Peer-to-Peer Networks

Outline

Survey

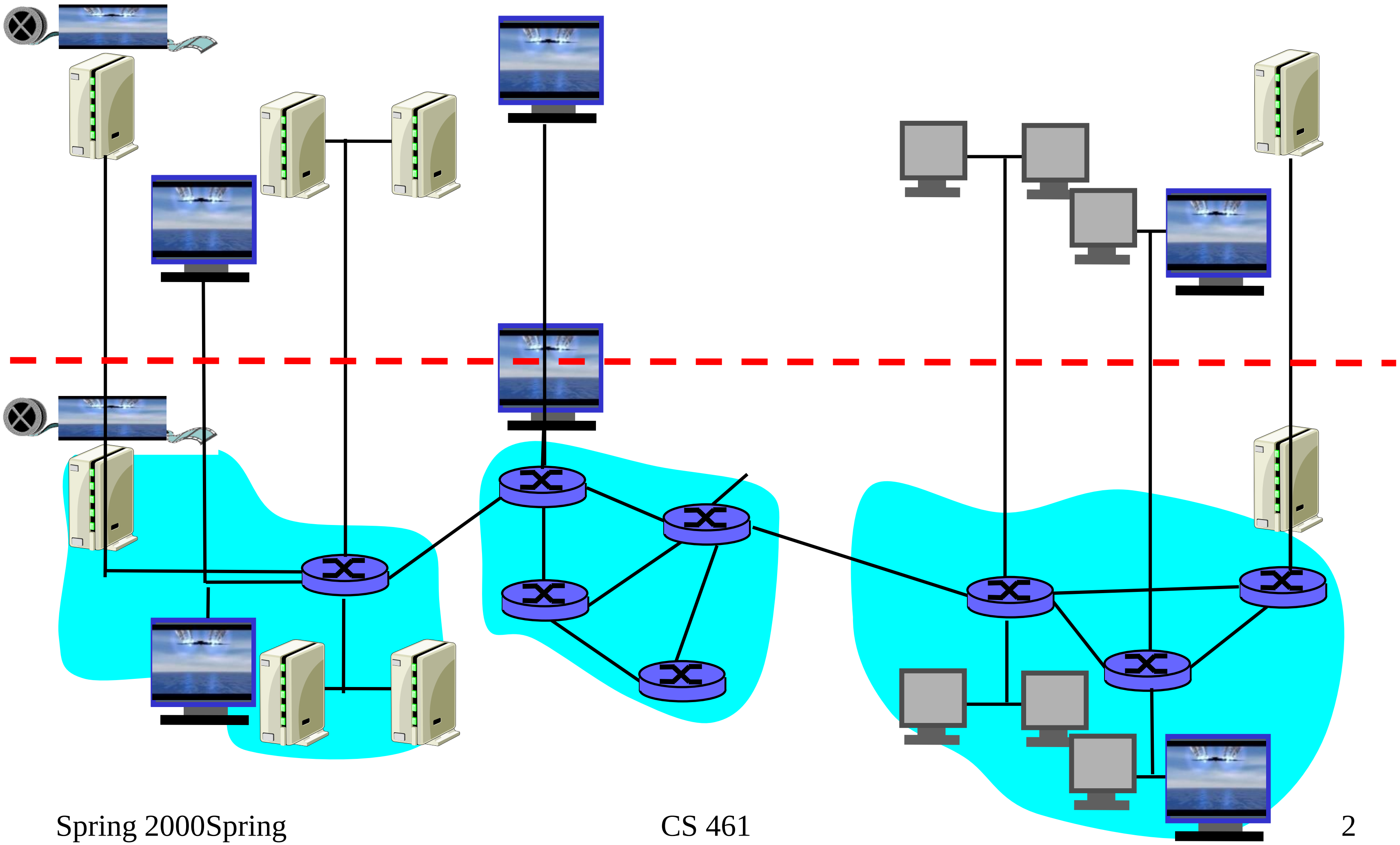
Self-organizing overlay network

File system on top of P2P network

Contributions from Peter Druschel and Don Towsley UMass-Amherst

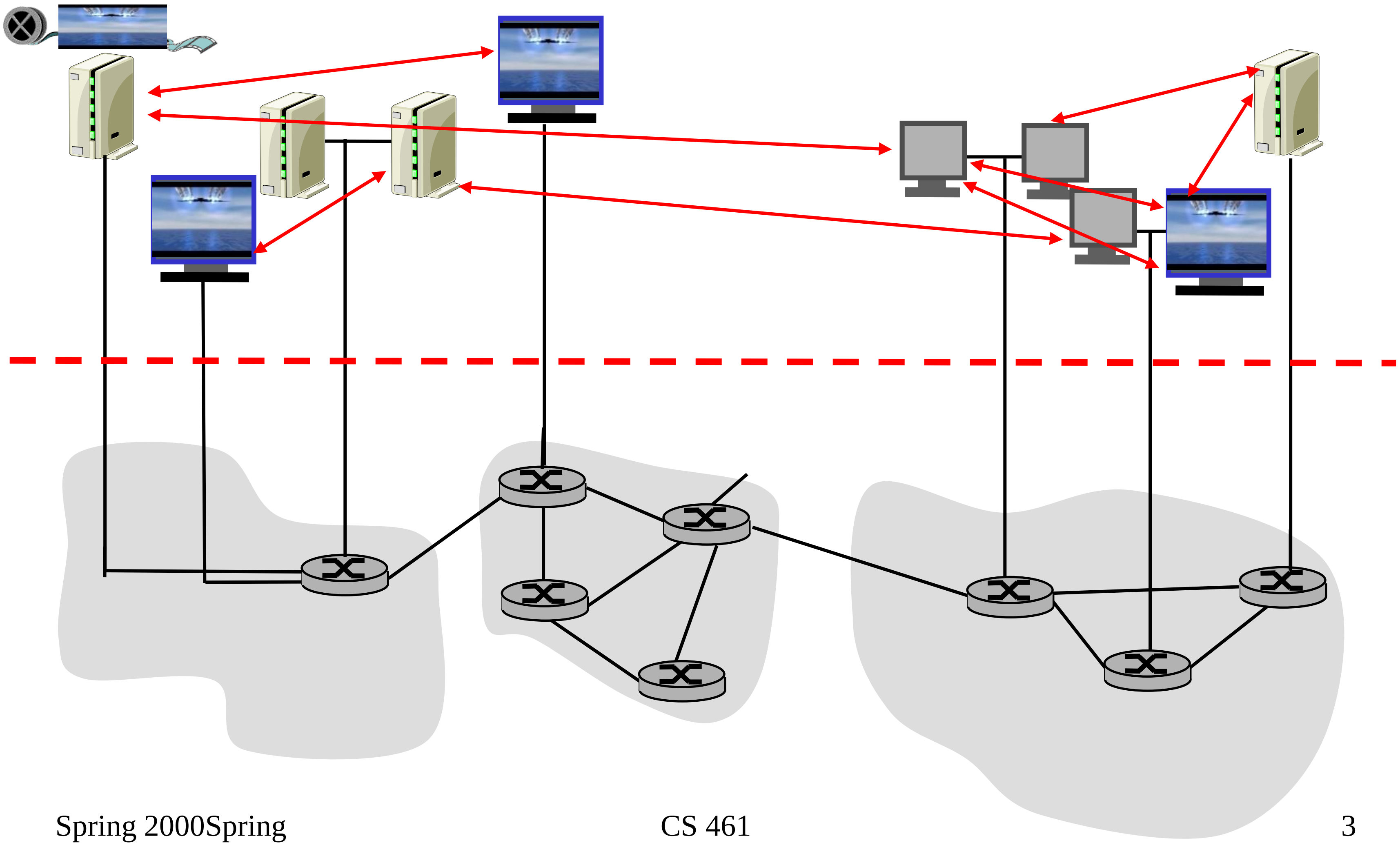
- with help of lots of others (J. Kurose, B. Levine, J. Crowcroft, CMPSCI 791N class)

Peer-peer networking



Peer-peer networking

Focus at the application level



Background

- Distribution
- Decentralized control
- Self-organization
- Symmetric communication

Examples

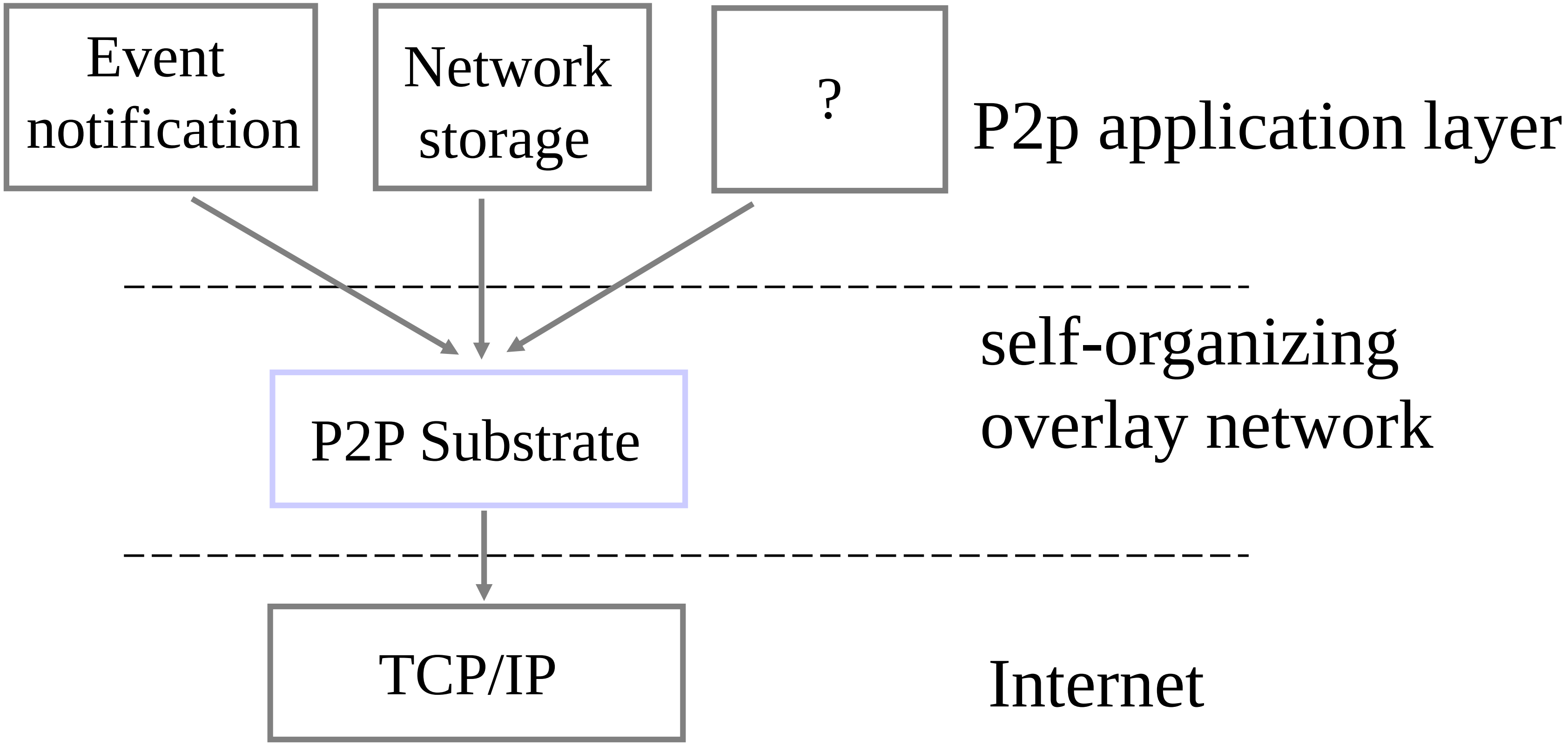
- Pioneers
  - Napster, Gnutella, FreeNet
- Academic Prototypes
  - Pastry, Chord, CAN,…

Common Issues

- Organize, maintain overlay network
  - node arrivals
  - node failures
- Resource allocation/load balancing
- Resource location
- Locality (network proximity)

Idea: generic p2p substrate

Architecture



Event notification

Network storage

?

P2p application layer

P2P Substrate

self-organizing overlay network

TCP/IP

Internet

Client Server v. Peer to Peer(1)

- RPC/RMI
- synchronous
- assymmetric
- emphasis on language integration and binding models (stub *IDL/XDR* compilers etc)
- Kerberos style security – access control, crypto

- messages
- asynchronous
- symmetric
- emphasis on service location, content addressing, application layer routing.
- anonymity, high availability, integrity.
- harder to get right☺

Peer to peer systems actually old

- IP routers are peer to peer.
- routers discover topology, and maintain it
- routers are neither client nor server
- routers continually talk to each other
- routers inherently fault tolerant
- routers are autonomous

Peer to peer systems

- nodes have no distinguished role
- no single point of bottleneck or failure.
- need distributed algorithms for
  - service discovery (name, address, route, metric, etc)
  - neighbour status tracking
  - application layer routing (based possibly on content, interest, etc)
  - resilience, handing link and node failures
  - …

Ad hoc networks and peer2peer

- wireless ad hoc networks have many similarities to peer to peer systems
- no *a priori knowledge*
- no given infrastructure
- have to construct it from “thin air”!

Overlays and peer 2 peer systems

- P2p technology often used to create overlays offering services that could be offered in the IP level
- useful **deployment** strategy
- often economically a way around other barriers to deployment
- IP was an overlay (on telephone core infrastructure)
- not all overlays are P2P (AKAMAI)

P2P Architecture Classification

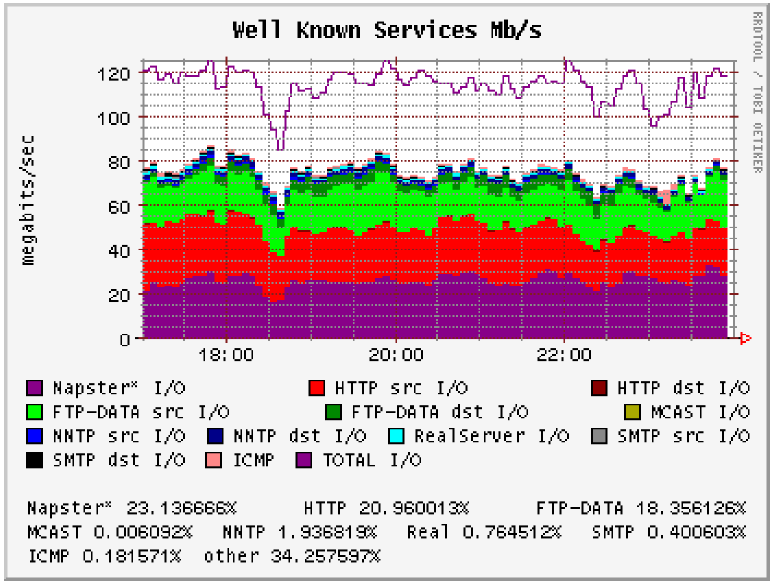
- centralized service location (CSL)
  - Napster
- distributed service location with flooding (DSLF)
  - Gnutella
- distributed service location with hashing (DSLH)
  - CAN, Pastry, Tapestry, Chord

NAPSTER

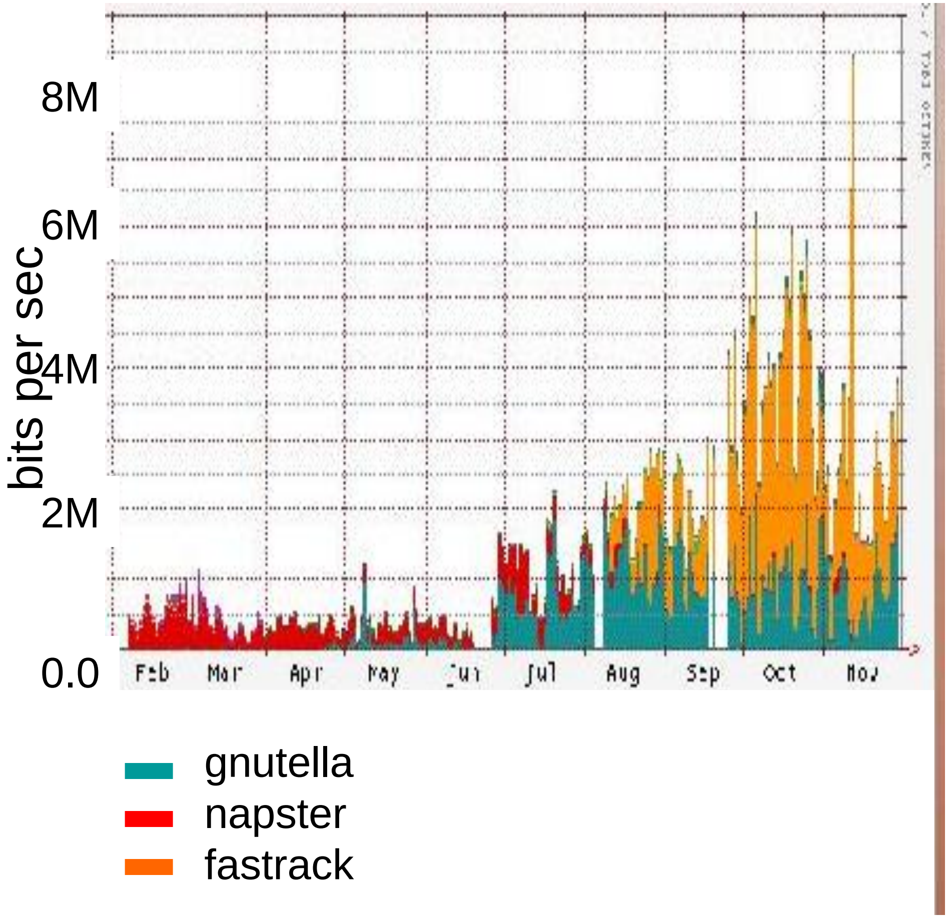
- the most (in)famous
- not the first (c.f. probably Eternity, from Ross Anderson in Cambridge)
- but instructive for what it gets right, and
- also wrong…
- also has a political message…and economic and legal…

Napster

- program for sharing files over the Internet
- a “disruptive” application/technology?
- history:
  - 5/99: Shawn Fanning (freshman, Northeasten U.) founds Napster Online music service
  - 12/99: first lawsuit
  - 3/00: 25% UWisc traffic Napster
  - 2/01: US Circuit Court of Appeals: Napster knew users violating copyright laws
  - 7/01: # simultaneous online users: Napster 160K, Gnutella: 40K, 300K



- judge orders napster to stop in July '01
- other filesharing apps take over!



Napster: how does it work

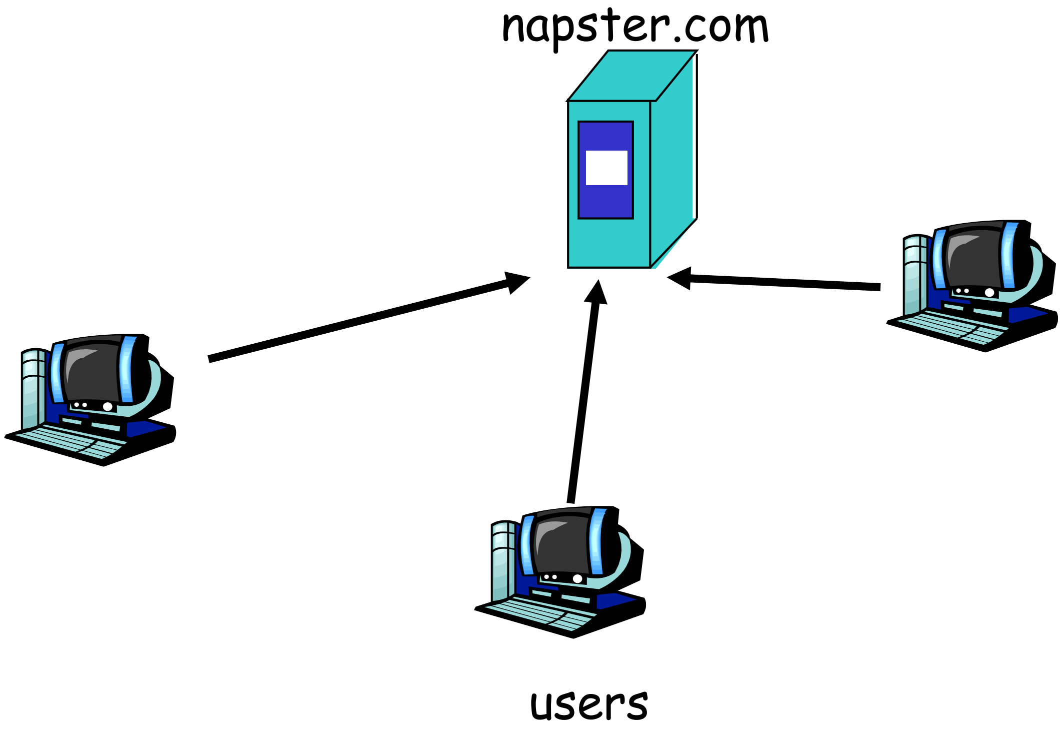
Application-level, client-server protocol over point-to-point TCP

Four steps:

- connect to Napster server
- upload your list of files (push) to server.
- give server keywords to search the full list with.
- select "best" of correct answers. (pings)

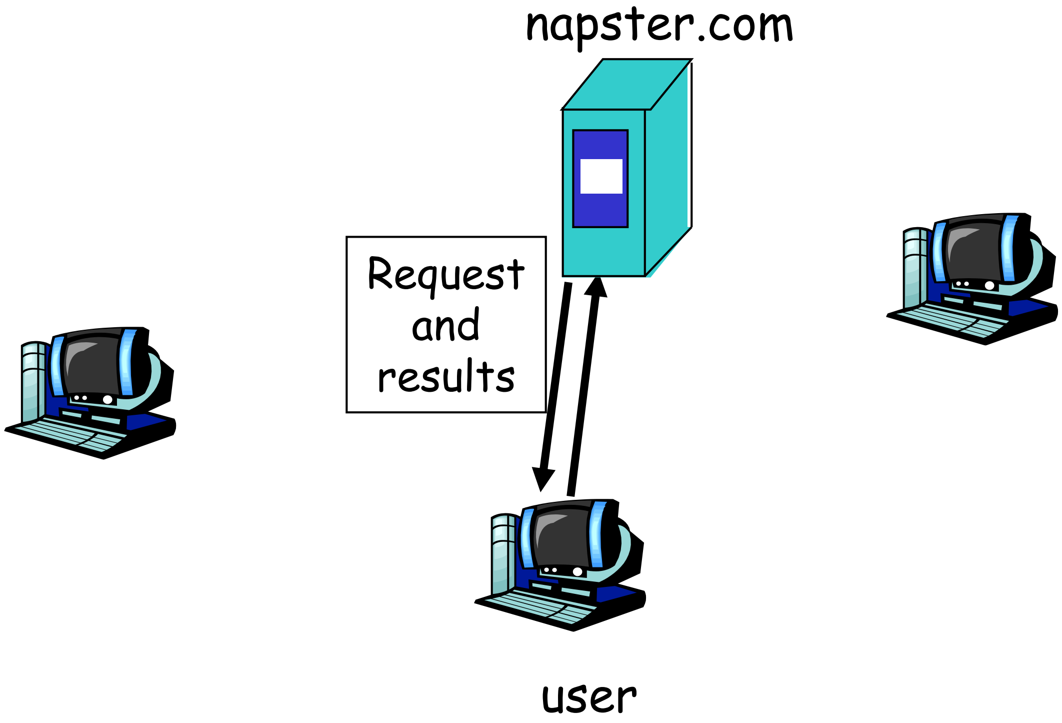
Napster

1. File list is uploaded



Napster

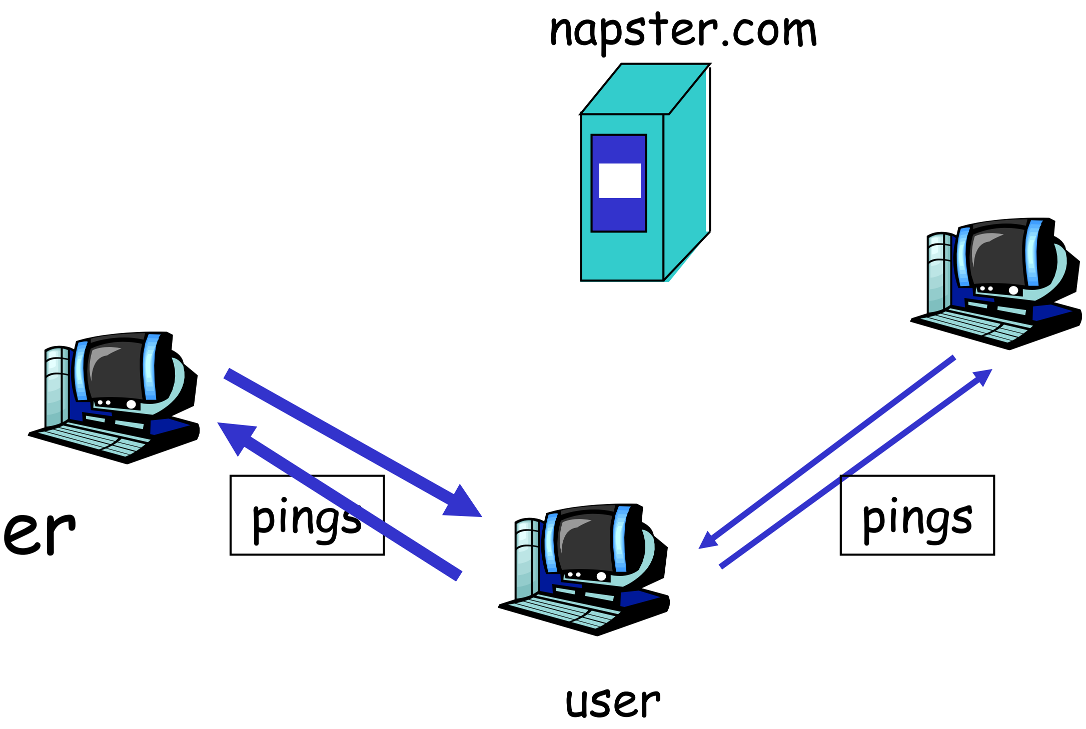
2. User requests search at server.



Napster

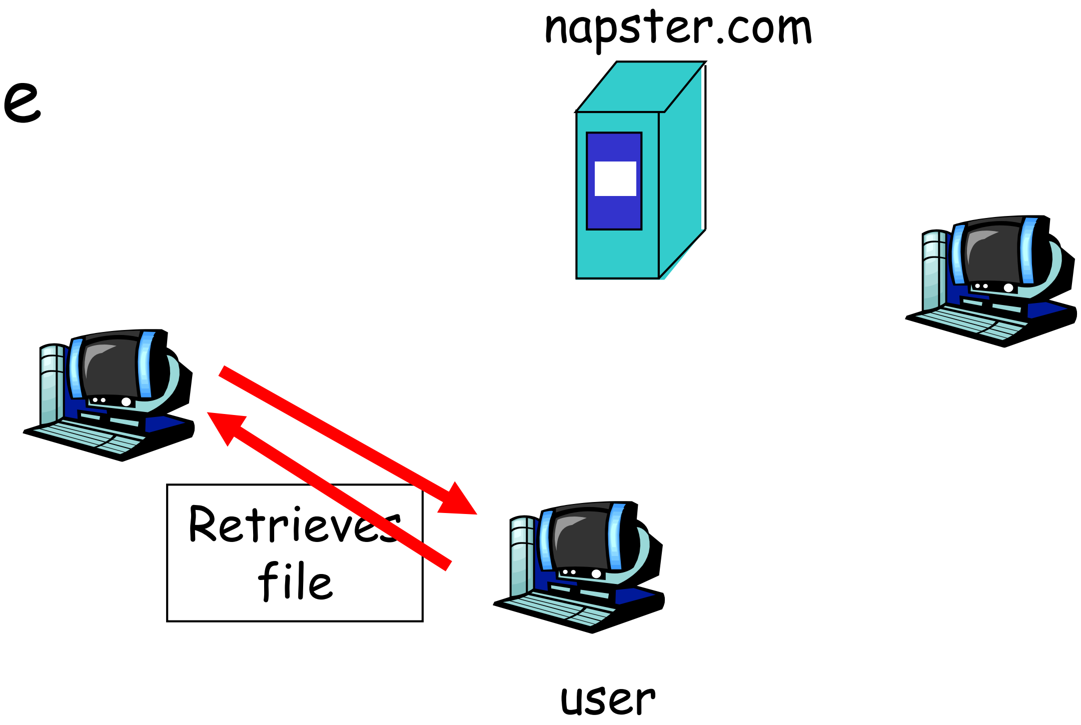
3. User pings hosts that apparently have data.

   Looks for ***best*** transfer rate.



Napster

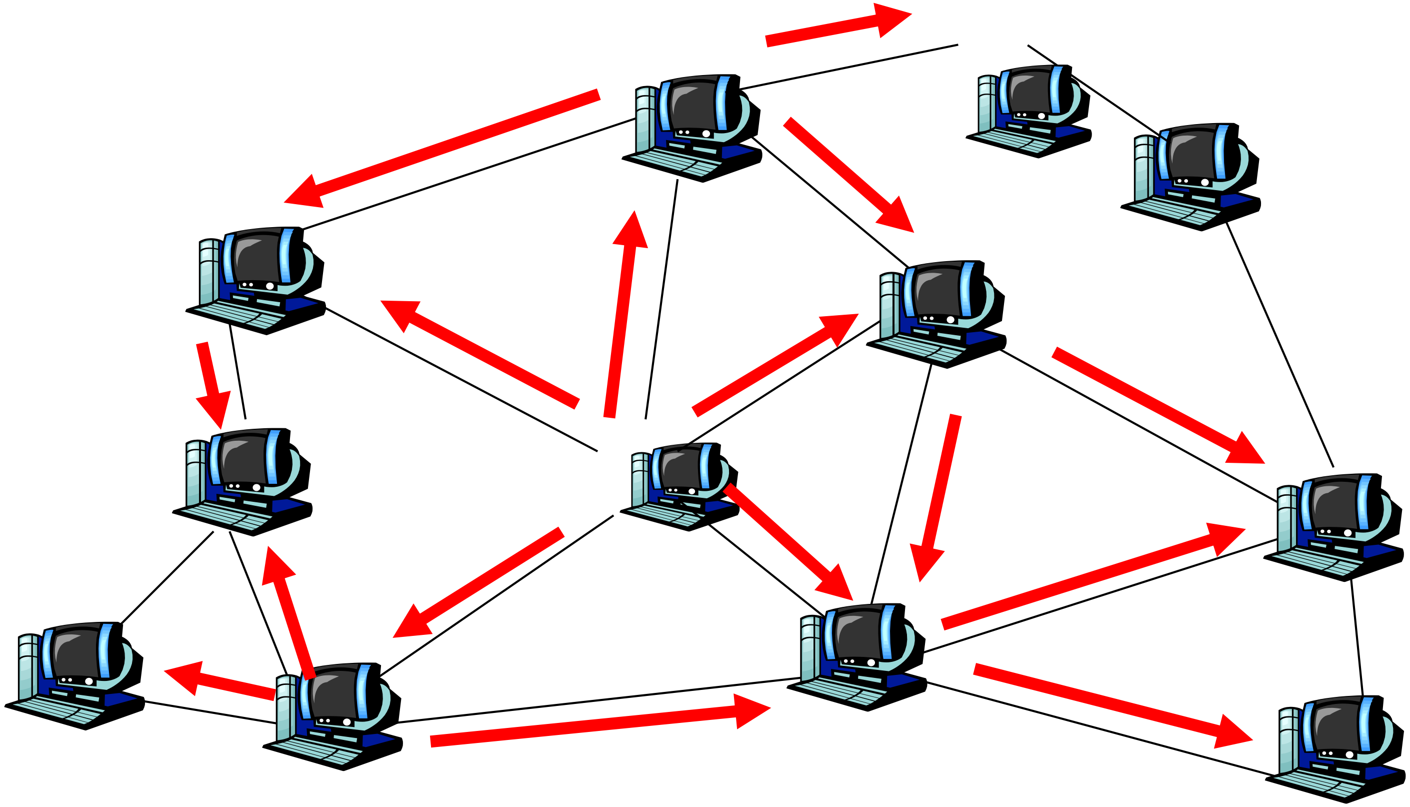
4. User retrieves file



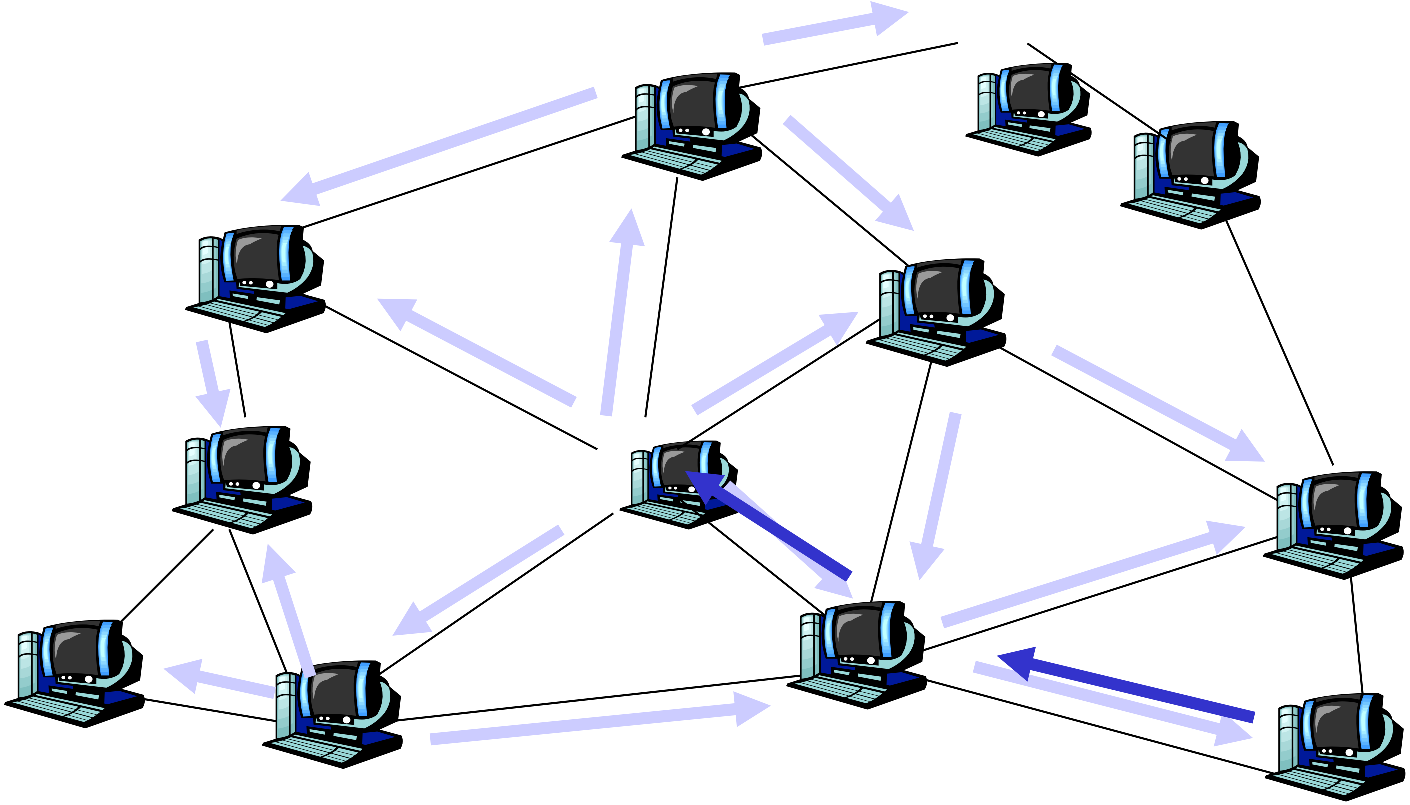
Napster: architecture notes

- centralized server:
  - single logical point of failure
  - can load balance among servers using DNS rotation
  - potential for congestion
  - Napster “in control” (freedom is an illusion)
- no security:
  - passwords in plain text
  - no authentication
  - no anonymity

Distributed Search/Flooding



Distributed Search/Flooding



Gnutella

- peer-to-peer networking: applications connect to peer applications
- focus: decentralized method of searching for files
- each application instance serves to:
  - store selected files
  - route queries (file searches) from and to its neighboring peers
  - respond to queries (serve file) if file stored locally

Gnutella

- Gnutella history:
  - 3/14/00: release by AOL, almost immediately withdrawn
  - too late: 23K users on Gnutella at 8 am this AM
  - many iterations to fix poor initial design (poor design turned many people off)
- what we care about:
  - how much traffic does one query generate?
  - how many hosts can it support at once?
  - what is the latency associated with querying?
  - is there a bottleneck?

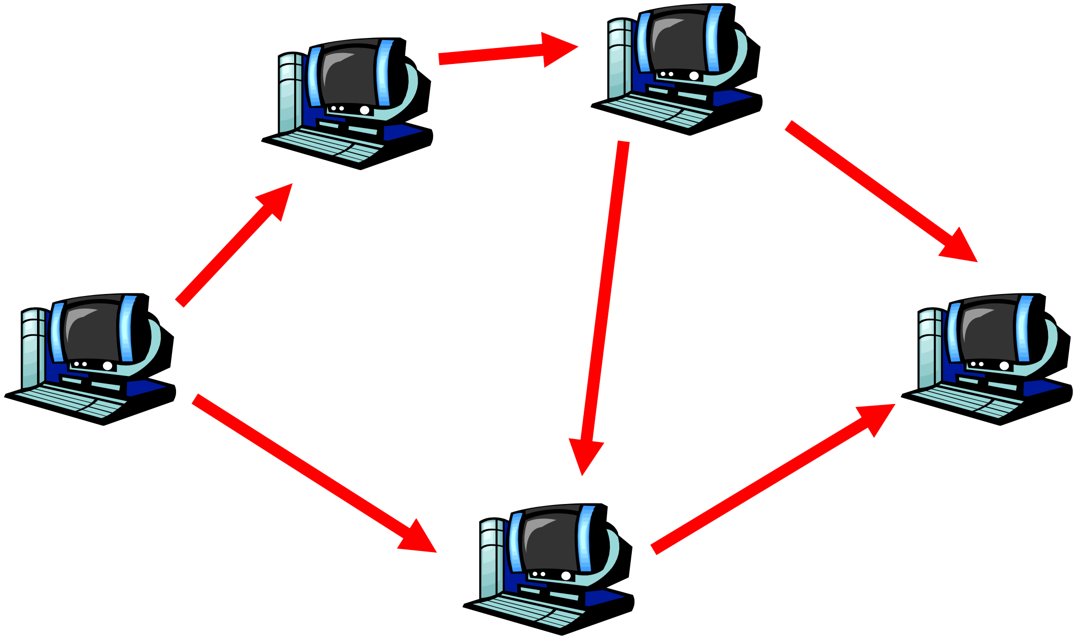
Gnutella: how it works

Searching by flooding:

- if you don't have the file you want, query 7 of your partners.
- if they don't have it, they contact 7 of their partners, for a maximum hop count of 10.
- requests are flooded, but there is no tree structure.
- no looping but packets may be received twice.
- reverse path forwarding(?)

Note: Play gnutella animation at:
http://www.limewire.com/index.jsp/p2p

Flooding in Gnutella: loop prevention



Seen already list: "A"

www.limewire.com/index.jsp/net_improvements

Gnutella: initial problems and fixes

- freeloading: WWW sites offering search/retrieval from Gnutella network without providing file sharing or query routing.
  - Block file-serving to browser-based non-file-sharing users
- prematurely terminated downloads:
  - long download times over modems
  - modem users run gnutella peer only briefly (Napster problem also!) or any users becomes overloaded
  - fix: peer can reply "I have it, but I am busy. Try again later"
  - late 2000: only 10% of downloads succeed
  - 2001: more than 25% downloads successful (is this success or failure?)

www.limewire.com/index.jsp/net_improvements

Gnutella: initial problems and fixes (more)

- 2000: avg size of reachable network only 400-800 hosts. Why so smalll?
  - modem users: not enough bandwidth to provide search routing capabilities: routing black holes
- Fix: create peer hierarchy based on capabilities
  - previously: all peers identical, most modem blackholes
  - connection preferencing:
    - favors routing to well-connected peers
    - favors reply to clients that themselves serve large number of files: prevent freeloading
  - Limewire gateway functions as Napster-like central server on behalf of other peers (for searching purposes)

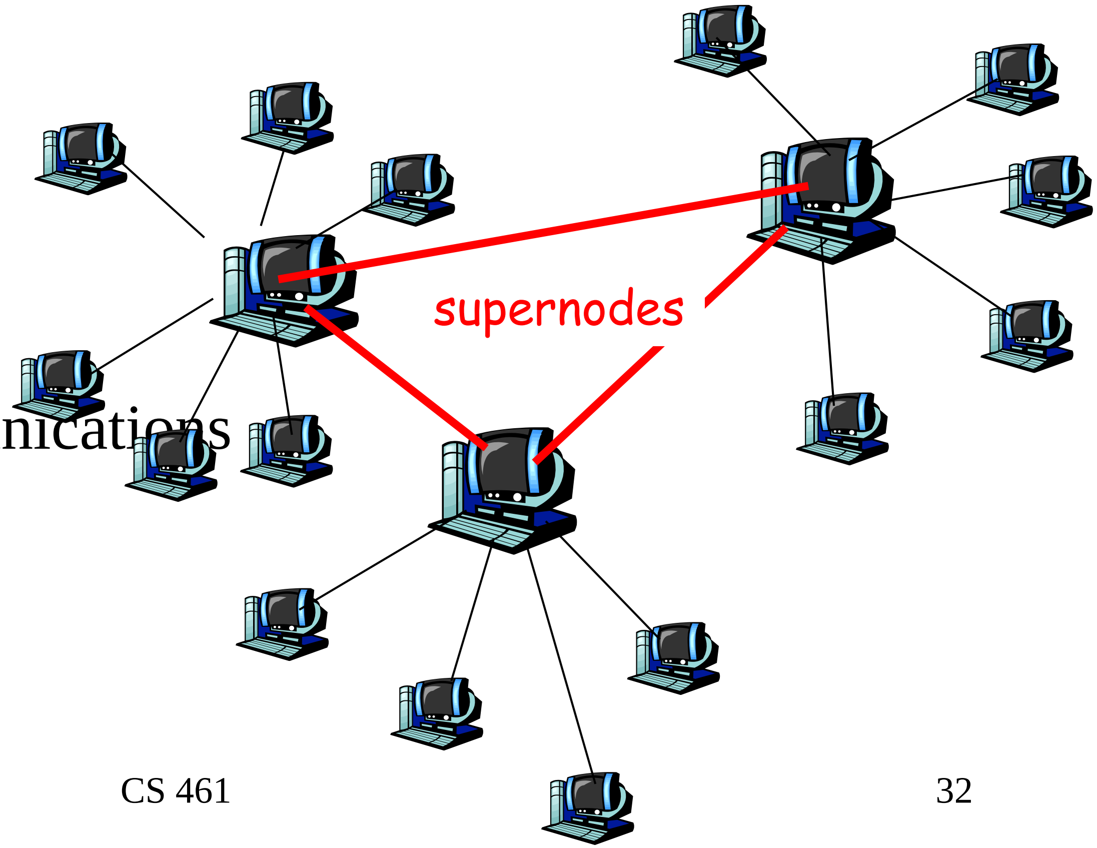
Gnutella Discussion:

- architectural lessons learned?
- anonymity and security?
- other?
- good source for technical info/open questions:

  http://www.limewire.com/index.jsp/tech_papers

Kazaa

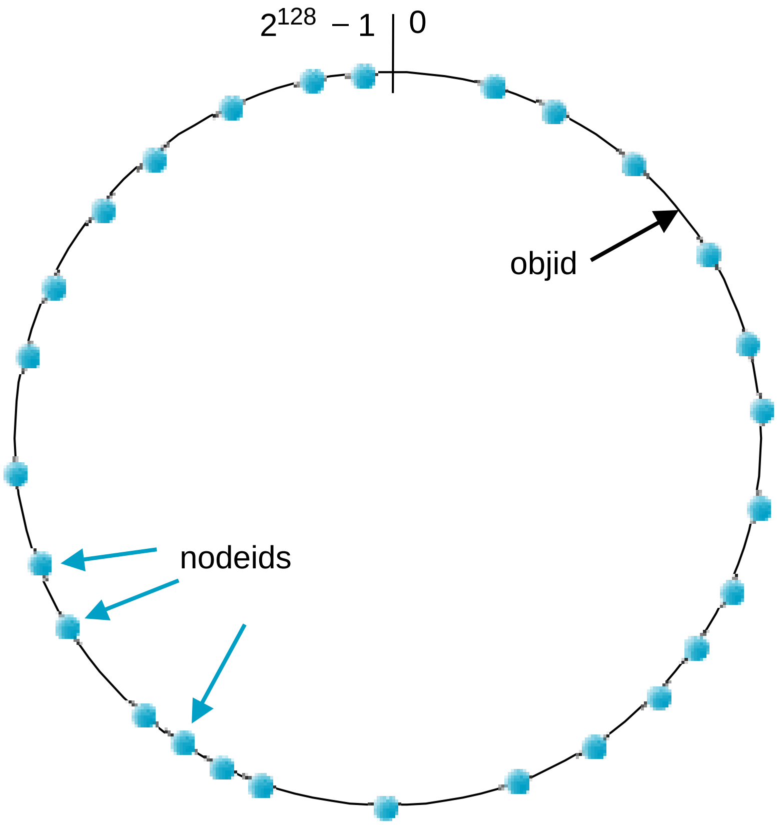
- hierarchical Gnutella
  - supernodes and regular nodes
- most popular p2p app
  - >120M downloads
- not well understood
  - binaries
  - encrypted communications



Pastry

- Self-organizing overlay network
- Consistent hashing
- Lookup/insert object in < $log_{16} N$ routing steps (expected)
- *O(log N)* per-node state
- Network locality heuristics

Object Distribution



Consistent hashing [*Karger et al. ‘97*]

128 bit circular id space

nodeIds (uniform random)

objIds (uniform random)

Invariant: node with numerically closest nodeId maintains object

Content-Addressable Network

[Ratnasamy,etal]

- introduction
- design
- evaluation
- strengths & weaknesses
- ongoing work

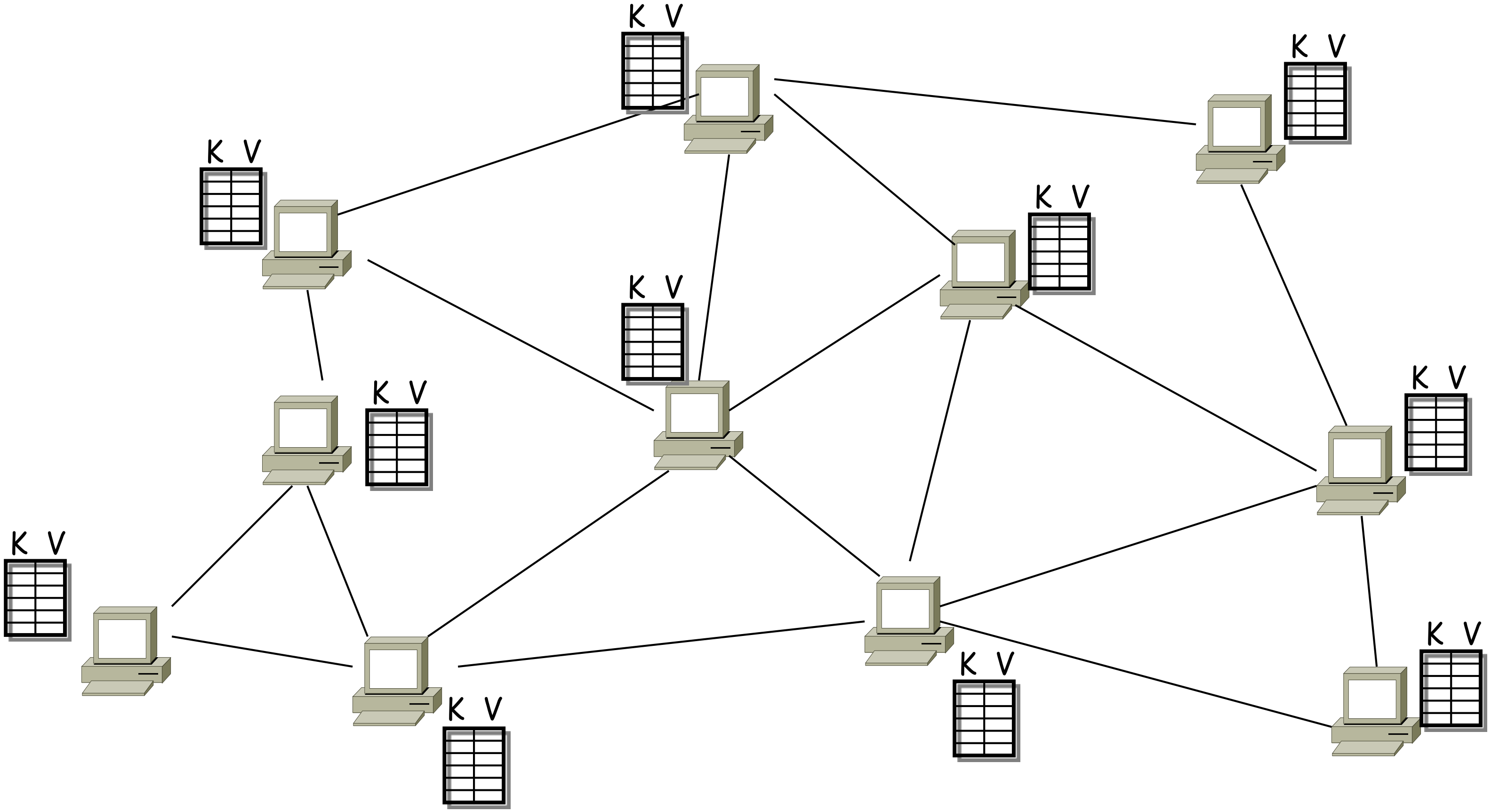
Content-Addressable Network (CAN)

- CAN: Internet-scale hash table
- interface
  - insert(key,value)
  - value = retrieve(key)

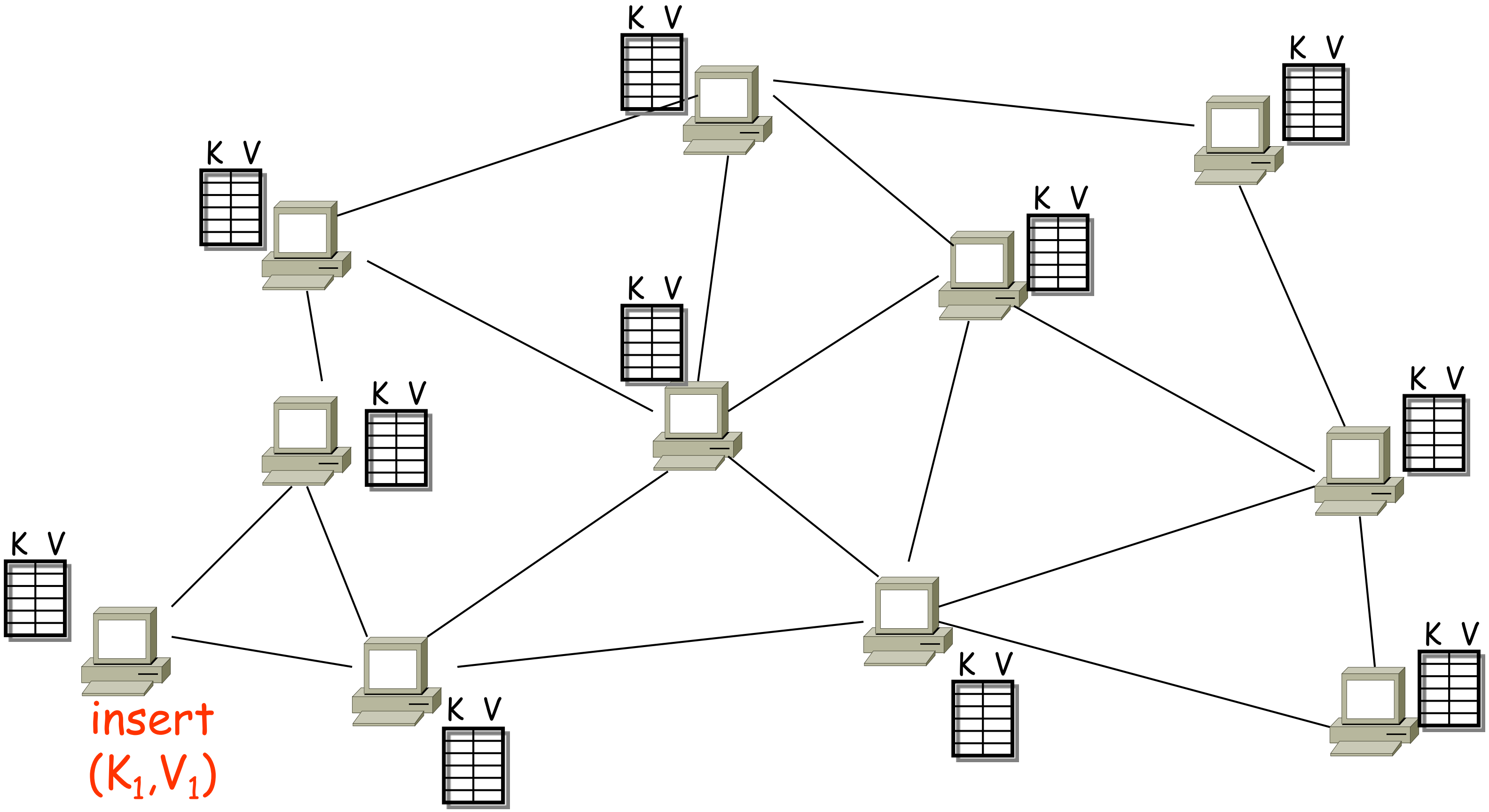
Content-Addressable Network (CAN)

- CAN: Internet-scale hash table
- interface
  - insert(key,value)
  - value = retrieve(key)
- properties
  - scalable
  - operationally simple
  - good performance (w/ improvement)

CAN: basic idea

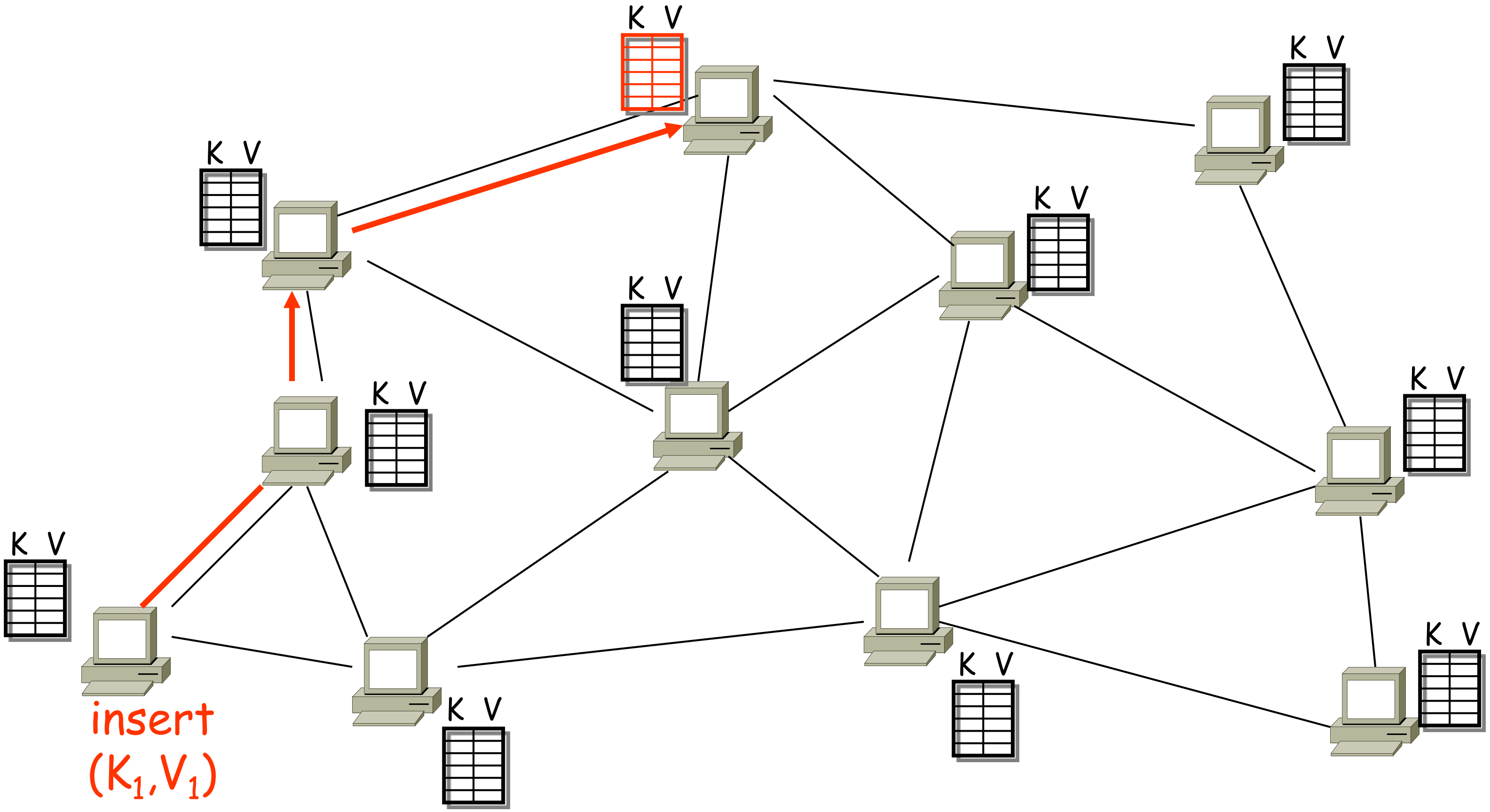


CAN: basic idea

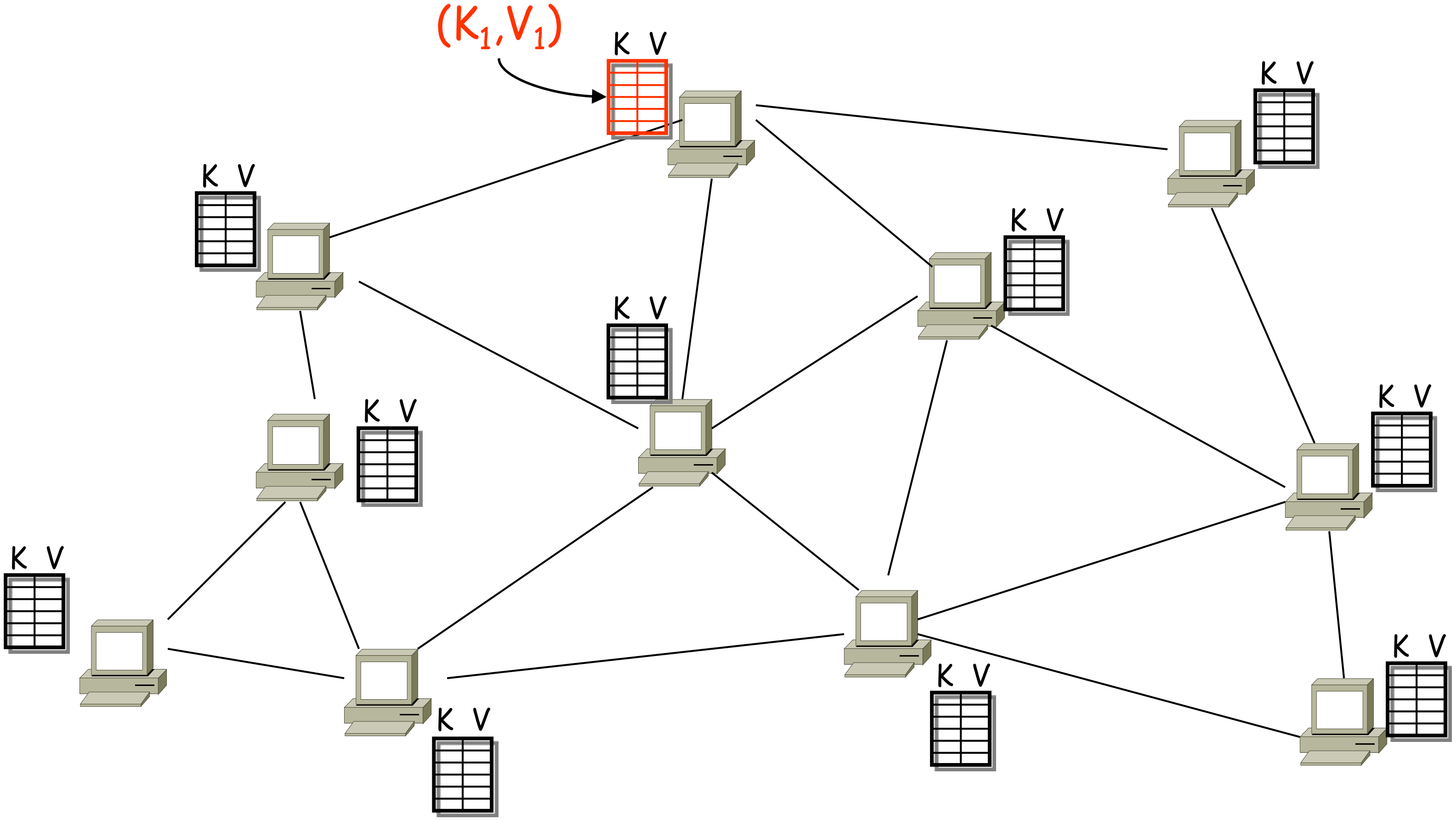


insert
(K_1,V_1)

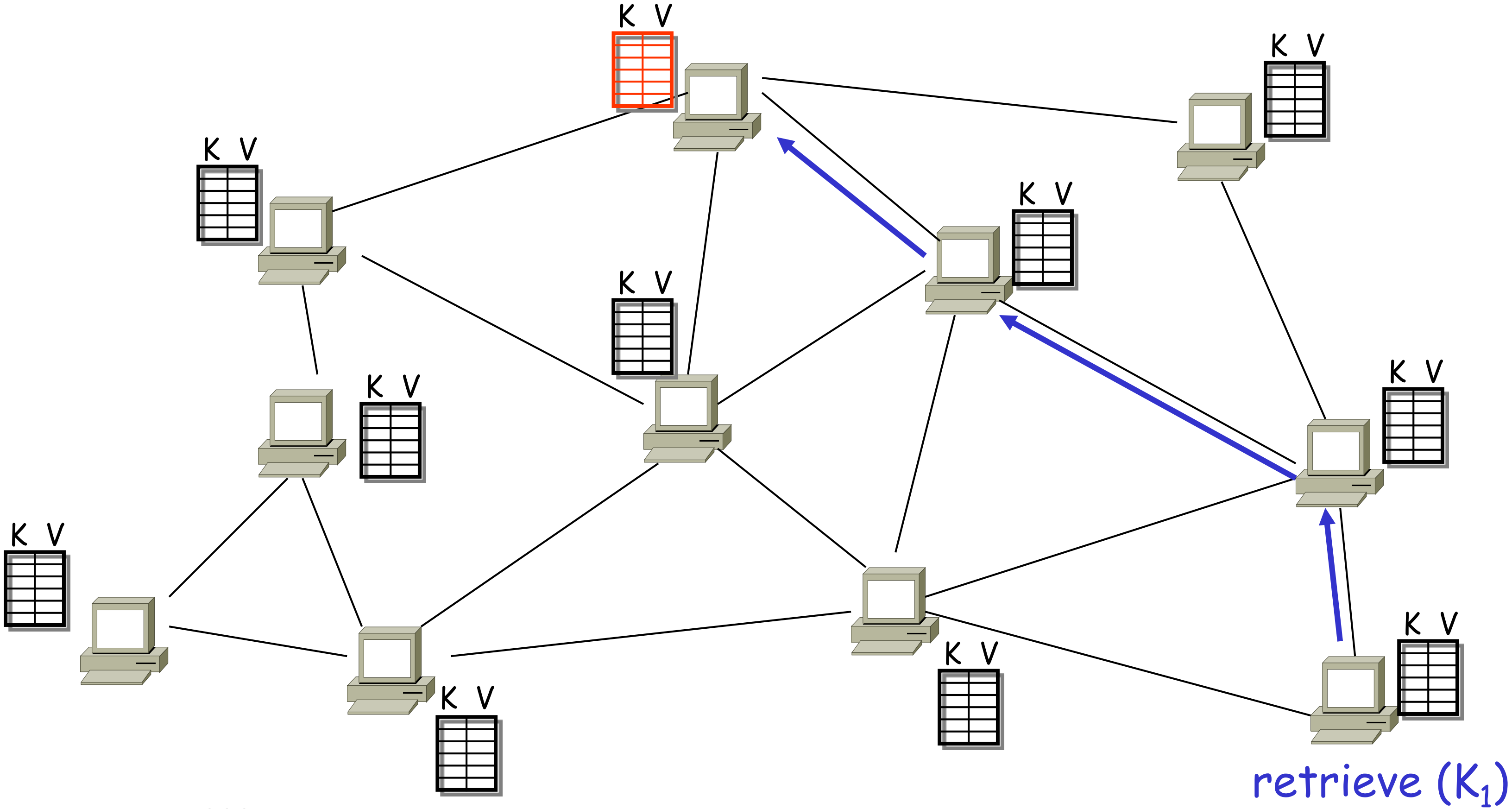
CAN: basic idea



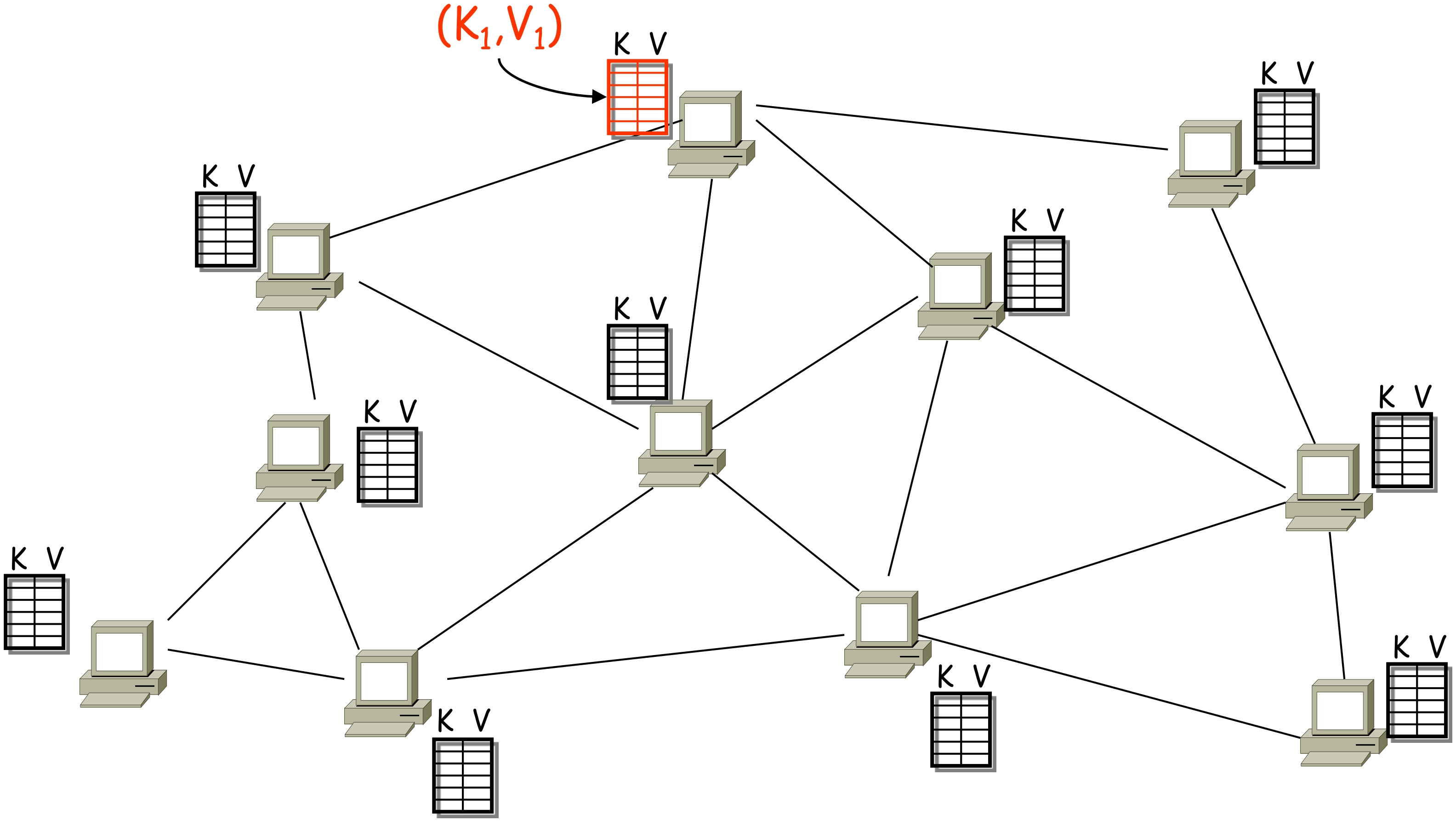
CAN: basic idea



CAN: basic idea



CAN: basic idea



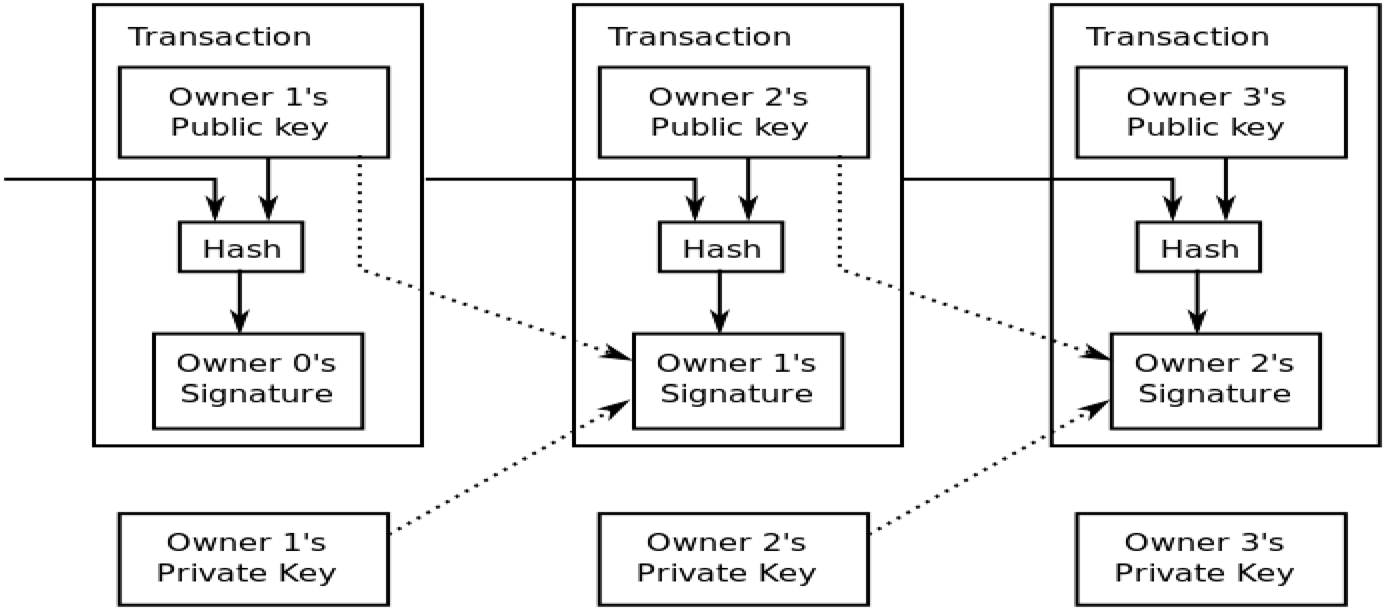
Name based routing for mobile users

Name based
routing (locality)

The rest of slides for blockchains, Bitcoin, and NFTs, are abstracted from Wikipedia

Blockchain (from Wikipedia)

- A blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptography.
- Each block typically contains a hash pointer as a link to a previous block, a timestamp (and a nouce) and transaction data.



Blockchain

- It is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way".
- For use as a distributed ledger, a blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for validating new blocks.
- Once recorded, the data in any given block cannot be altered retroactively without the alteration of all subsequent blocks, which requires collusion of the network majority.

Blockchain (P2p substrate)

- A node in the P2p network supports relaying transactions, validation or hosting a copy of the blockchain.
- In terms of relaying transactions, each node has a copy of the blockchain of the cryptocurrency it supports.
- When a transaction is made the node creating the transaction broadcasts details of the transaction using encryption to other nodes throughout the node network so that the transaction (and every other transaction) is known.
- Node owners are either volunteers, those hosted by the organization or body responsible for developing the cryptocurrency blockchain network technology, or those who are enticed to host a node to receive rewards from hosting the node.

Blockchain (Mining)

- *Mining* is a validation of transactions (through hashing algoritms).
- For this effort, successful miners obtain new cryptocurrency as a reward.
- The reward for finding a hash has diminished and often does not justify the investment in equipment and cooling facilities (to mitigate the heat the equipment produces), and the electricity required to run them.
- By July 2019, Bitcoin's electricity consumption was estimated to be approximately 7 gigawatts, around 0.2% of the global total, or equivalent to the energy consumed nationally by Switzerland.

Blockchain (Wallet)

- A cryptocurrency wallet stores the public and private "keys" (address) which can be used to receive or spend the cryptocurrency.
- With the private key, it is possible to write in the public ledger.
- With the public key, it is possible for others to send currency to the wallet.
- Bitcoin is pseudonymous in that the cryptocurrency within a wallet is not tied to people, but rather to specific private “keys”.

Blockchain (Bitcoin)

- A block (in Bitcoin blockchain) contains a SHA-256 cryptographic hash of the previous block, thus linking it to the previous block.
- To be accepted by the rest of the network, a new block must contain a proof-of-work (PoW).
- The PoW requires miners to find a number called a nonce (number used once), such that when the block content is hashed along with the nonce, the result is numerically smaller than the network's difficulty target.
- This proof is easy for any node in the network to verify, but extremely time-consuming to generate.
- By adjusting this difficulty target, the amount of work needed to generate a block can be changed. (As of 11 May 2020, the reward is currently 6.25 newly created bitcoins per block.)

Blockchain (Bitcoin)

- Bitcoin does not have a central authority.
- The bitcoin network is peer-to-peer, without central servers.
- The network also has no central storage; the bitcoin ledger is distributed.
- The ledger is public; anybody can store it on a computer.
- There is no single administrator; the ledger is maintained by a network of equally privileged miners.
- Anyone can become a miner.
- The additions to the ledger are maintained through competition. Until a new block is added to the ledger, it is not known which miner will create the block.
- The issuance of bitcoins is decentralized. They are issued as a reward for the creation of a new block.
- Anybody can create a new bitcoin address and send a transaction to the network.

Non-fungible token (NFT)

- Fungibility is the property of a good or a commodity whose individual units are essentially interchangeable and each of whose parts is indistinguishable from another part.
- Fungible tokens are the ones that can be exchanged or replaced; for example, a ten rupees note can easily be exchanged with two five rupees coins.
- Gold is fungible since a specified amount of pure gold is equivalent to that same amount of pure gold.
- Cryptocurrencies are fungible assets.
- An NFT is a unit of data, stored on a type of digital ledger called a blockchain (Ethereum), which can be sold and traded.
- NFTs are not mutually interchangeable, and so are not fungible.

Non-fungible token (NFT)

- The cryptographic transaction process in the underlying blockchain ensures the authentication of each digital file by providing a digital signature that tracks NFT ownership.
- Ownership of an NFT does not inherently grant copyright or intellectual property rights to the digital asset the NFT.
- An NFT is merely proof of ownership separate from copyright.
- Digital art is a common use case for NFTs.
- NFTs can represent in-game assets, such as digital plots of land.
- NFTs representing digital collectables and artworks are a speculative asset.
- NFTs, as with other blockchain securities and with traditional art sales, can potentially be used for money laundering.

Ethereum and smart contracts

- A smart contract is a computer program or a transaction protocol which is intended to automatically execute, control or document legally relevant events and actions according to the terms of a contract or an agreement.
- Ethereum is a decentralized, open-source blockchain with smart contract functionality.
- Ether is the native cryptocurrency of the platform.
- Ether is second only to Bitcoin in market capitalization.
- Ethereum also allows users to create and exchange NFTs.
- Ethereum 2.0 (also known as Serenity) aims to increase transaction throughput by splitting up the workload into many blockchains running in parallel.
- Decentralized finance (DeFi) is a use case of Ethereum.