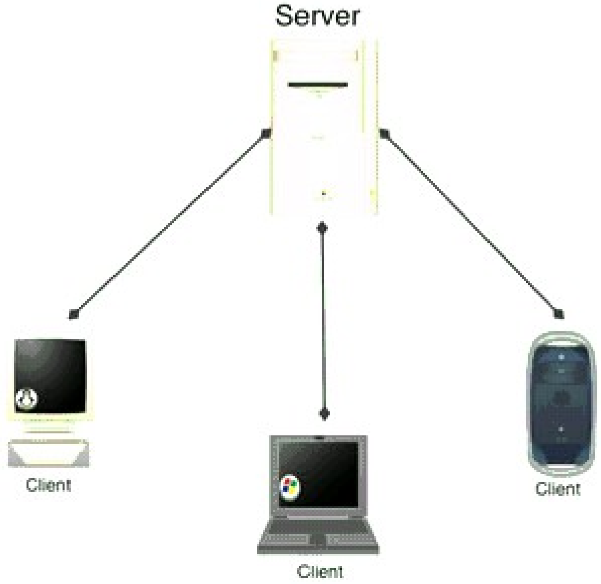
Advanced Computer Networks

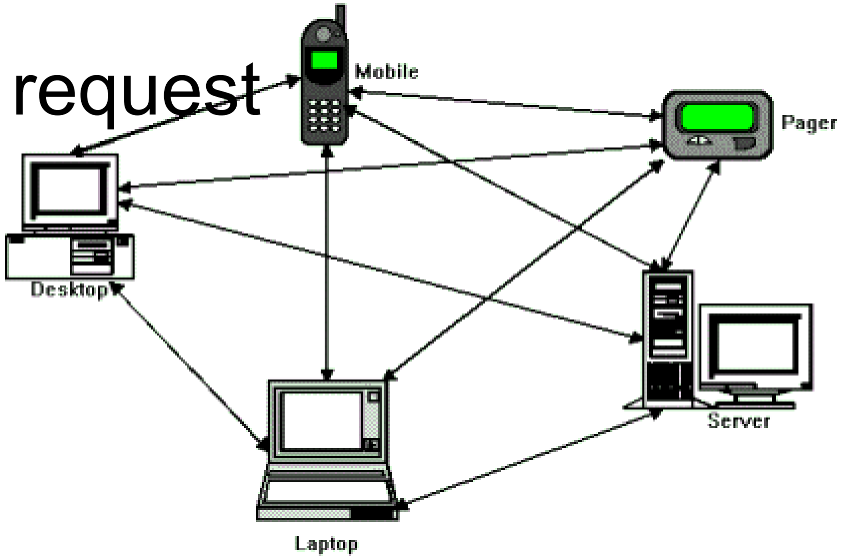
P2P Systems: Beyond DHT

Jianping Pan
Spring 2017

C/S vs P2P



- Client-server
  - Ĥ server is well-known
  - Ĥ server may become a bottleneck
- Peer-to-peer
  - Ĥ everyone is a (potential) server
    - intrinsically scalable
  - Ĥ how to match a “server” for a request
    - e.g., locate a file by its name
  - Ĥ search is a challenge
    - put() and get()



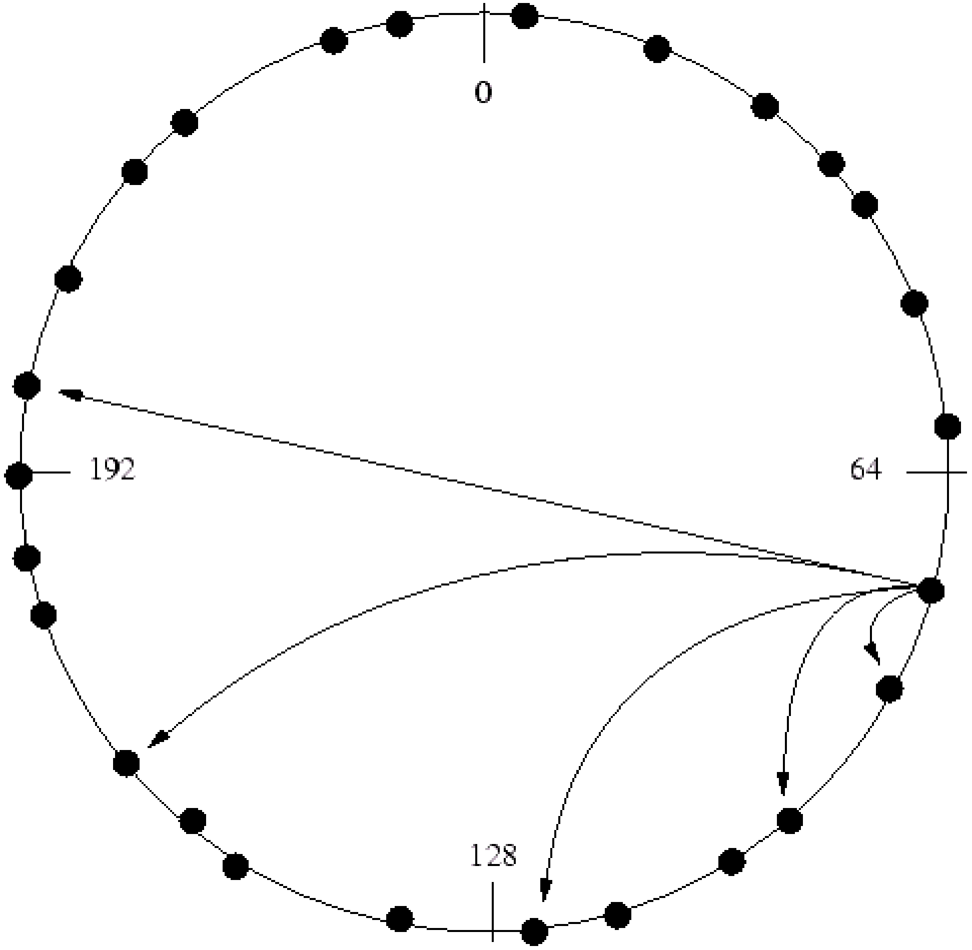
Review: structured P2P

- Structured P2P networks
  - Ĥ Chord (MIT)
  - Ĥ CAN (Berkeley, ICSI)
  - Ĥ *and more: Pastry (Microsoft, Rice), Tapestry (Berkeley), Kademlia (NYU)*
    - *included in the midterm reading materials*
- Reading groups formed on connex
  - – R1 announced on connex too!
  - – see reading guideline and template
- Unstructured P2P networks

http://www.cs.uvic.ca/~pan/csc466/reading.txt
http://www.cs.uvic.ca/~pan/csc466/rs.txt (or rs.tex for latex)

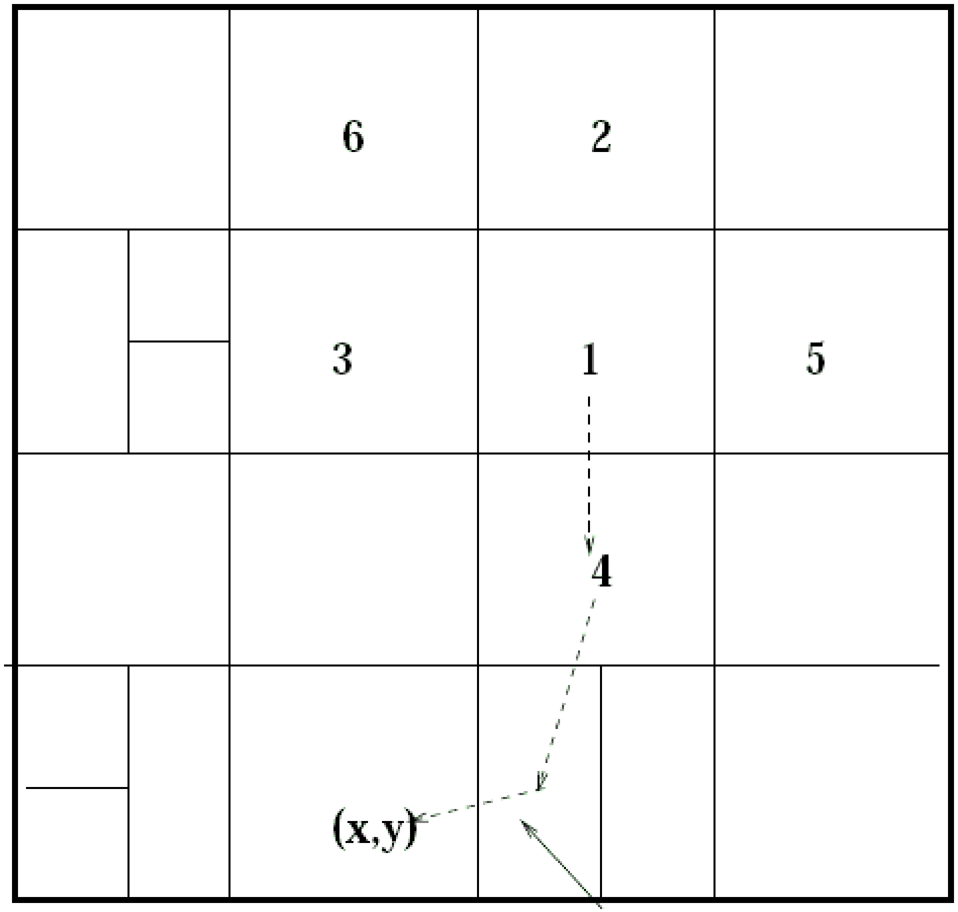
Chord

- Virtual circular space
  - Ή consistent hashing
  - Ή node ID, object key
- With successor list
  - Ή O(n) hops
  - Ή O(1) entry
- With “finger” table
  - Ή O(log n) hops
  - Ή O(log n) entries



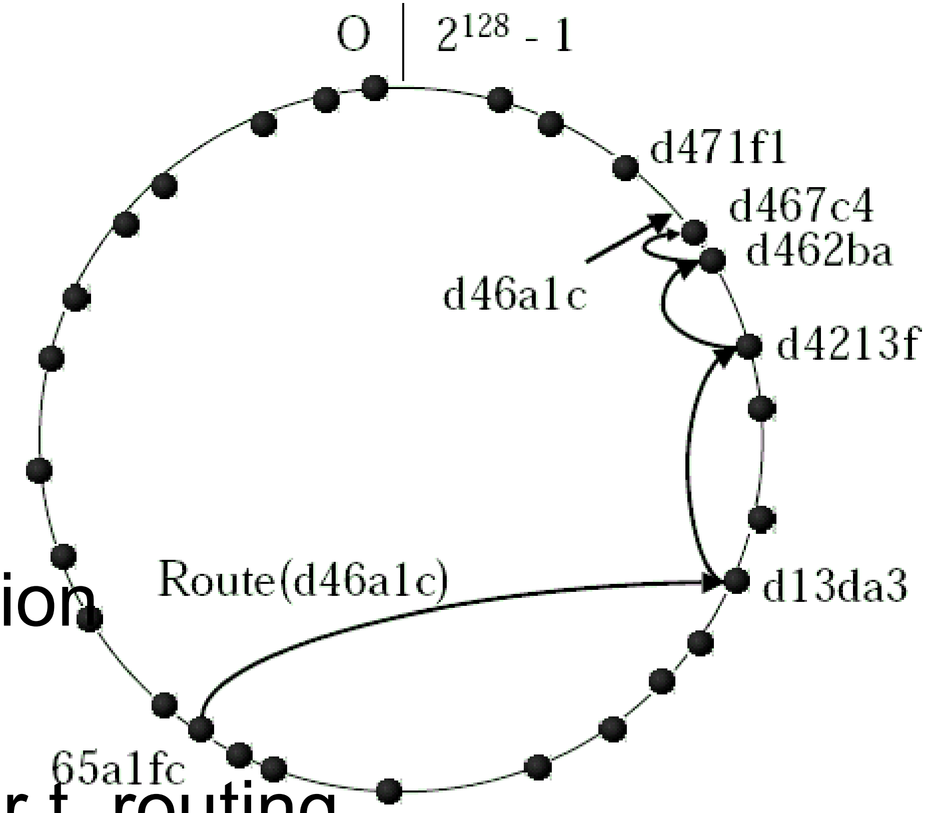
Content Addressable Network

- Virtual d-torus space
  - H́ consistent hashing
  - H́ e.g., 2-d: h_x(key), h_y(key)
- Routing performance
  - H́ O(d $n^{1/d}$) hops
  - H́ O(d) entries
    - neighborhood routing



Pastry

- Virtual circular space
  - H́ consistent hashing
- Routing performance
  - H́ O(log $_{2^b}$ n) hops
  - H́ leaf: L/2 closest each direction
    - tree-like routing
  - H́ neighborbood: M closest w.r.t. routing
    - maintain locality; later this design is dropped
  - H́ routing table: O((2^b-1)log $_{2^b}$ n)
    - prefix-matching

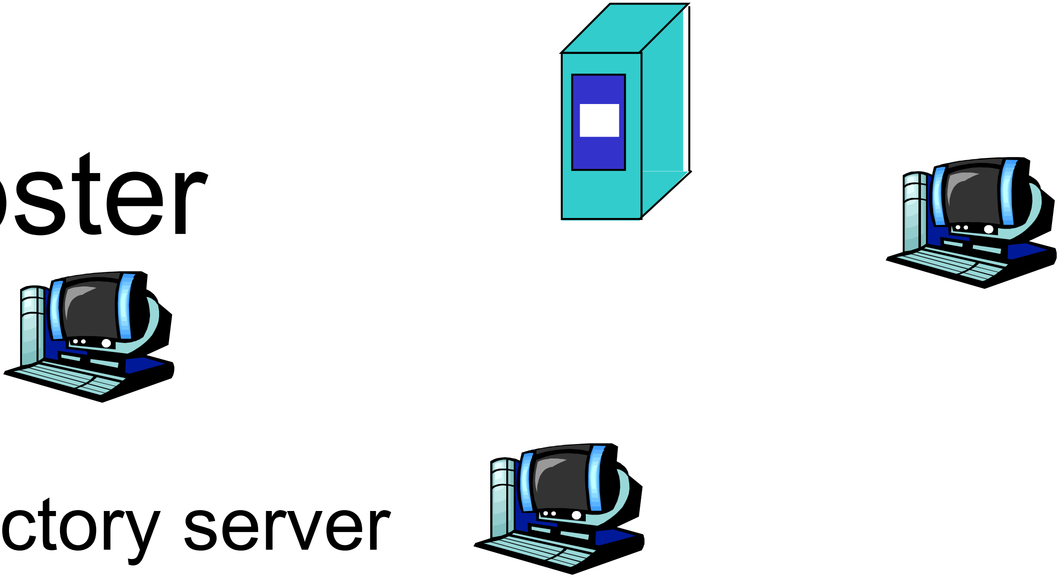


Today: unstructured P2P

- Structured P2P networks: applications
  - Chord: CFS (coop FS)
  - Pastry: PAST (file system), SCRIBE (pub/sub)
  - OpenDHT: DHT as a service over Planet-lab
- Unstructured P2P networks
  - Napster: one of the fastest growing Internet apps
  - Gnutella: first fully distributed one
  - BitTorrent: still most popular now?
  - Skype: P2P VoIP *

* after Microsoft acquisition?

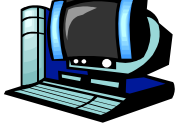
Napster

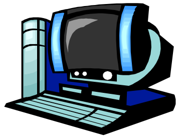


- Napster: C/S + P2P
  - connect to Napster directory server
  - upload a list of file information
  - send keyword queries to the server
  - receive a list of “hosts” from the server
  - choose the “best” host (with ping)
  - send the request to the host
  - receive the file from the host, or try the next host
- Discussion: critics on Napster
  - from the viewpoint of network protocol

Explore further: http://david.weekly.org/code/napster.php3

Gnutella





- Gnutella: P2P + flooding
  - no centralized server
    - even for string search
  - send keyword queries to up to 7 neighbors
    - if a neighbor can answer, reverse the query path
    - if not, the neighbor sends queries to its neighbors
    - maximum hops: e.g., 7
  - controlled flooding
    - no same queries sent by the same node twice
    - the same queries can be received more than once
- Q: pros and cons vs Napster?

Bootstrap

- Need to know at least one “working” node
  - initially, embedded in software
  - host cache from working nodes
    - the dominant approach
  - other means: e.g., manual configuration
- Connect to known nodes
  - Based on TCP/IP, ASCII strings
  - GNUTELLA CONNECT/0.4\n\n
  - GNUTELLA OK\n\n
  - only a small set of directly connected nodes

Protocol descriptors

- Descriptor ID
  - Ή global unique ID (GUID)
- Payload descriptor
- TTL
  - Ή at each hop: TTL--
  - Ή when TTL == 0, drop
- Hops
  - Ή TTL(0) = TTL + Hops

Descriptor ID	Payload Descriptor	TTL	Hops	Payload Length
0 15	16	17	18	19 22

Explore further: http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf

PING-PONG

- PING (0x00)
    - H́ probe for other nodes
    - H́ null payload
- PONG (0x01)
    - H́ response to PING
        - it possible to have multiple PONGs for one PING
    - H́ reverse PING path
    - H́ contain the IP address of the responder
    - H́ and the number/amount of files to be shared
- PING-PONG traffic should be minimized

QUERY-HIT

- QUERY (0x80)
  - minimum speed in Kbps
  - search string
- QUERYHIT (0x81)
  - reverse QUERY path
  - contain: number of hits
  - port number and IP address of the “host”
  - “supported” speed in Kbps
  - search results: file index, file size, file name
  - and the GUID of the responder

File retrieval

- File retrieval
  - H́ over HTTP
  - H́ request from the QUERY node to QUERYHIT node
    - fail if QUERYHIT node is behind firewall/NAT
- PUSH (0x40)
  - H́ contain: the GUID of the QUERYHIT node
  - H́ file index at the QUERYHIT node
  - H́ IP address at the QUERY node
  - H́ and port number at the QUERY node
  - H́ Q: if QUERY is also behind firewall/NAT?

Discussion

- Critics on Gnutella/0.4
  - hints
    - node structure
    - message handling
    - load balance
    - bootstrap process

Improving Gnutella

- Node structure
  - from flat to hierarchical
- GNUTELLA/0.6
  - more HTTP/1.0 like
- Ultra-peer: handle message forwarding
  - qualification: not behind firewall/NAT
  - sufficient computing and storage resources
  - and reliable network condition
  - leaf nodes only connects to ultra-peer nodes
- Also in KaZaA: super-node

[CRBLS03] Yatin Chawathe, S. Ratnasamy, Lee Breslau, Nick Lanham, Scott Shenker, "Making Gnutella-like P2P Systems Scalable", Sigcomm 2003. Gnutella

GNUTELLA/0.6

- Ultra-leaf node hierarchy
- Other features
  - H́ GWebCache
    - working nodes discovery
  - H́ cache PONG, QUERYHIT
  - H́ flow control, direct response to ultra-peer
    - limit/reduce the amount of message handling
  - H́ PUSH through ultra-peer
  - H́ reject with X-Try
    - be more friendly
  - H́ BYE (0x02)

Explore Further: http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

Non-flooding search

- Random walk
  - Ĥ unbiased random walk
    - Q: pros and cons?
  - Ĥ biased random walk
    - toward better connected nodes
    - which node is “better”?
- Network-aware search
  - Ĥ network-aware cluster

This lecture

- Gnutella
  - full distributed, flooding based
  - ways to improve Gnutella
    - Gia and why it is better
- Explore further
  - in “8. REFERENCES”
    - papers cited by this one
  - in scholar.google.com
    - papers citing this paper
  - “Should we build Gnutella on a structured overlay?”

Next lectures

- BitTorrent
  - [QS04] Dongyu Qiu, R. Srikant. Modeling and Performance Analysis of Bit Torrent-Like Peer-to-Peer Networks. SIGCOMM 2004 [BitTorrent]
- 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 groups are now on crosscourse
  - reading summary guideline and templates

http://www.cs.uvic.ca/~pan/csc466/reading.txt
http://www.cs.uvic.ca/~pan/csc466/rs.txt (or rs.tex for latex)