading speed (typically the third maximum speed in the recent past)
- If a peer j uploaded to i at a rate $\geq r$, unchoke j in the next period

BitTorrent Optimistic Unchoking Algorithm

Tracker is a centralized entity that controls the traffic, tracks the connection between peers and their speed of upload, download etc.

Reference Client Protocol:

- Set a threshold r of uploading speed (typically the third maximum speed in the recent past)
- If a peer j uploaded to i at a rate $\geq r$, unchoke j in the next period
- If a peer j uploaded to i at a rate $< r$, choke j in the next period

BitTorrent Optimistic Unchoking Algorithm

Tracker is a centralized entity that controls the traffic, tracks the connection between peers and their speed of upload, download etc.

Reference Client Protocol:

- Set a threshold r of uploading speed (typically the third maximum speed in the recent past)
- If a peer j uploaded to i at a rate $\geq r$, unchoke j in the next period
- If a peer j uploaded to i at a rate $< r$, choke j in the next period
- Every three time periods, optimistically unchoke a random peer from the neighborhood who is currently choked, and leave that peer unchoked for three time periods.

BitTorrent Optimistic Unchoking Algorithm

Tracker is a centralized entity that controls the traffic, tracks the connection between peers and their speed of upload, download etc.

Reference Client Protocol:

- Set a threshold r of uploading speed (typically the third maximum speed in the recent past)
- If a peer j uploaded to i at a rate $\geq r$, unchoke j in the next period
- If a peer j uploaded to i at a rate $< r$, choke j in the next period
- Every three time periods, optimistically unchoke a random peer from the neighborhood who is currently choked, and leave that peer unchoked for three time periods.

Forcing a repeated game by fragmenting the files

BitTorrent Optimistic Unchoking Algorithm

Tracker is a centralized entity that controls the traffic, tracks the connection between peers and their speed of upload, download etc.

Reference Client Protocol:

- Set a threshold r of uploading speed (typically the third maximum speed in the recent past)
- If a peer j uploaded to i at a rate $\geq r$, unchoke j in the next period
- If a peer j uploaded to i at a rate $< r$, choke j in the next period
- Every three time periods, optimistically unchoke a random peer from the neighborhood who is currently choked, and leave that peer unchoked for three time periods.

Forcing a repeated game by fragmenting the files
The leecher-seeder game is a repeated Prisoners' Dilemma

BitTorrent Optimistic Unchoking Algorithm

Tracker is a centralized entity that controls the traffic, tracks the connection between peers and their speed of upload, download etc.

Reference Client Protocol:

- Set a threshold r of uploading speed (typically the third maximum speed in the recent past)
- If a peer j uploaded to i at a rate $\geq r$, unchoke j in the next period
- If a peer j uploaded to i at a rate $< r$, choke j in the next period
- Every three time periods, optimistically unchoke a random peer from the neighborhood who is currently choked, and leave that peer unchoked for three time periods.

Forcing a repeated game by fragmenting the files
The leecher-seeder game is a repeated Prisoners' Dilemma
Strategy of the seeder is tit-for-tat

Illustration

Illustration

Strategic Behaviors

- How often to contact tracker?
- Which pieces to reveal?
- How many upload slots, which peers to unchoke, at what speed?
- What data to allow others to download?
- Possible goals: min upload, max download speed, some balance

Attacks on BitTorrent

- BitThief
- Strategic piece revealer
- BitTyrant

BitThief

- Goal: download files without uploading
- Keep asking for peers from tracker, grow neighborhood quickly
- Exploit the optimistic unchoking part
- Never upload!

BitThief

- Goal: download files without uploading
- Keep asking for peers from tracker, grow neighborhood quickly
- Exploit the optimistic unchoking part
- Never upload!
- Fix: modify the tracker (block same IP address within 30 minutes).

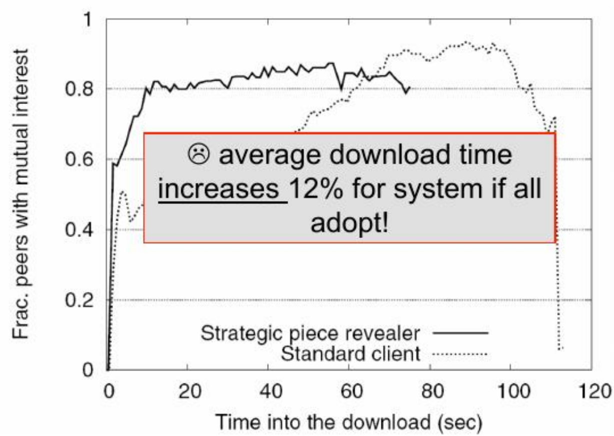
Ref: Locher et al., “Free Riding in BitTorrent is Cheap”, HotNets 2006

Strategic Piece Revealer

- Reference client: tell neighbors about new pieces, use “rarest-first” to request
- Manipulator strategy: reveal most common piece that reciprocating peer does not have!
- Try to protect a monopoly, keep others interested

Ref: Levin et al., “BitTorrent is an Auction: Analyzing and Improving BitTorrents Incentives”, SIGCOMM 2008

Strategic Piece Revealer



☹ average download time increases 12% for system if all adopt!

Summary

- P2P demonstrates importance of game-theory in computer systems
- Early systems were easily manipulated
- BitTorrent's innovation was to break files into pieces, enabling TitForTat.
- Still some vulnerabilities, but generally very successful example of incentive-based protocol design.