Peer-to-Peer Networking

Daniel Zappala

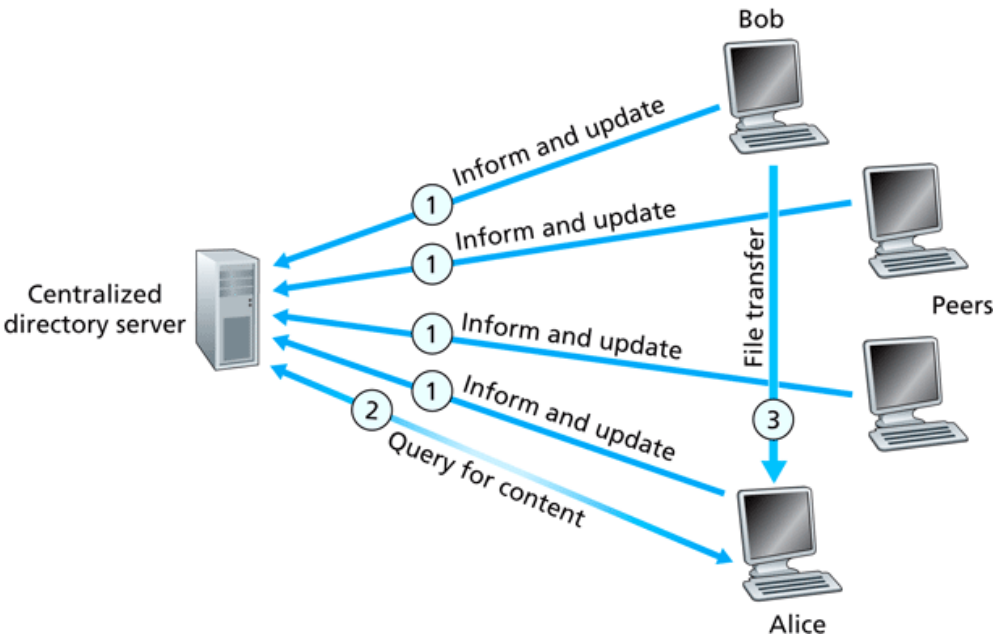
CS 460 Computer Networking
Brigham Young University

Definition

- hosts exchange data directly with each other
- hosts act as both clients and servers

Gnutella

Napster



- Napster stores a directory of music on your computer, so others can search it, download songs directly from you
- Like sharing cassette tapes or CDs or MP3s with your friends

Copyright Law

- copyright: owner has exclusive rights to reproduce, adapt, publicly distribute, perofrm, and display their work
  - direct infringement: copying part or all of a copyrighted work without authorization
  - vicarious liability: operator has (1) the right and ability to control users and (2) a direct financial benefit from allowing their acts of piracy.
  - contributory infringement: requires (1) knowledge of the infringing activity and (2) a material contribution – actual assistance or inducement – to the alleged piracy.

Fair Use

- use or copying of all or a portion of a copyrighted work without permission of the owner, e.g. for criticism, comment, news reporting, teaching, scholarship, or research
- courts consider:
  - purpose and character of use (commercial vs non profit)
  - nature of work
  - amount and substantiality of portion used (including size and quality)
  - the effect of use on market for or value of copyrighted work
- ▸ A Fair(y) Use Tale

Napster in Court

- Napster claims they are not infringing copyright because they are not storing any songs
- shutdown by court injunction because case against them was likely to succeed
  - Napster users likely guilty of *direct* copyright infringement - copying of a work by another
  - Napster likely to be guilty of *contributory* infringement because they learned of infringement and failed to purge the materials from its system
  - Napster likely to be guilty of *vicarious* infringement because they supervised or controlled the party engaging in infringing activity and had a financial interest in the activities
- see Wikipedia for background information

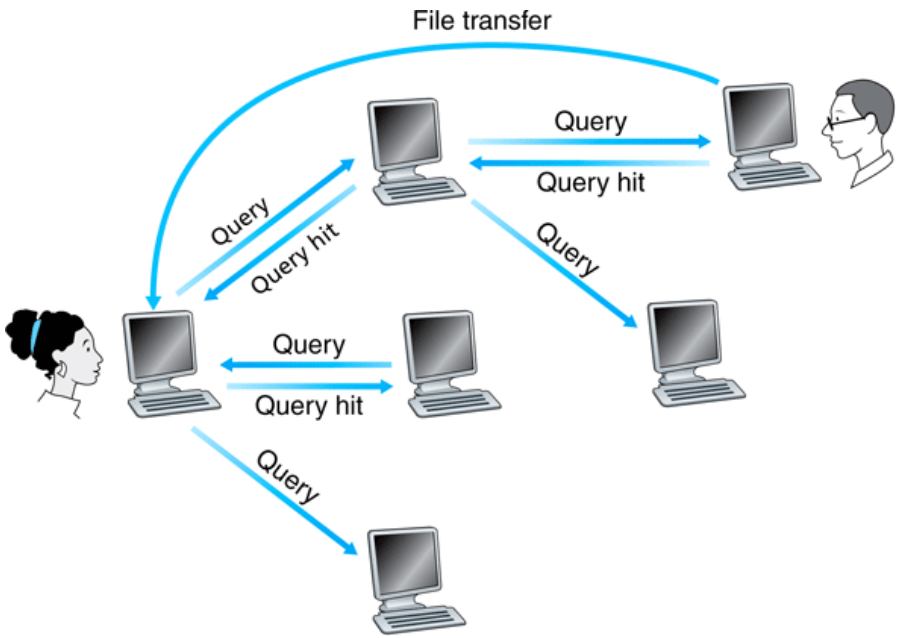
Promotional Power of Free Music

- record companies have claimed that free downloads suppress sales
- some proof of the opposite effect
  - April 2000: tracks from Radiohead's *Kid A* album on Napster three months before CD release
  - millions of downloads by the time the record is released
  - number one spot on the charts in debut week, had never been in the top 20 before
  - beat many other heavily marketed artists
- **this example doesn't excuse piracy, but it does indicate that file sharing can provide a marketing opportunity for new bands**

Gnutella – version 0.4

- can we share music illegally and not get caught?
- fully distributed, peer-to-peer system
- bootstrapping
  - first time: connect to a peer you heard about outside the system
  - for example, in a chat room
  - keep a cache of all peers discovered and use for bootstrapping next time
- peer discovery
  - try to always be connected to a fixed number of peers (TCP)
  - send a Ping message to existing neighbors, which is flooded to their neighbors
  - other peers respond to Ping with one or more Pong messages, containing IP address, port number, number of files sharing, number of KB sharing

Gnutella – version 0.4

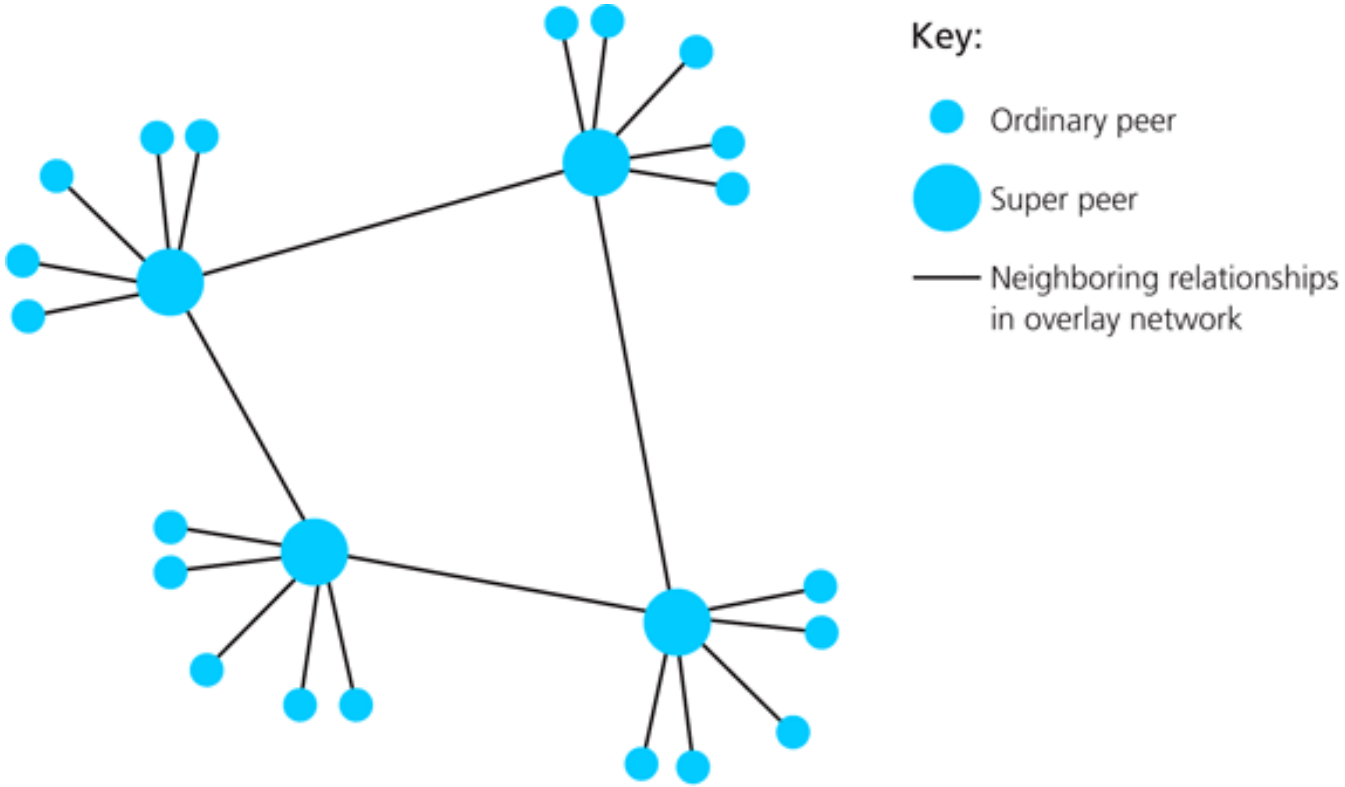


- queries
  - send a query to your neighbors
  - neighbors flood query, limited by a TTL
  - includes minimum speed in kb/s for responding peers, search criteria
- query hit
  - provide IP address, port, number of hits, speed, result set (file name, size)
  - sent along reverse path

Gnutella – version 0.4

- download songs directly from peer
- problems
  - no explicit rate limit on ping frequency or query frequency - quickly leads to overload
  - slow peers can hinder faster peers

Gnutella – version 0.6



- **use hierarchy to scale**
  - super peer: peers with high bandwidth
  - ordinary peer: peers with low bandwidth
- super peers cache names of content held by children
- queries sent among only the super peers

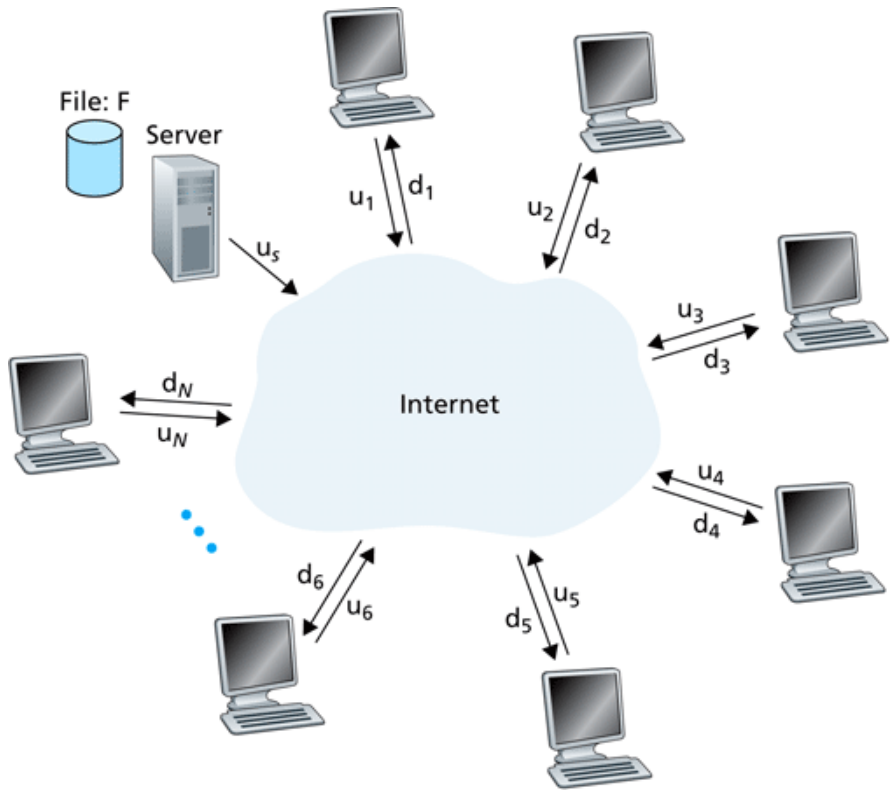
BitTorrent

Motivation

- how can an ordinary person, with limited money and bandwidth, serve content to a worldwide audience?
- web servers are limited in their scalability
  - the more clients that need to be served, the slower they access the content
  - eventually the wait becomes so long, TCP connections time out
- solutions
  - Content Delivery Network: spreads the load among a set of servers, but it is expensive
  - Peer-to-Peer File Distribution: spreads the load among a set of peers, inexpensive, must rely on the good will of others

Modeling File Download

- server upload rate: u_s
- peer upload rate: u_i
- peer download rate: d_i
- file size (bits): F
- total number of peers: N
- assume plentiful bandwidth in the Internet core



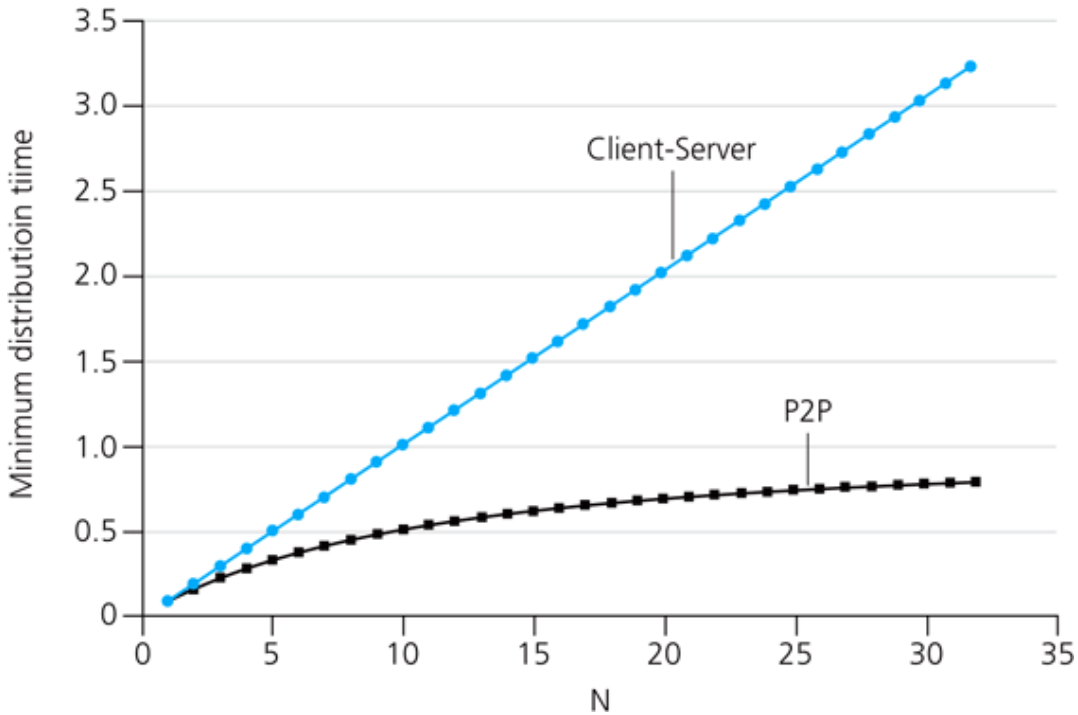
Client-Server Distribution Time

- min download time
  - $\frac{NF}{u_s}$ when constrained by server bandwidth
  - $\frac{F}{d_{min}}$ when constrained by slowest peer,
    $d_{min} = min(d_1, d_2, .., d_N)$
- $D_{CS} \geq max(\frac{NF}{u_s}, \frac{F}{d_{min}})$

Peer-to-Peer Distribution Time

- minimum download time
  - $\frac{F}{u_s}$ when constrained by server bandwidth (must deliver the file at least once)
  - $\frac{F}{d_{min}}$, when constrained by the slowest peer
  - $\frac{NF}{u_s + \sum_{i=1}^{N} u_i}$, when constrained by the overall upload rate
- $D_{P2P} \geq max(\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i})$

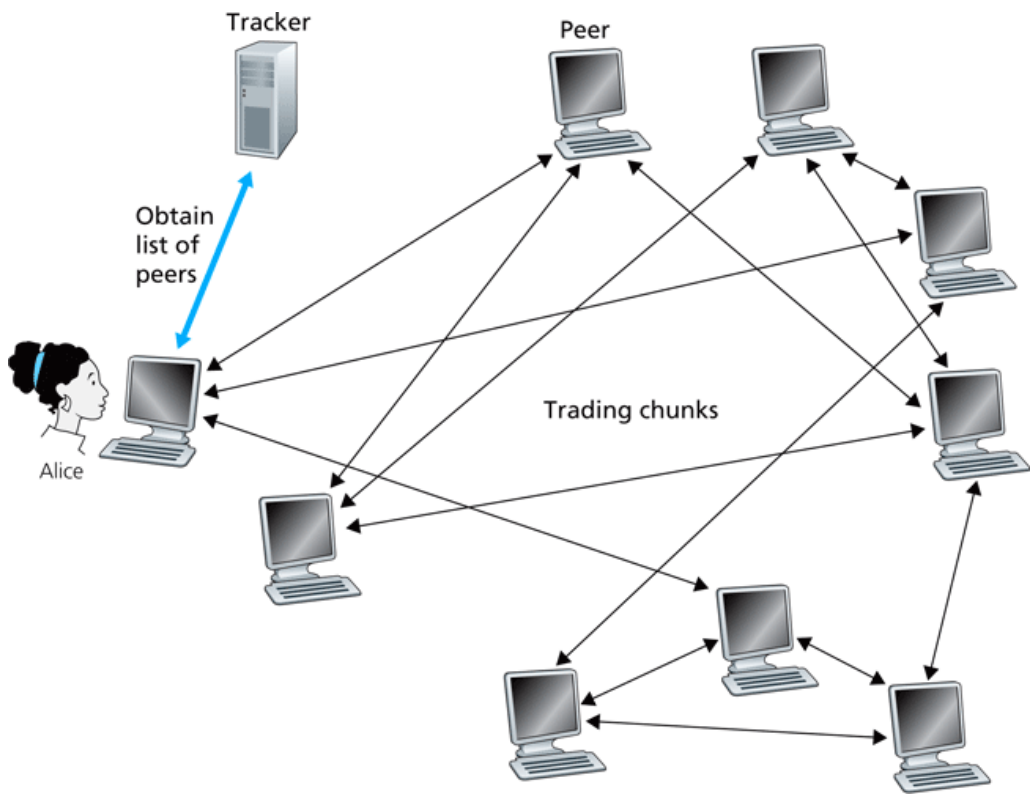
Comparison



- $F/u_i = 1$ hour, $u_s = 10u_i$, $d_min \geq u_s$
- peer-to-peer download is self-scaling: the more peers that download, the more bandwidth is available for upload

Basic Mechanisms

1. download a .torrent file from a web server
2. contact the listed *tracker* for a list of peers
3. refresh peers as needed
4. check with each peer to determine which blocks they have
5. parallel download, j connections, rarest block first



Incentives

- problem: freeloaders
  - people who try to download without uploading
  - breaks the self-scaling behavior of peer-to-peer distribution
- tit-for-tat
  - serve content to k connections at a time
  - serve the connections that give you the best download rate
  - periodically serve content to a random connection to see if it can do better than a current connection
  - deny content to all others

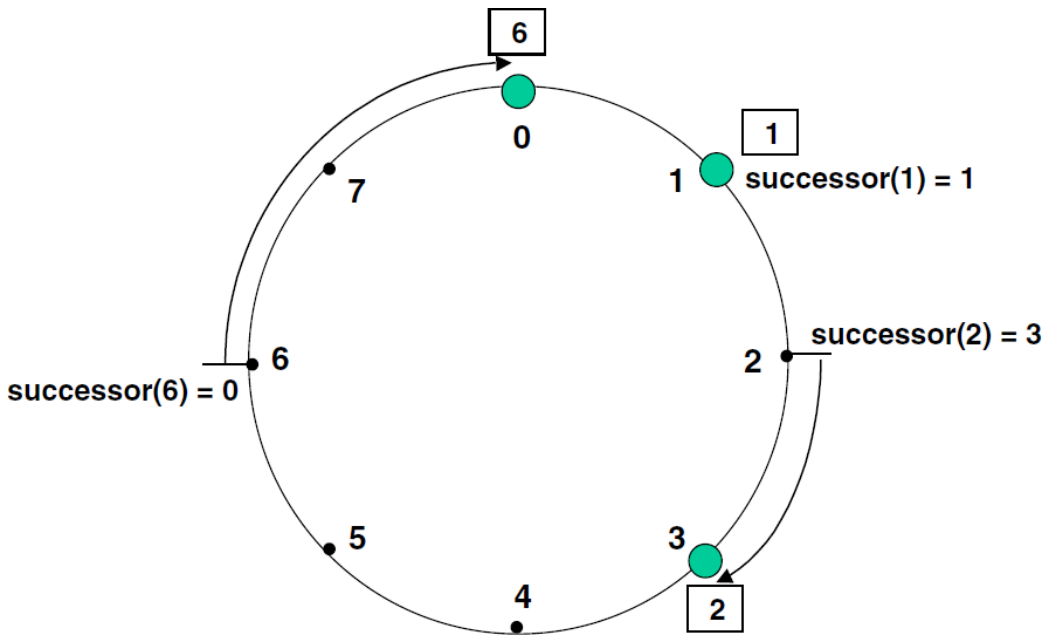
Distributed Hash Tables

Re-Thinking Peer-to-Peer Search

- how do you build a general, Internet-wide lookup service for content?
- Gnutella
  - everyone has files to share
  - connect them and allow searches
  - hard to find unpopular files
  - you don't know who has what
- Distributed Hash Table
  - everyone has storage to share
  - connect them in a structured network
  - map each file to a particular node
  - easy to find any file if you know its name

Distributed Hash Table

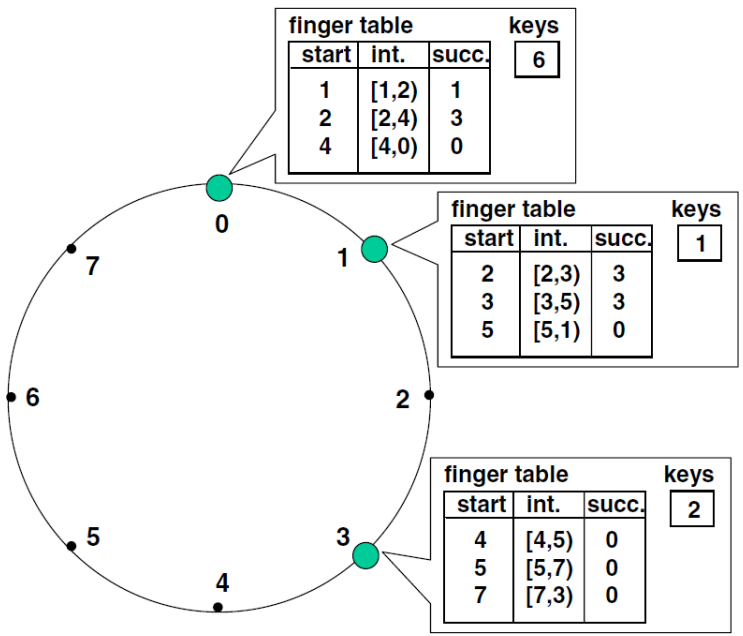
- form a ring of peers
  - use SHA-1 to hash node's IP address into an m-bit identifier
  - store pointer to next node on ring
- store keys on peers
  - hash key into an m-bit identifier
  - store key on $successor(k)$: first node whose identifier is equal to or greater than k
- goal: navigate from a peer to the node with key k



Optimizing Key Location

- finger table entries
  - each node maintain at most m finger table entries
  - the i^{th} entry at node n has the identity of the first node, s that succeeds n by at least 2^{i-1}, where $1 \leq i \leq m$ and math is modulo 2^m

- node 0 finger table:
  - one-hop away: key 1, successor 1
  - two hops away: key 2, successor 3
  - four hops away: key 4, successor 0
- finger tables mean a node can jump at least halfway to its target



Complications

- need algorithms for
  - joining
    - link a node into the structure
    - get the content it should store
  - leaving
    - fix the structure
    - move your content onto other nodes
  - failures
    - want high data availability
    - replicate data on other nodes
  - caching to improve performance

Properties

- consistent hashing
  - with high probability, hash function balances load: all nodes receive roughly the same number of keys
  - with high probability, when an N^{th} node joins or leaves the network only an $O(1/N)$ fraction of keys are moved to a different location
- in an N-node network, each node maintains information about only $O(logN)nodes$, but even if this information is inaccurate lookup will always succeed
- a lookup requires $O(logN)$ messages and the path length is $O(logN)$
- joining or leaving the network requires at most $O(log^2 N)$ messages

Uses of a DHT

- distributed index service: take common keyword searches for Gnutella and map to the set of machines holding matching files
- BitTorrent: uses a DHT To keep track of which peers are in the system and what they have
- Coral, Codeen: cache of web pages that is accessed when origin web server is overloaded or unavailable
- OpenDHT: public DHT service
- PAST: distributed file system layered on top of the Pastry DHT