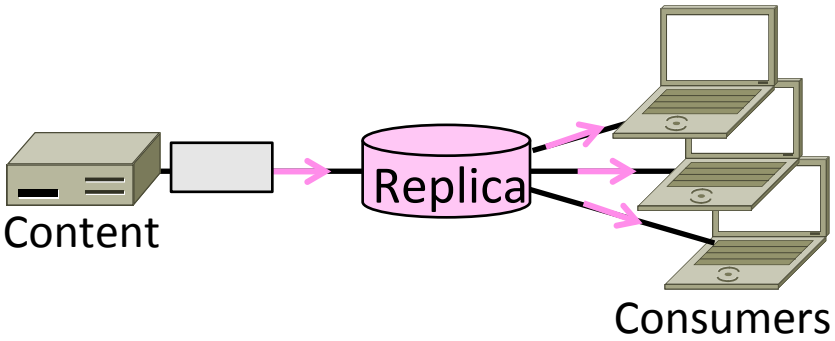
Topic

- CDNs (Content Delivery Networks)
  - Efficient distribution of popular content; faster delivery for clients

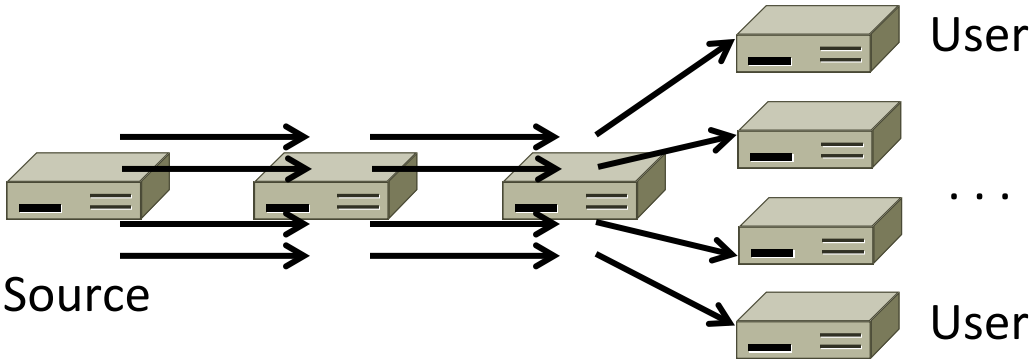


Context

- As the web took off in the 90s, traffic volumes grew and grew. This:
  1. Concentrated load on popular servers
  2. Led to congested networks and need to provision more bandwidth
  3. Gave a poor user experience
- Idea:
  - Place popular content near clients
  - Helps with all three issues above

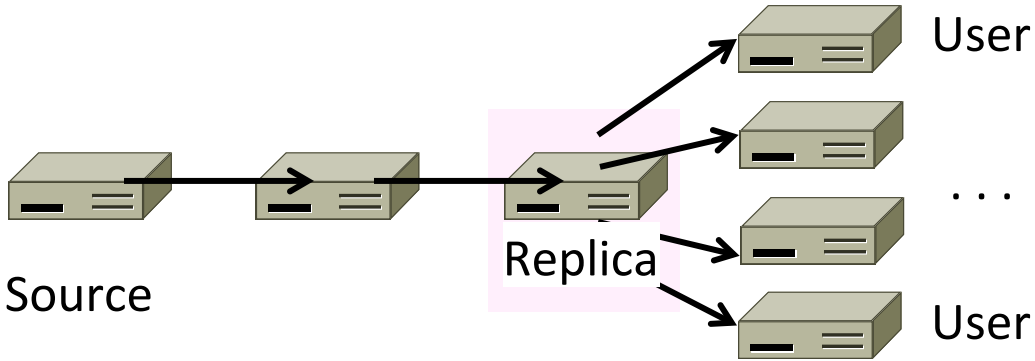
Before CDNs

- Sending content from the source to 4 users takes 4 x 3 = 12 “network hops” in the example



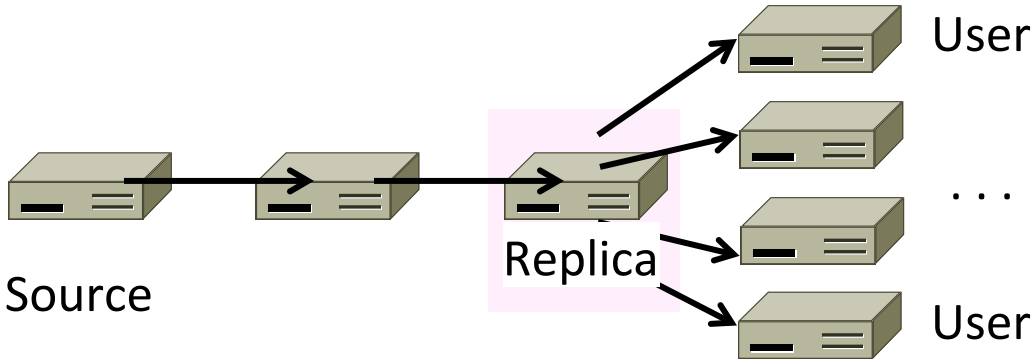
After CDNs

- Sending content via replicas takes only 4 + 2 = 6 “network hops”



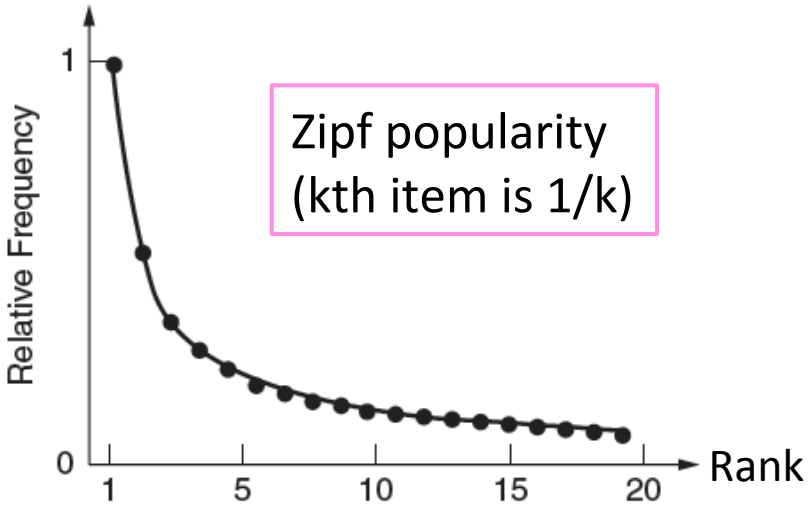
After CDNs (2)

- Benefits assuming popular content:
  - Reduces server, network load
  - Improves user experience (PLT)



Popularity of Content

- Zipf's Law: few popular items, many unpopular ones; both matter



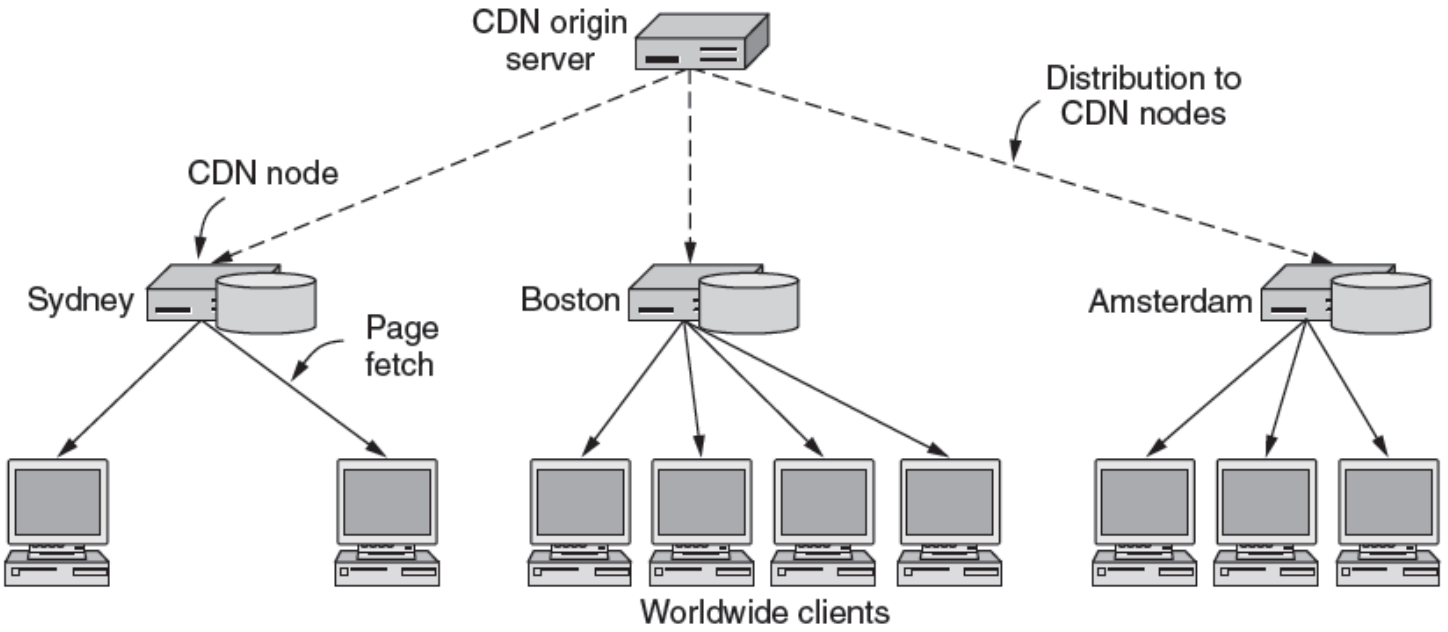
George Zipf (1902-1950)

Source: Wikipedia

How to place content near clients?

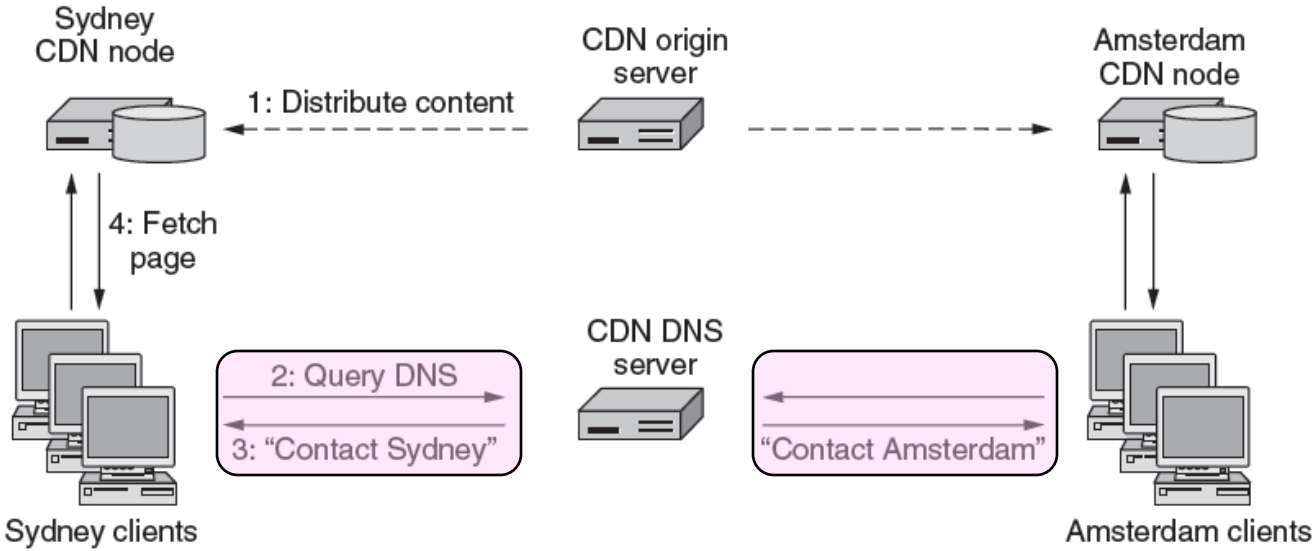
- Use browser and proxy caches
  - Helps, but limited to one client or clients in one organization
- Want to place replicas across the Internet for use by all nearby clients
  - Done by clever use of DNS

Content Delivery Network



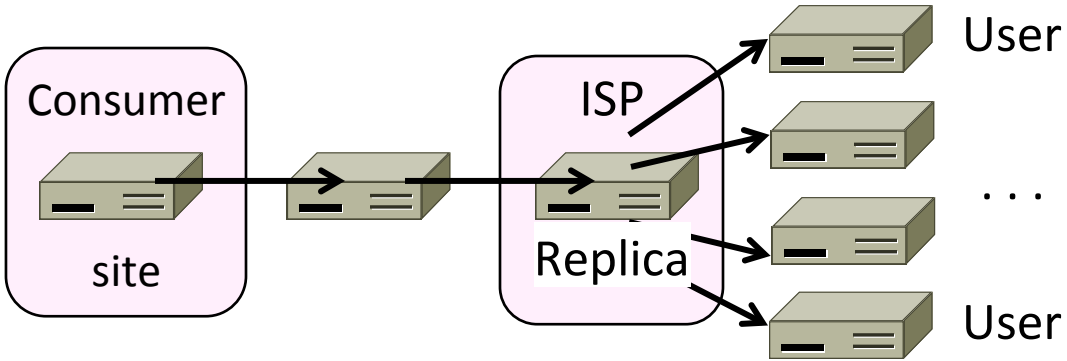
Content Delivery Network (2)

- DNS resolution of site gives different answers to clients
  - Tell each client the site is the nearest replica (map client IP)



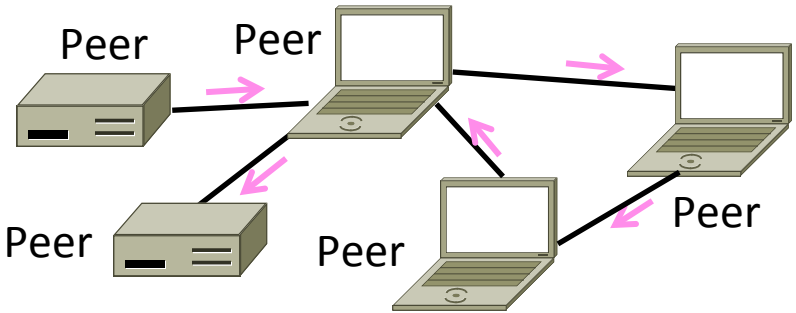
Business Model

- Clever model pioneered by Akamai
  - Placing site replica at an ISP is win-win
  - Improves site experience and reduces bandwidth usage of ISP



Topic

- Peer-to-peer content delivery
  - Runs without dedicated infrastructure
  - BitTorrent as an example



Context

- Delivery with client/server CDNs:
  - Efficient, scales up for popular content
  - Reliable, managed for good service
- … but some disadvantages too:
  - Need for dedicated infrastructure
  - Centralized control/oversight

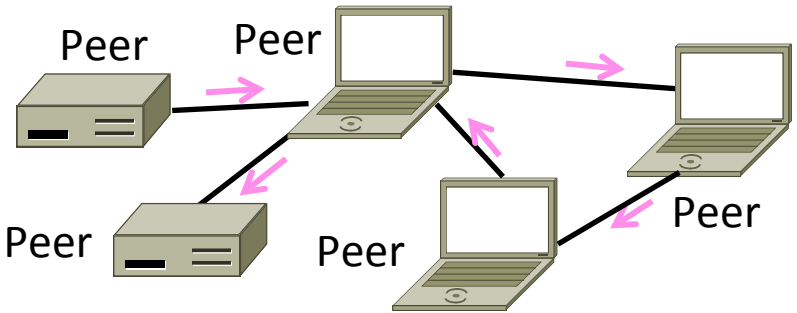
P2P (Peer-to-Peer)

- Goal is delivery *without* dedicated infrastructure or centralized control
  - Still efficient at scale, and reliable

- Key idea is to have participants (or peers) help themselves
  - Initially Napster '99 for music (gone)
  - Now BitTorrent '01 onwards (popular!)

P2P Challenges

- No servers on which to rely
  - Communication must be peer-to-peer and self-organizing, not client-server
  - Leads to several issues at scale …

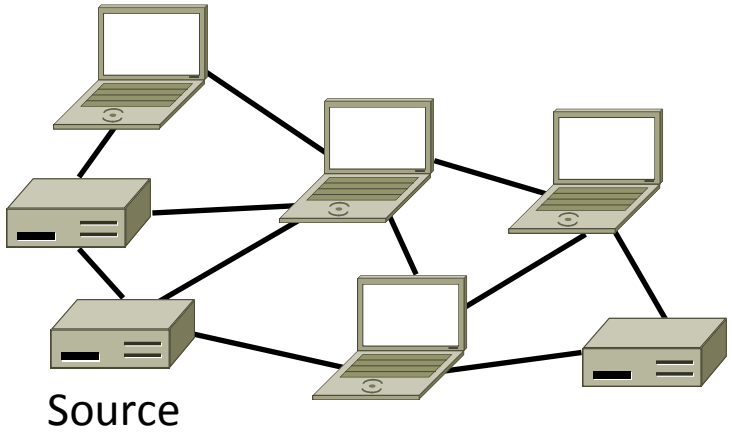


P2P Challenges (2)

1. Limited capabilities
   - How can one peer deliver content to all other peers?
2. Participation incentives
   - Why will peers help each other?
3. Decentralization
   - How will peers find content?

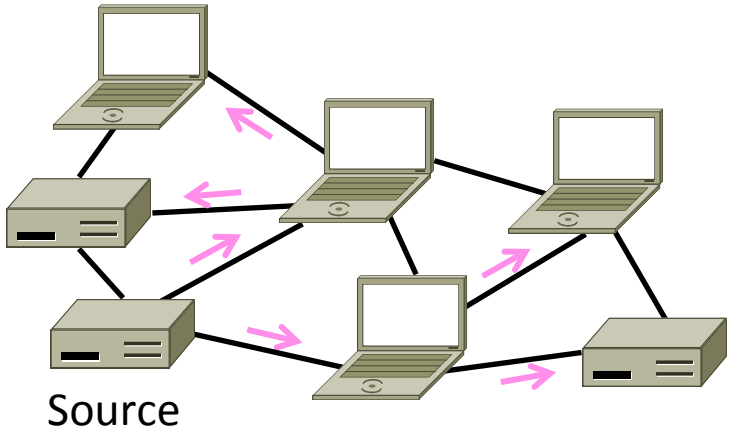
Overcoming Limited Capabilities

- Peer can send content to all other peers using a distribution tree
  - Typically done with replicas over time
  - Self-scaling capacity



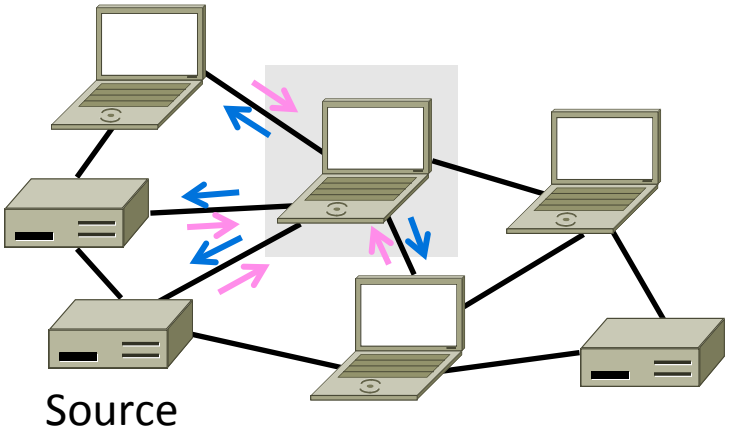
Overcoming Limited Capabilities (2)

- Peer can send content to all other peers using a distribution tree
  - Typically done with replicas over time
  - Self-scaling capacity



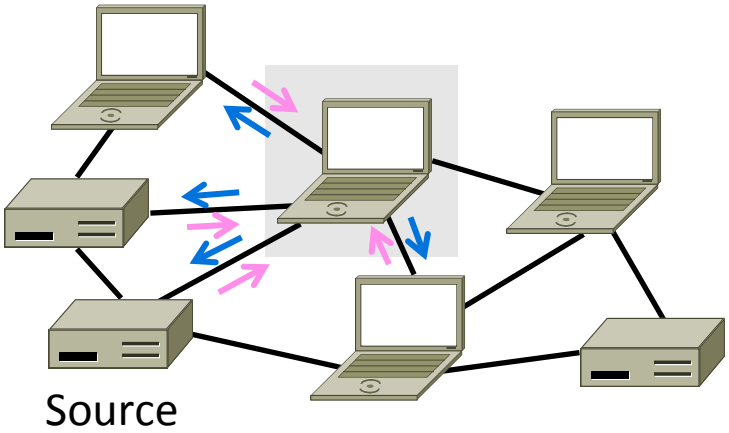
Providing Participation Incentives

- Peer play two roles:
  - Download (→) to help themselves, and upload (←) to help others



Providing Participation Incentives (2)

- Couple the two roles:
  - I'll upload for you if you upload for me
  - Encourages cooperation



Enabling Decentralization

- Peer must learn where to get content
  - Use DHTs (Distributed Hash Tables)
- DHTs are fully-decentralized, efficient algorithms for a distributed index
  - Index is spread across all peers
  - Index lists peers to contact for content
  - Any peer can lookup the index
  - Started as academic work in 2001

BitTorrent

- Main P2P system in use today
  - Developed by Cohen in ‘01
  - Very rapid growth, large transfers
  - Much of the Internet traffic today!
  - Used for legal and illegal content
- Delivers data using “torrents”:
  - Transfers files in pieces for parallelism
  - Notable for treatment of incentives
  - Tracker or decentralized index (DHT)

Bram Cohen (1975—)

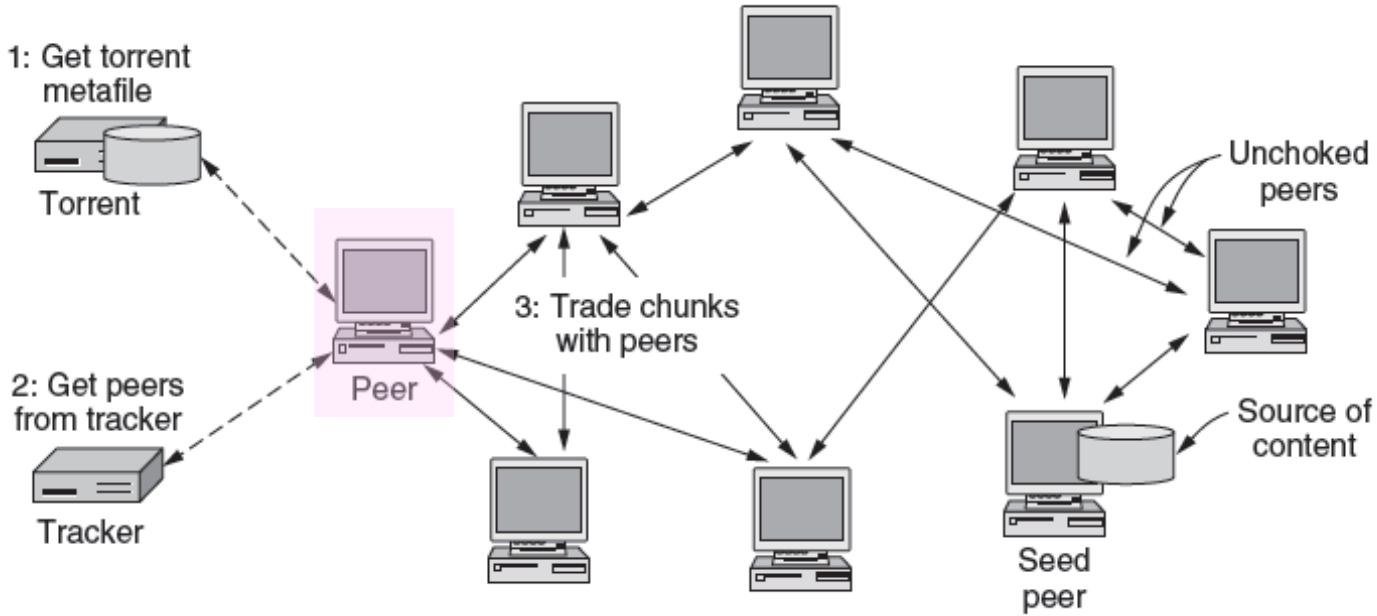
By Jacob Appelbaum, CC-BY-SA-2.0, from Wikimedia Commons

BitTorrent Protocol

- Steps to download a torrent:
  1. Start with torrent description
  2. Contact tracker to join and get list of peers (with at least seed peer)
  2. Or, use DHT index for peers
  3. Trade pieces with different peers
  4. Favor peers that upload to you rapidly; “choke” peers that don’t by slowing your upload to them

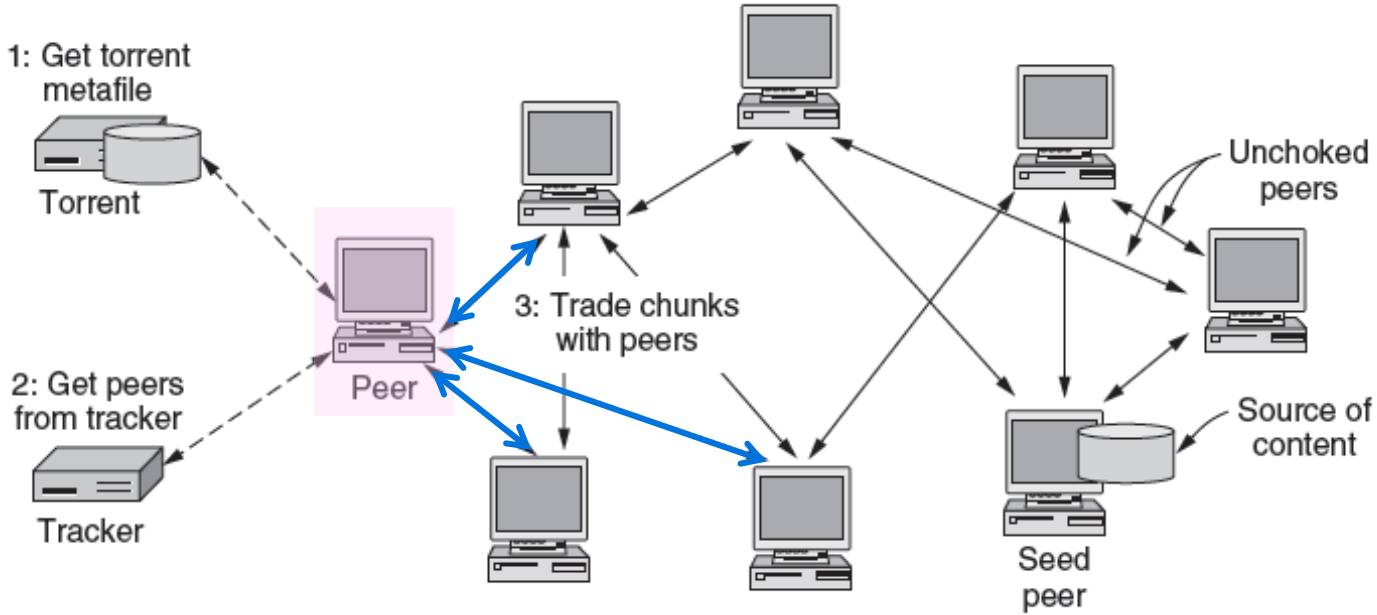
BitTorrent Protocol (2)

- All peers (except seed) retrieve torrent at the same time



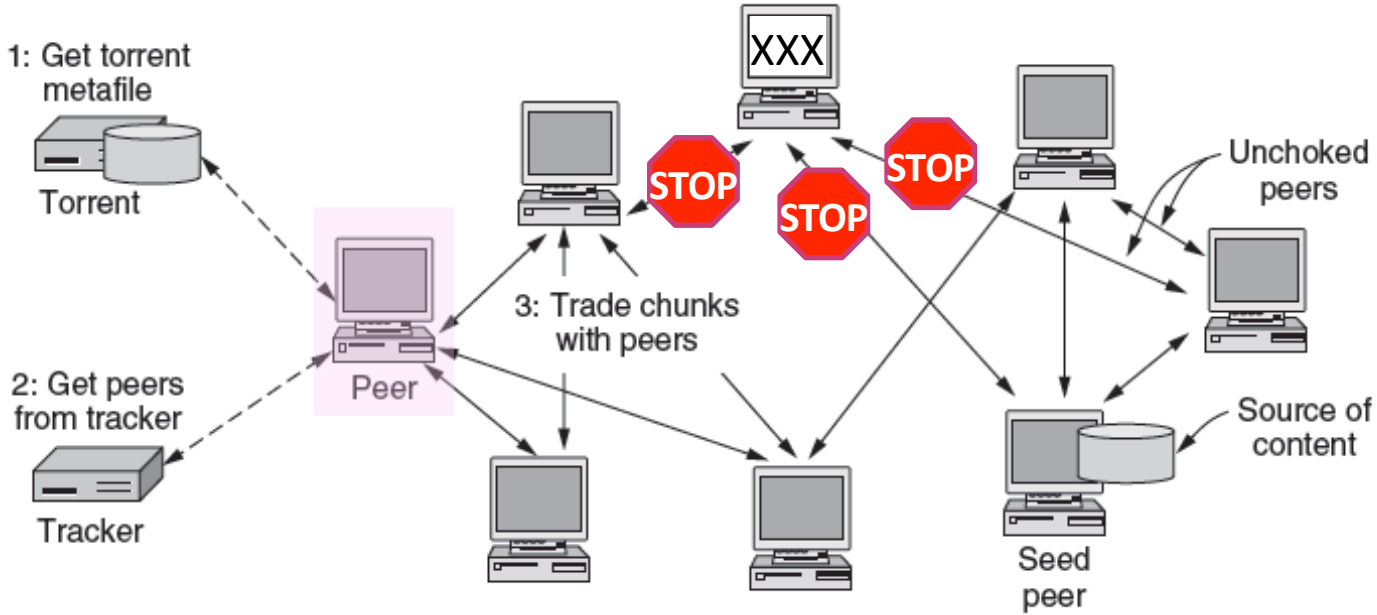
BitTorrent Protocol (3)

- Dividing file into pieces gives parallelism for speed



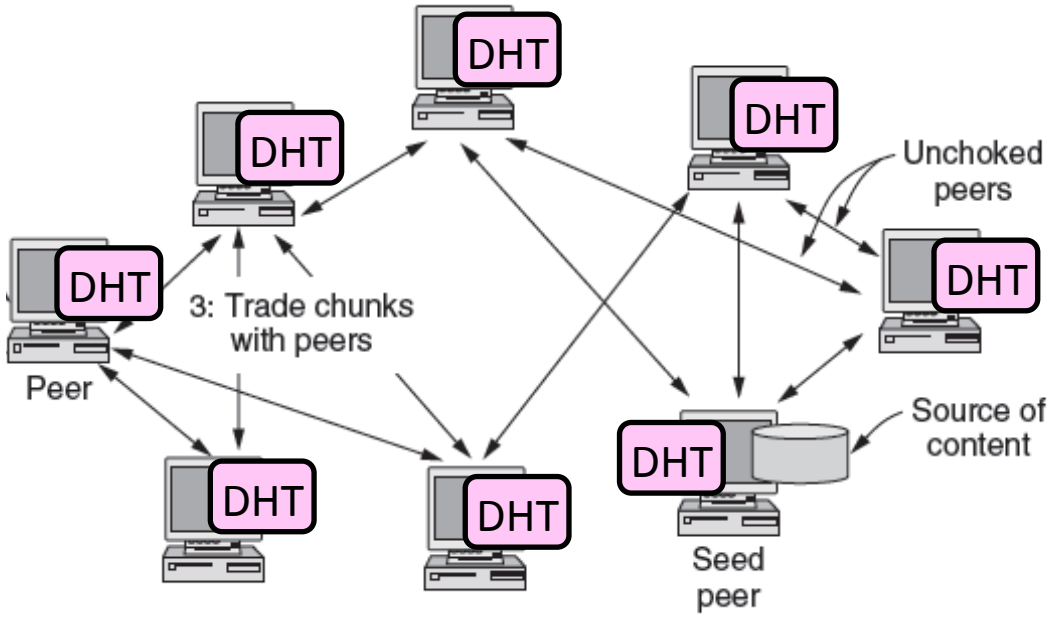
BitTorrent Protocol (4)

- Choking unhelpful peers encourages participation



BitTorrent Protocol (5)

- DHT index (spread over peers) is fully decentralized



P2P Outlook

- Alternative to CDN-style client-server content distribution
  - With potential advantages
- P2P and DHT technologies finding more widespread use over time
  - E.g., part of skype, Amazon
  - Expect hybrid systems in the future