

# Advanced Computer Networks

## P2P Swarming

Jianping Pan  
Summer 2007

# Feedback on project proposals

- Give your project a name!
  - “TCP congestion control” (Hong-Yi)
  - XON: Xbox overlay network (Justyn and Dale)
  - “large amount of data transfer” (Leo and Ching-Chang)
  - “vehicular area networks” (Dandan)
  - Multimedia over multi-link (Ming)
  - “Directional antenna” (Emad)
  - “JXTA vs ...” (Ryan)
  - “Collaborate Whiteboard” (Andy and Chun-Hung)
  - “mobility detection in WSN” (Haoling)
- Individual feedback handed out in class
- Build project web at google and use as a log book

# Review: going P2P

- Client-server
  - server is well-known and serves all client requests
  - scalability issue
- Peer-to-peer
  - structured or unstructured
  - every peer is a (potential) server
    - search is a challenge
  - one request is still served by one peer
    - until the peer fails, then try to use another peer

# Napster and Gnutella

- Napster
  - centralized directory server
    - list uploading and query handling
  - peer-to-peer file download
- Gnutella
  - fully distributed
    - scoped flooding search
  - peer-to-peer file download
- Improving Gnutella
  - node hierarchy
  - non-flooding search

# More design choices

- If more than one peer can serve, why do they not serve the same request together?
- Benefit
  - more resilient to node dynamic
    - does not rely on any particular peer
  - fit better with the asymmetric access link
    - higher download bandwidth than upload
- Overhead
  - how to get served from multiple peers
    - work together constructively

# The BitTorrent approach

- Chop a file into small, fixed-size pieces
  - e.g., pieces (usually 256 KB each)
  - and then into blocks (usually 16 KB each)
- .torrent
  - meta information about the file
  - out-of-band retrieval
- Tracker
  - return a list of peers may have some pieces
- Seed and leecher/downloader
  - peers have the complete/incomplete file

# .bittorrent

- Tracker URL
- File info
  - name, length
- Piece info
  - length, hash
- Other info
  - date, comment, etc
- Bencoding
  - strings, integers, lists, directories
  - e.g., 4:spam, i3e, l4:spam4:eggse, d4:spaml1:a1:bee

# Tracker protocol

- HTTP GET request
  - info\_hash: to identify the file
  - peer\_id: of the requesting peer
  - client address and port: to respond to incoming requests
  - bytes uploaded, downloaded, left, etc
  - numwant: the number of peers in the response list
- Tracker response
  - failure reason, if any
  - contact interval
  - peer list and stat (seed and leecher, etc)
- Tracker-less mode (on Kademlia DHT)



# Tit-for-tat

- Download while upload: tit-for-tat
  - upload to whom from which download: trading pieces
  - prevent free-riding
    - fairness?
- Choking/unchoking
  - a limited number of uploads
    - default: 4
  - evaluate peers based on their recent download speed
    - 20-second average
  - upload to the peers with the fastest download speed
    - adjust every 10 seconds

# Optimistic unchoking

- Stuck with poor peers?
- Optimistic unchoking
  - upload to other peers as well
    - rotate every 30 seconds
  - hope to get better download
  - also help bootstrap other peers
- Seed's unchoking
  - seed does not download from other peers
  - try to equally distribute its upload to leechers
  - or upload to the one downloads fastest

# Peer wire protocol

- Messages over TCP
  - handshake
  - keep-alive
  - choke/unchoke
  - interested/not-interested
    - a block is downloaded if the client is interested and unchoked
    - a block is uploaded if the peer is interested and unchoked
  - have
    - advertise new pieces
  - request/piece
    - request blocks in a piece

# Piece selection

- Initially, a few random pieces
  - anything is better than nothing
  - easy to find at the beginning
- Then, rarest-first in neighborhood
  - become less dependent on seed
  - more interested by peers
- Finally, “end game” mode
  - look for missing pieces aggressively
  - send requests to all peers
  - cancel requests after last pieces are collected

# Student presentation

- Ryan Chen: BitTorrent
  - [QS04] Dongyu Qiu, R. Srikant. Modeling and Performance Analysis of Bit Torrent-Like Peer-to-Peer Networks. SIGCOMM 2004 [BitTorrent]

# More discussion

# This lecture

- BitTorrent
  - P2P swarming
  - protocol overview
  - performance analysis
- Explore further
  - measurement-based modeling
  - measurement-based performance analysis
  - BitTorrent extensions
    - <http://wiki.theory.org/BitTorrentSpecification>

# Next lecture

- June 6: Skype
  - [BS06] Salman A. Baset and Henning Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol", IEEE Infocom 2006. [Skype]
  
- Notice
  - reading list and schedule are online
  - presenter to be contacted one week in advance