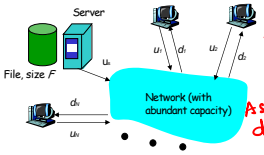
File Distribution: Server-Client vs P2P

Question: How much time to distribute file from one server to *N peers*?



$D_{P2P} \geq \max\left(\frac{F}{u_s}, \cancel{\frac{F}{d_{min}}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right)$

Assume $d_{min} > u_s > u_i$

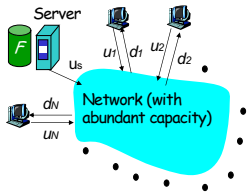
$\frac{NF}{u_s + \sum_{i=1}^{N} u_i} = \frac{NF}{u_s + Nu} = \frac{F}{\frac{u_s}{N} + u} \xrightarrow{\lim_{N\to\infty}} F/u$

$\lim_{N\to\infty} D_{P2P} \geq \max\left(\frac{F}{u_s}, \frac{F}{u}\right) = \frac{F}{u}$

71

File distribution time: client-server

- server sequentially sends N copies:
  - NF/u_s time
- client i takes F/d_i time to download



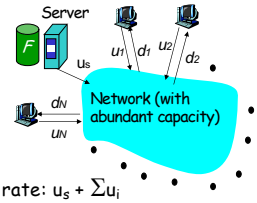
Time to distribute F to N clients using client/server approach $= d_{cs} >= \max\{ NF/u_s, F/\min_i(d_i) \}$

increases linearly in N (for large N)

72

File distribution time: P2P

- server must send one copy: F/u_s time
- client i takes F/d_i time to download
- NF bits must be downloaded (aggregate)
  - fastest possible upload rate: $u_s + \Sigma u_i$

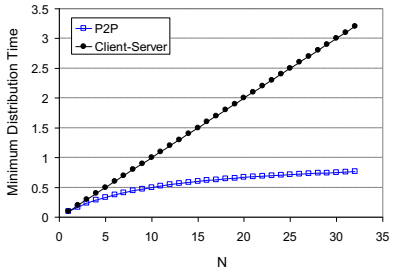


$d_{P2P} >= \max\{ F/u_s, F/\min_i(d_i), NF/(u_s + \Sigma u_i) \}$

73

Server-client vs. P2P: example

Client upload rate = u, F/u = 1 hour, u_s = 10u, $d_{min} \geq u_s$



74

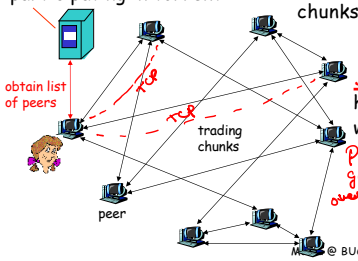
File distribution: BitTorrent

- P2P file distribution

file name → list of IP addresses of existing peers

tracker: tracks peers participating in torrent

torrent: group of peers exchanging chunks of a file

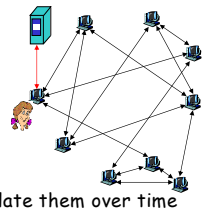


Seed: peer that has the whole file

P2P "overlay" graph (network) over TCP/IP "underlay"

75

BitTorrent (1)



- file divided into 256KB *chunks*
- peer joining torrent:
  - has no chunks, but will accumulate them over time
  - registers with tracker to get list of peers, connects to subset of peers ("neighbors")
- while downloading, peer uploads chunks to other peers
- peers may come and go
- once peer has entire file, it may (selfishly) leave or (altruistically) remain

76

BitTorrent (2)

Pulling Chunks (1) chunk/piece selection

- at any given time, different peers have different subsets of file chunks
- periodically, a peer (Alice) asks each neighbor for list of chunks that they have
- Alice sends requests for her missing chunks
  - rarest first policy

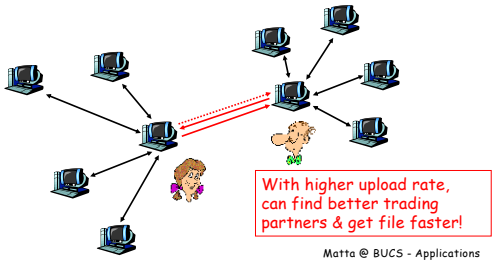
Sending Chunks: tit-for-tat (2) peer selection

- Alice sends chunks to four neighbors currently sending her chunks *at the highest rate*
  - re-evaluate top 4 every 10 secs
- every 30 secs: randomly select another peer, starts sending chunks
  - newly chosen peer may join top 4
  - "optimistically unchoke"

77

BitTorrent: Tit-for-tat

(1) Alice "optimistically unchokes" Bob
(2) Alice becomes one of Bob's top-four providers; Bob reciprocates
(3) Bob becomes one of Alice's top-four providers

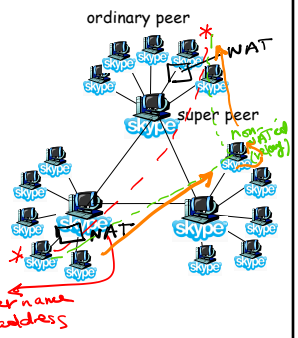


With higher upload rate, can find better trading partners & get file faster!

78

P2P Case study: Skype

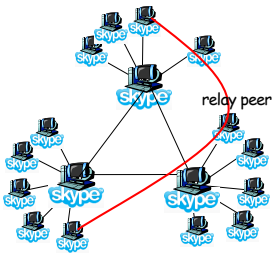
- proprietary application-layer protocol (inferred via reverse engineering)
- Hierarchical overlay of Skype peers
- Index maps usernames to IP addresses; distributed over super peers



79

Peers as relays

- Problem when both Alice and Bob are behind "NATs"
  - NAT prevents an outside peer from initiating a call to insider peer
- Solution:
  - Using Alice's and Bob's super peers, (non-NATed) Relay is chosen
  - Each peer initiates session with relay
  - Peers can now communicate through NATs via relay



80

Chapter 2: Summary

our study of network apps now complete!

- application architectures
  - client-server
  - P2P
  - hybrid
- application service requirements:
  - reliability, throughput, delay
- Internet transport service model
  - connection-oriented, reliable: TCP
  - unreliable, datagrams: UDP
- socket programming
- specific protocols:
  - HTTP
  - SMTP, POP, IMAP
  - DNS
  - P2P: BitTorrent, Skype, …

81

Chapter 2: Summary

Most importantly: learned about *protocols*

- typical request/reply message exchange:
  - client requests info or service
  - server responds with data, status code
- message formats:
  - headers: fields giving info about data
  - data: info being communicated

Important themes:

- persistent vs. non-persistent transport connections
- stateless vs. stateful
- caching
- reliable vs. unreliable msg transfer
- centralized vs. distributed
- Overlay vs. underlay

82