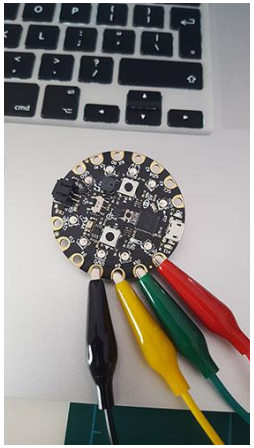
CIS373 - Pervasive Computing System Architectures

Erik Fredericks // frederer@gvsu.edu

Adapted from materials provided by Xiang Cao

But first, a clarification

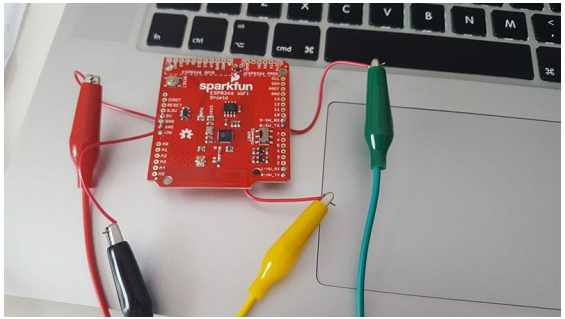


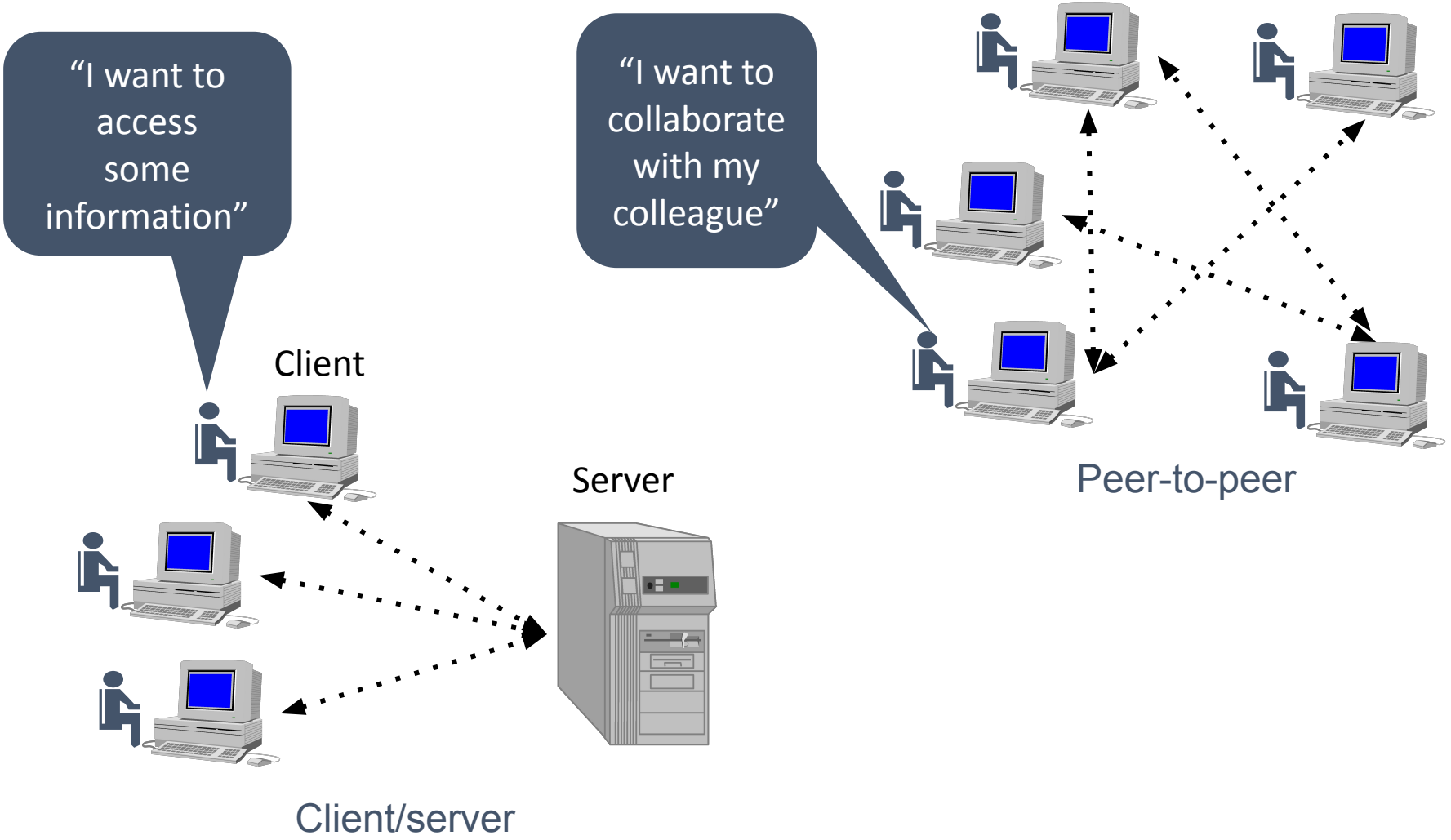
The Bluefruit does not have WiFi built in

- Bluetooth only

WiFi is an extra add-on shield
https://www.digikey.com/en/maker/projects/adding-wifi-to-the-circuit-playground-sparkfun-wifi-shield/cfc59fb70e6d49b1bd99f200ed4920ef

Fortunately, won't impact us - we can use BT for IoT!




"I want to access some information"
Client
Server
Client/server
"I want to collaborate with my colleague"
Peer-to-peer

Architecture types

Client-server

- Asymmetric relationship
- Client makes requests, server makes replies

Peer-to-peer (P2P)

- Symmetric relationship

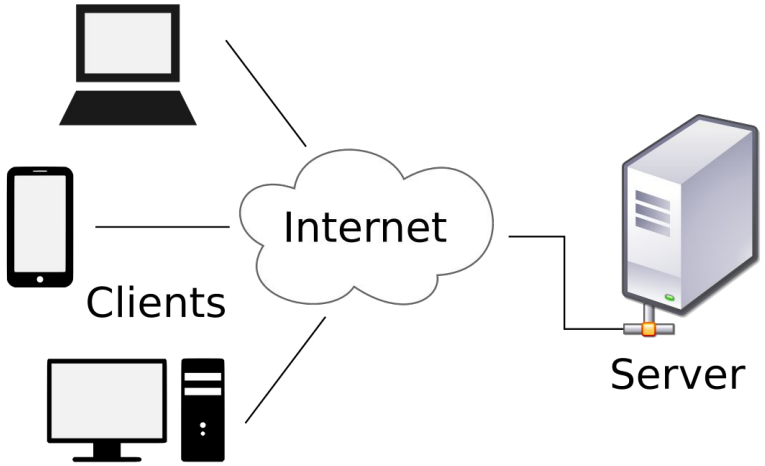
Client/Server Types???

Simple client-server

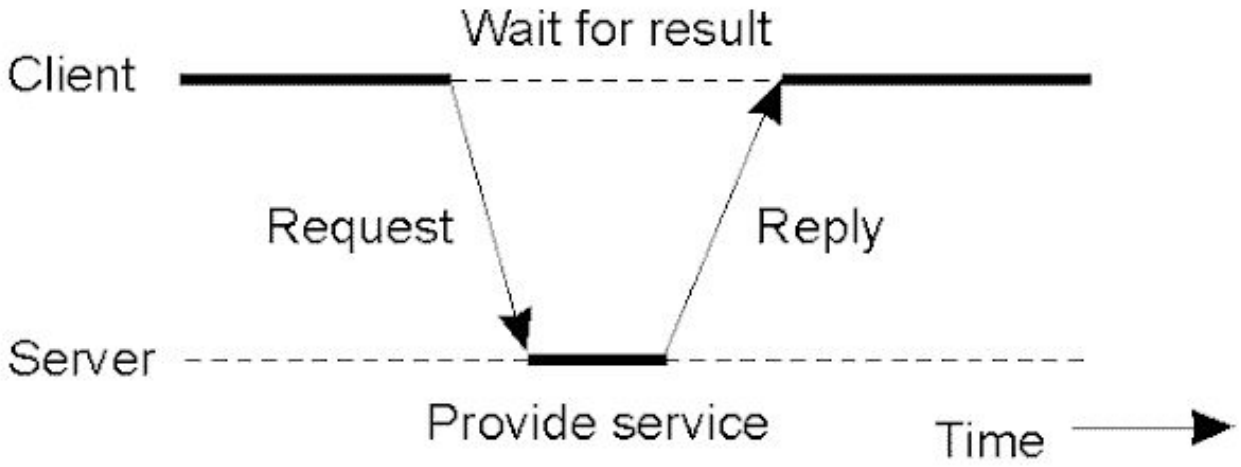
Server invoked by another server

Multiple servers

Web Applets



Client-Server Interaction



Discussion

In order to make client-server model work,

- What information should the server know from the client?

- What information should the client know from the server?

Discussion

In order to make client-server model work:

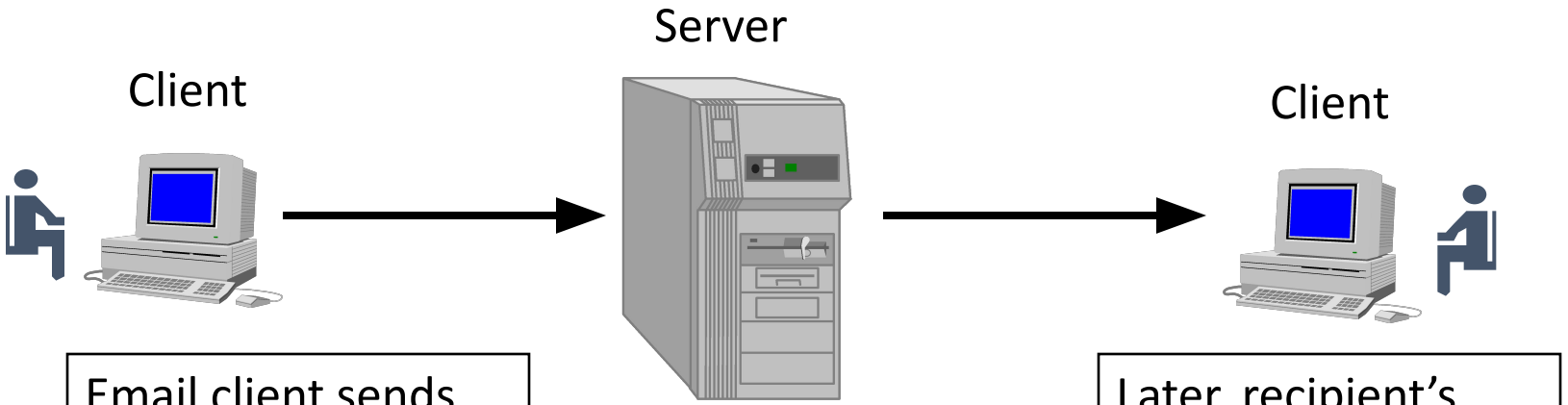
What information should the server know from the client?

- The server does not need to know the existence or address of the client prior to the connection.
- Just keep running for incoming connections from clients

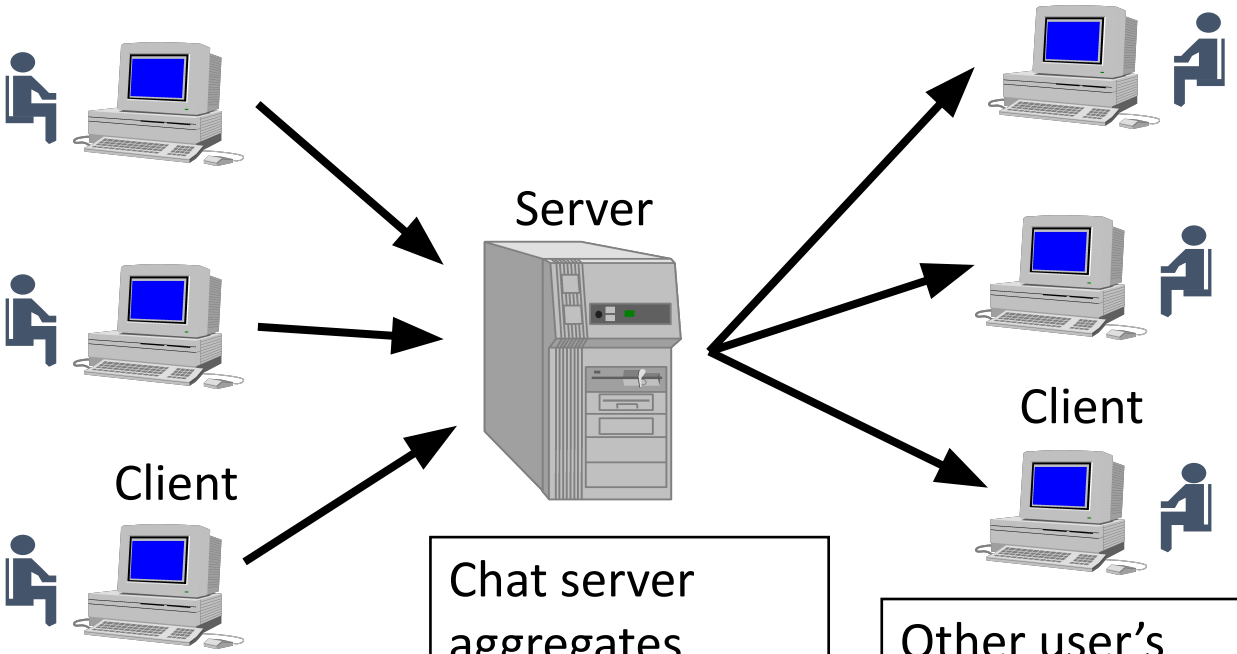
What information should the client know from the server?

- The client needs to know the existence and the address of the server.

Email application



Chat application

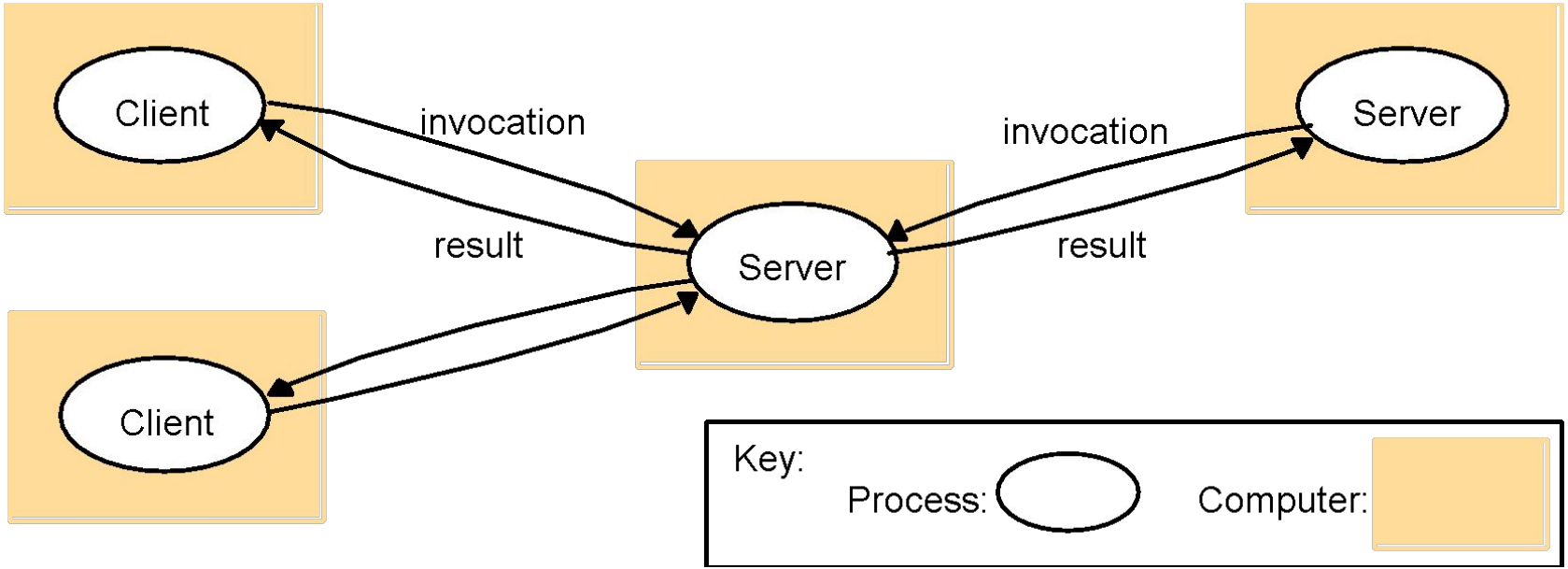


Chat clients send user's typing to server

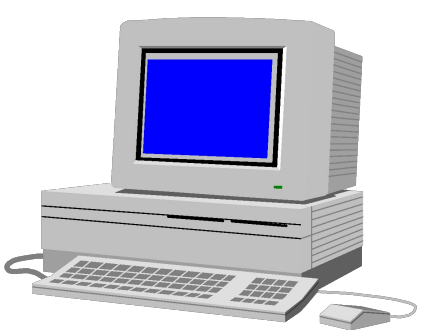
Chat server aggregates typing from all users and sends to all clients

Other user's clients display aggregated typing from chat server

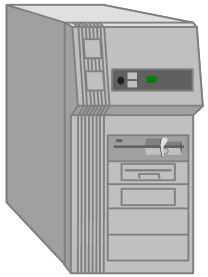
Client/Server (again)



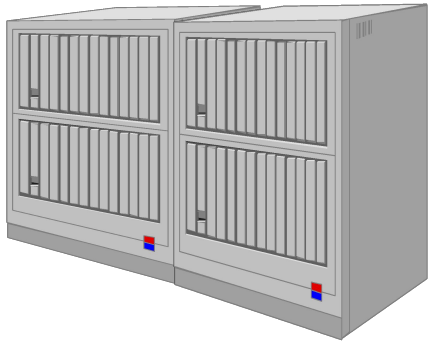
Three-tier client/server



Local-area network



Application server

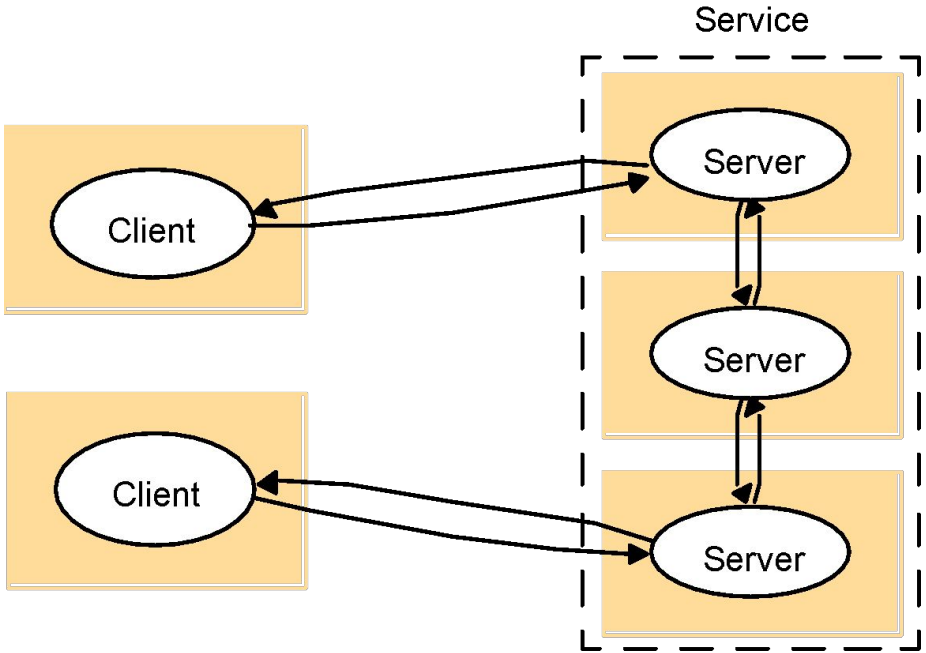


Data server

Client

Note: many clients per application server, several application servers per data server

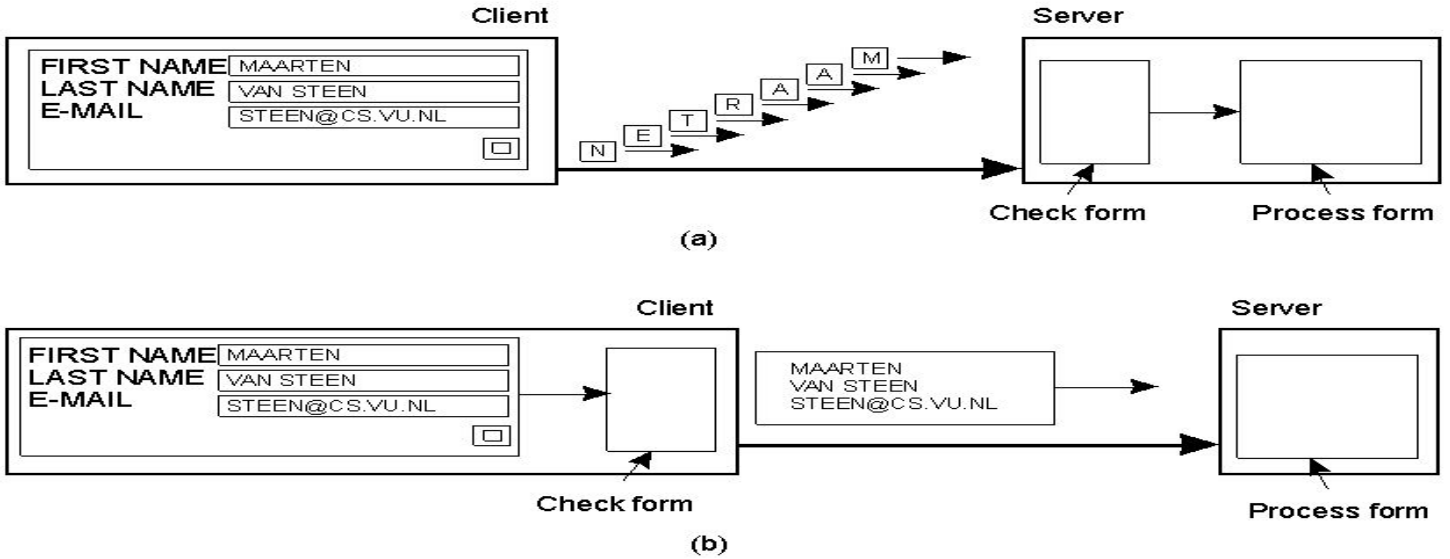
Client/Server (3)



A Service by Multiple Servers

Client/Server (4)

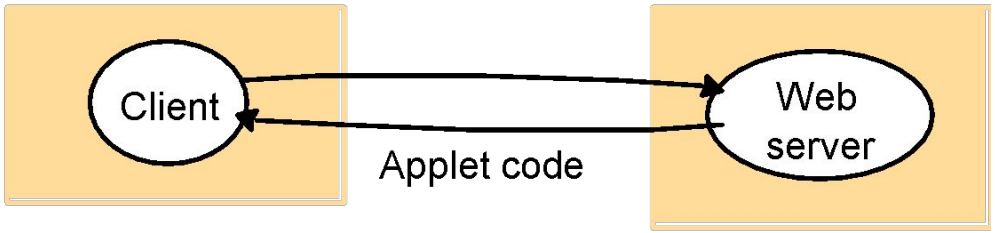
Could move computations to client ➜ but what is **wrong** with this?



Does the server check? Client check? Both?

Web applet (client/server) (mostly defunct)

a) client request results in the downloading of applet code



b) client interacts with the applet

Client/Server ➜ Limitations

- Scalability is hard to achieve
- Presents a single point of failure
- Requires administration
- Unused resources at the network edge (client side)
  - CPU cycles, storage, etc.
  - P2P systems try to address these limitations

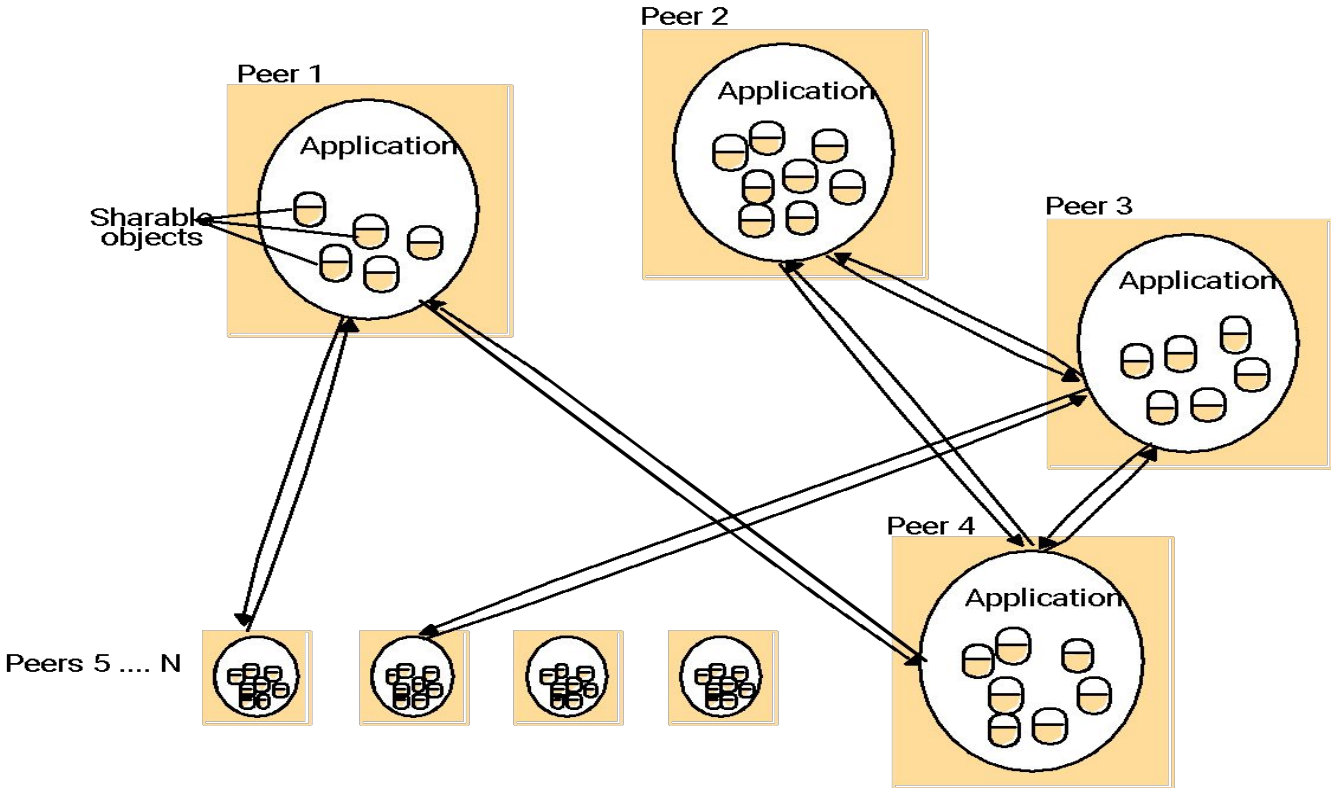
Peer-to-Peer (P2P)

What is it?

Use the **vast resources** of machines at the **edge of the Internet** to build a network that allows resource sharing with **limited central authority**

More than a system for sharing pirated music/movies!

P2P architecture



P2P - Characteristics

Exploit edge resources.

- Storage, content, CPU, human presence

Significant autonomy from any centralized authority

- Each node can act as a Client as well as a Server

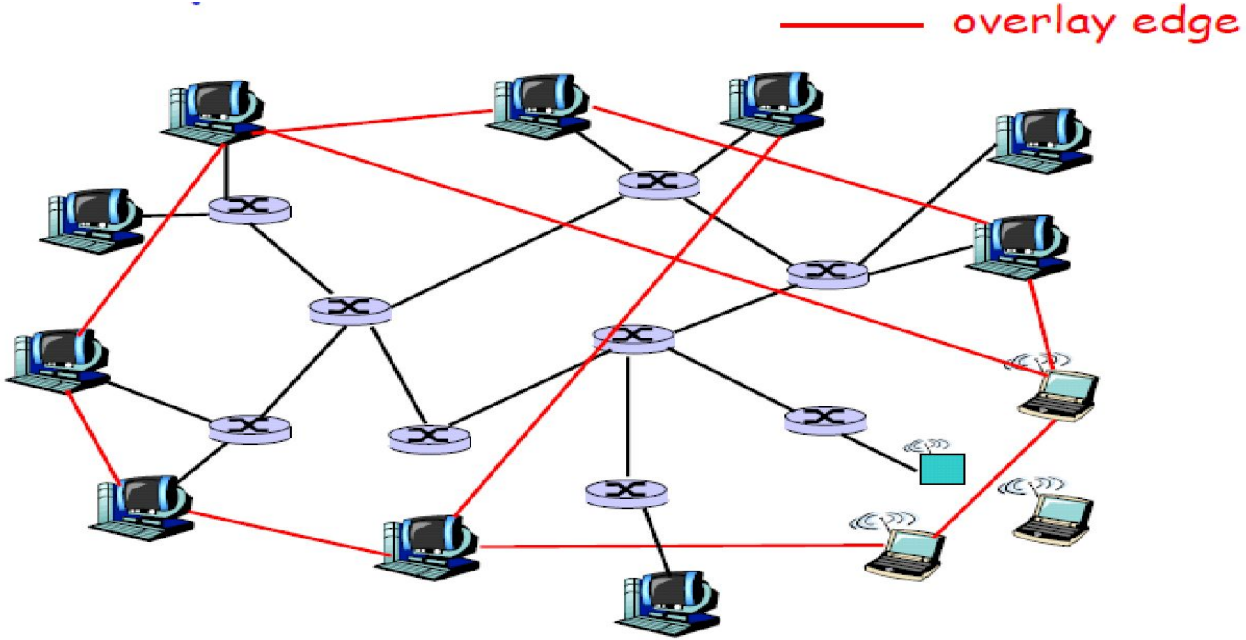
Resources at edge have intermittent connectivity

- Constantly being added & removed
- Infrastructure is untrusted and the components are unreliable

P2P - Overlay network

A P2P network is an **overlay network**

- Each link between peers consists of one or more IP links



P2P - Categories

Unstructured

- No restriction on overlay structures and data placement
- Napster, Gnutella, KaZaA

Structured

- Distributed hash tables (DHTs)
- Place restrictions on overlay structures and data placement
- Chord, Tapestry, and others

P2P - Lookup issues

How do you locate data/files/objects in a large P2P system built around a dynamic set of nodes in a scalable manner?

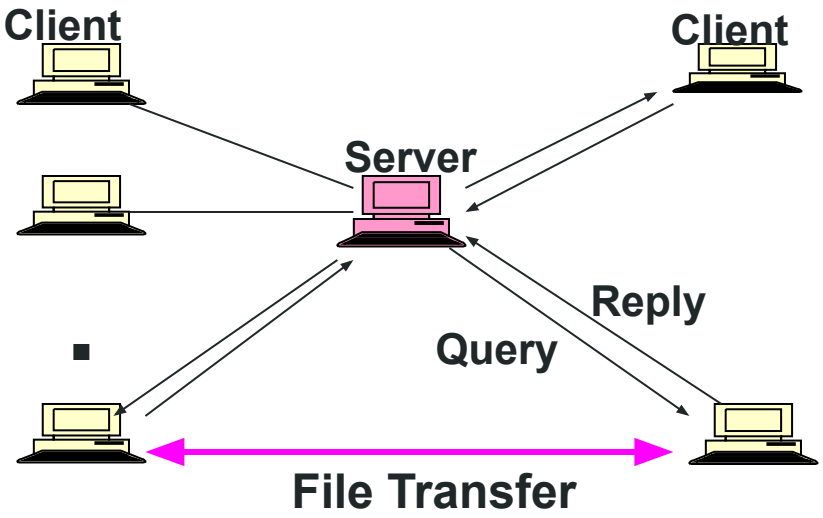
- For example, a client wants to download file A, but where to find A in the system?

Efficient routing even if the structure of the network is unpredictable

- Napster, Gnutella, KaZaA

Unstructured P2P

Napster



Share music files, MP3 data

Nodes register their contents (list of files) and IPs with server

Centralized server for searches

- The client sends queries to the centralized server for files of interest
- Keyword search (artist, song, album, bitrate, etc.)

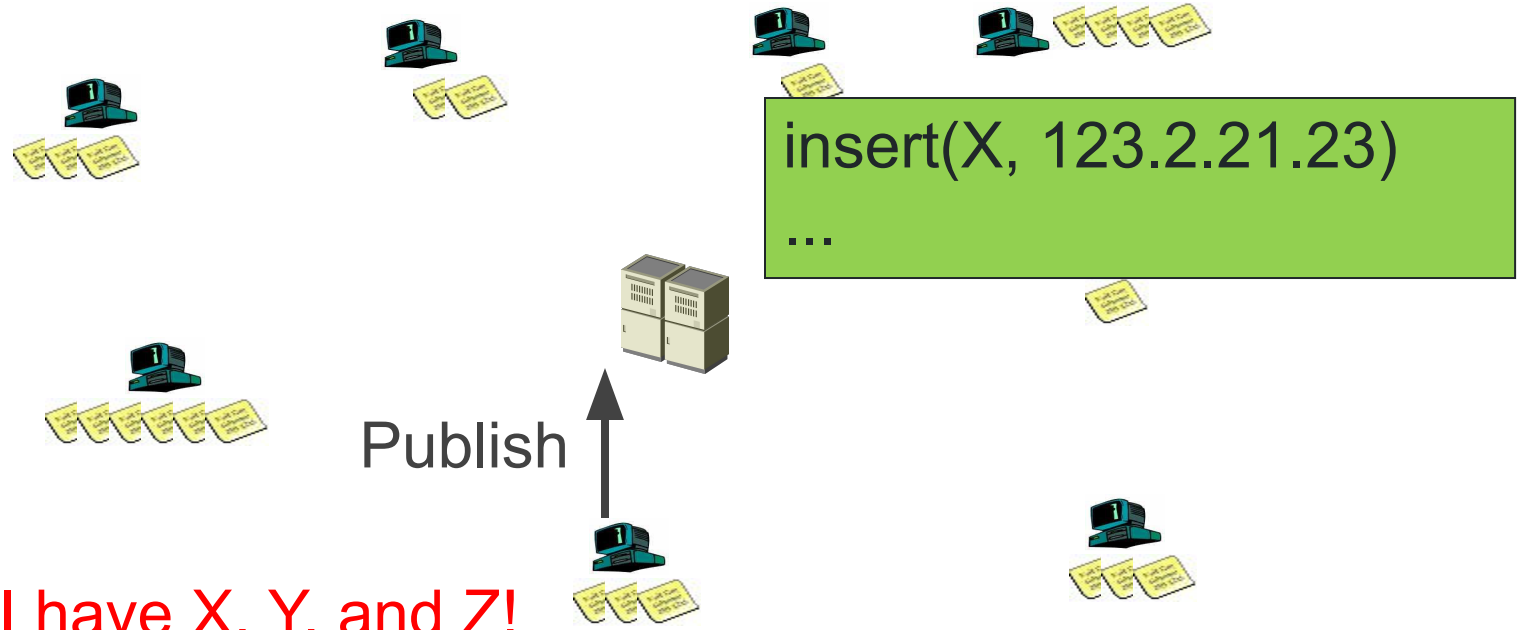
Napster server replies with IP address of users with matching files

File download done on a peer-to-peer basis

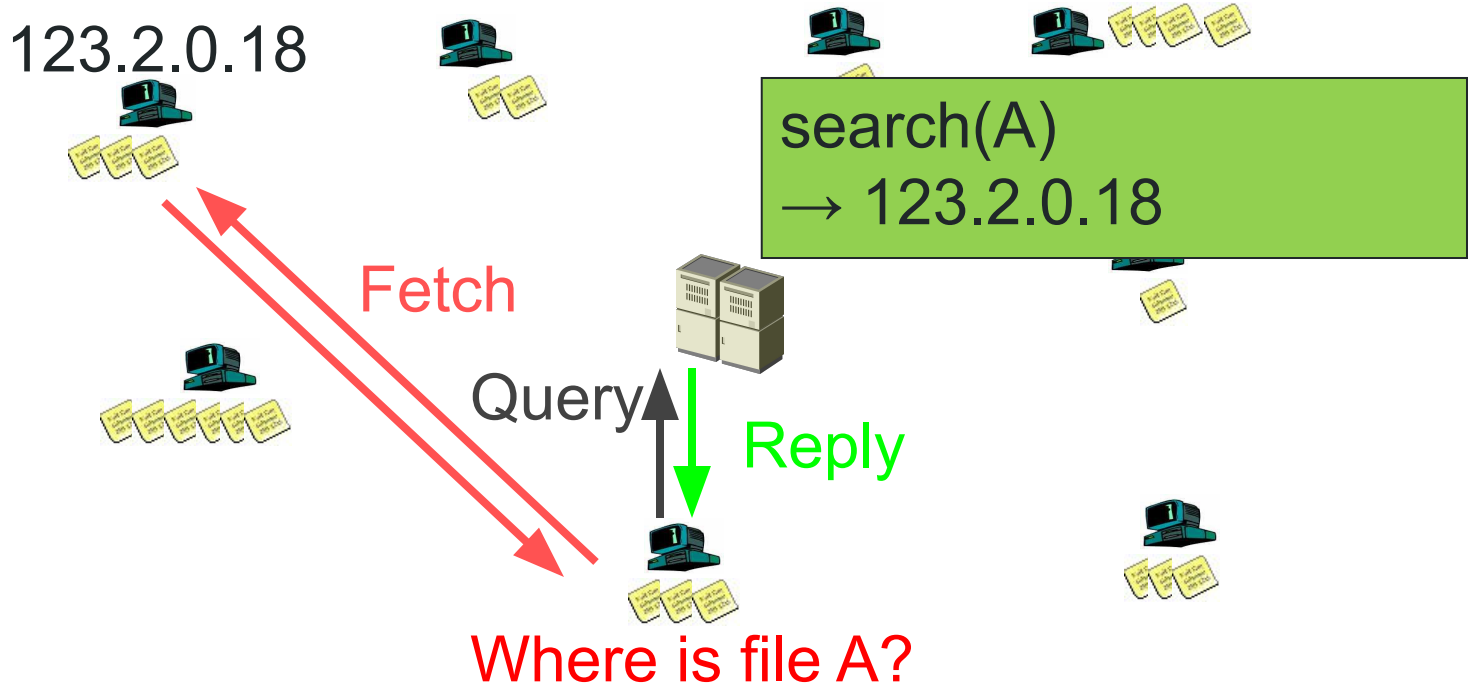
Disadvantages

- Poor scalability
- Single point of failure

Napster: publish!



Napster: search



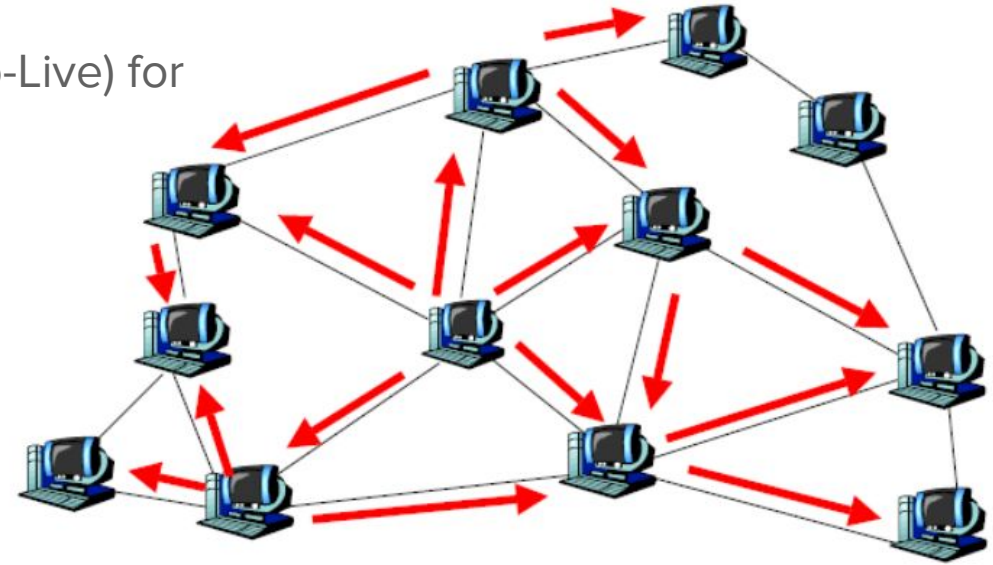

nutella
GRATIS
+75g
Dein nutella Foto-Kalender gratis!**
2 TAFEL-STICKER
FERRERO

Gnutella

Flooding-based lookup

- A node/peer connects to a set of Gnutella neighbors
- Forward queries to neighbors
- Flood network with TTL (Time-To-Live) for termination

Pro: Results are complete
Con: Bandwidth wastage



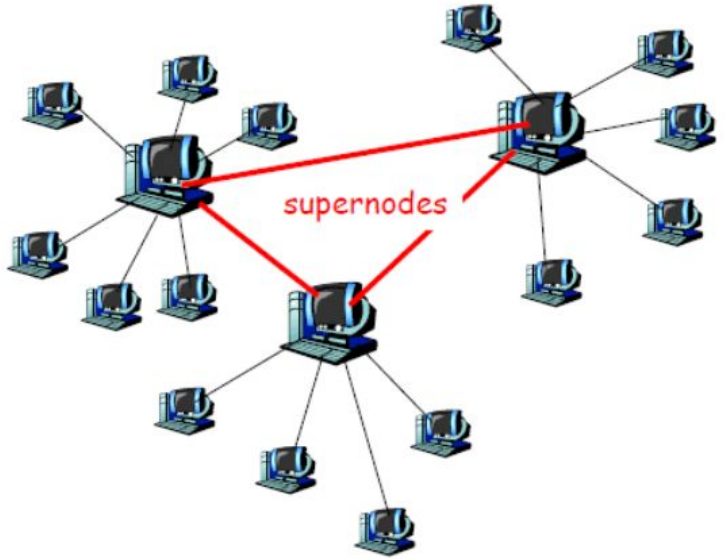
KaZaA

Hierarchical approach between Gnutella and Napster

Two-layered architecture.

- Powerful nodes (**supernodes**) act as local index servers
- Client queries are propagated to other supernodes

More efficient lookup than Gnutella and more scalable than Napster



KaZaA

Hybrid of centralized Napster and decentralized Gnutella

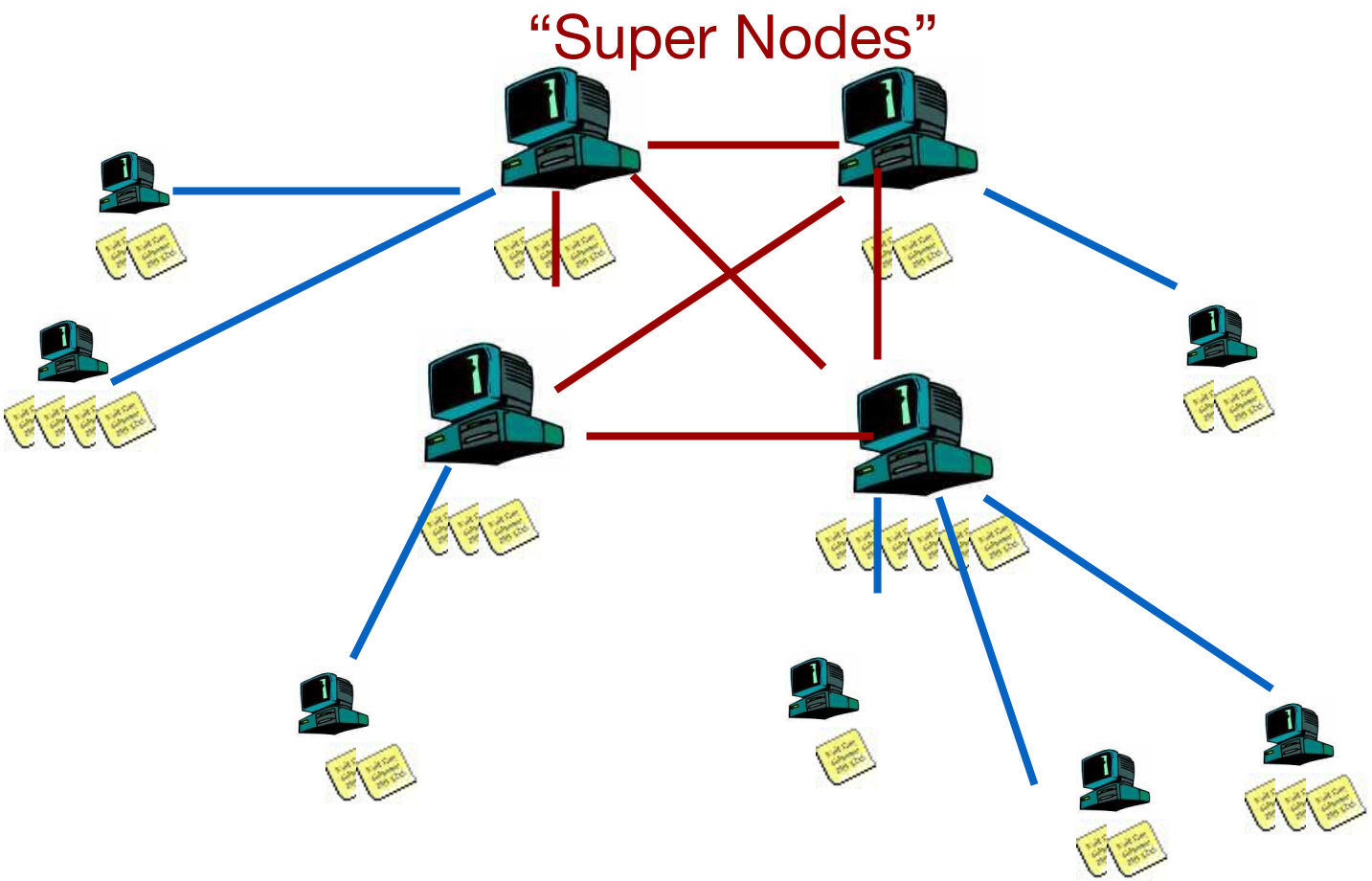
Super-peers act as local search hubs

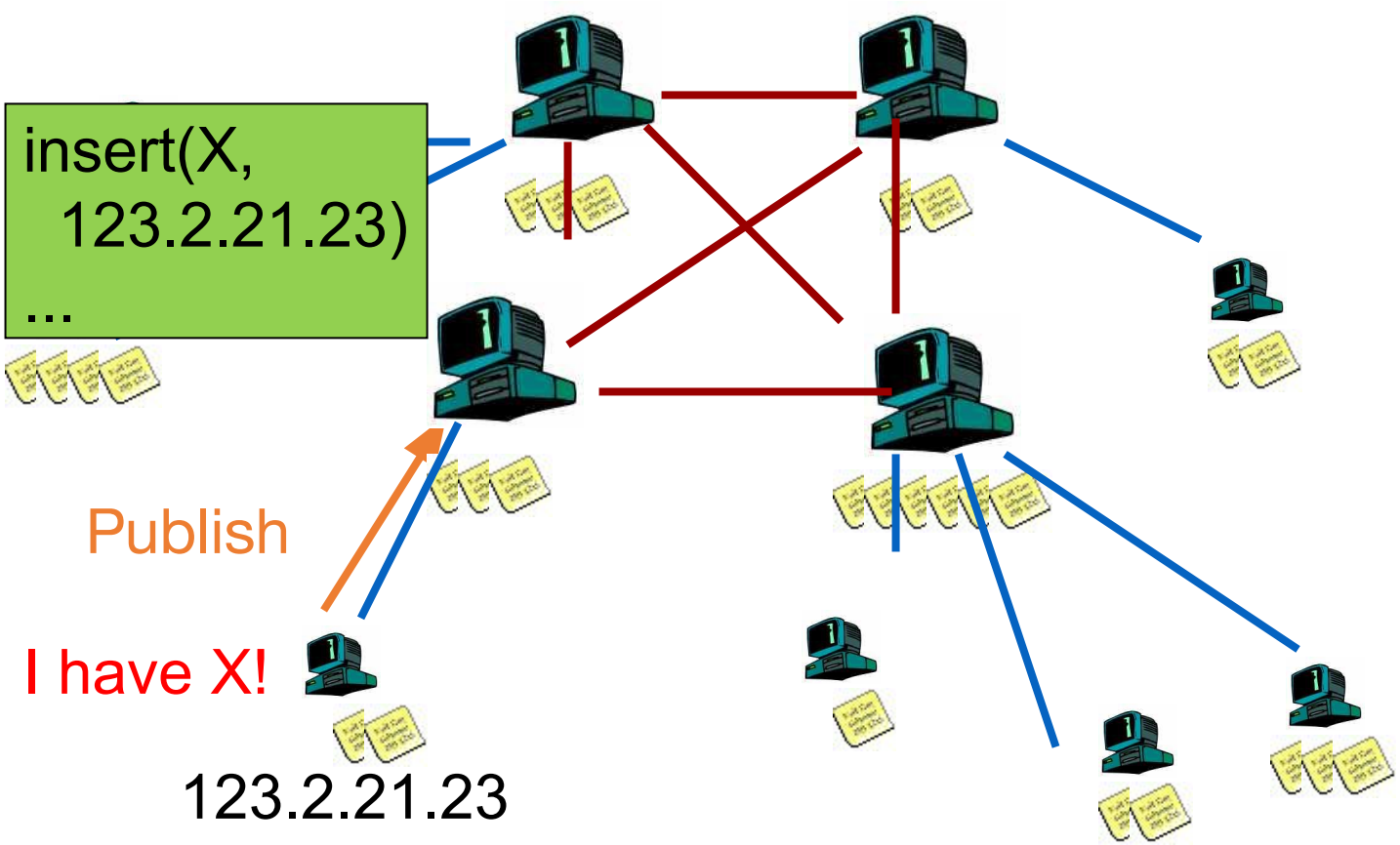
- Each super-peer is similar to a Napster server for a small portion of the network
- Super-peers are automatically chosen by the system based on their capacities (storage, bandwidth, etc.) and availability (connection time)

Users upload their list of files to a super-peer

- Super-peers periodically exchange file lists
- You send queries to a super-peer for files of interest
- The local super-peer may flood the queries to other super-peers for the files of interest, if it cannot satisfy the queries.

Exploit the heterogeneity of peer nodes


“Super Nodes”

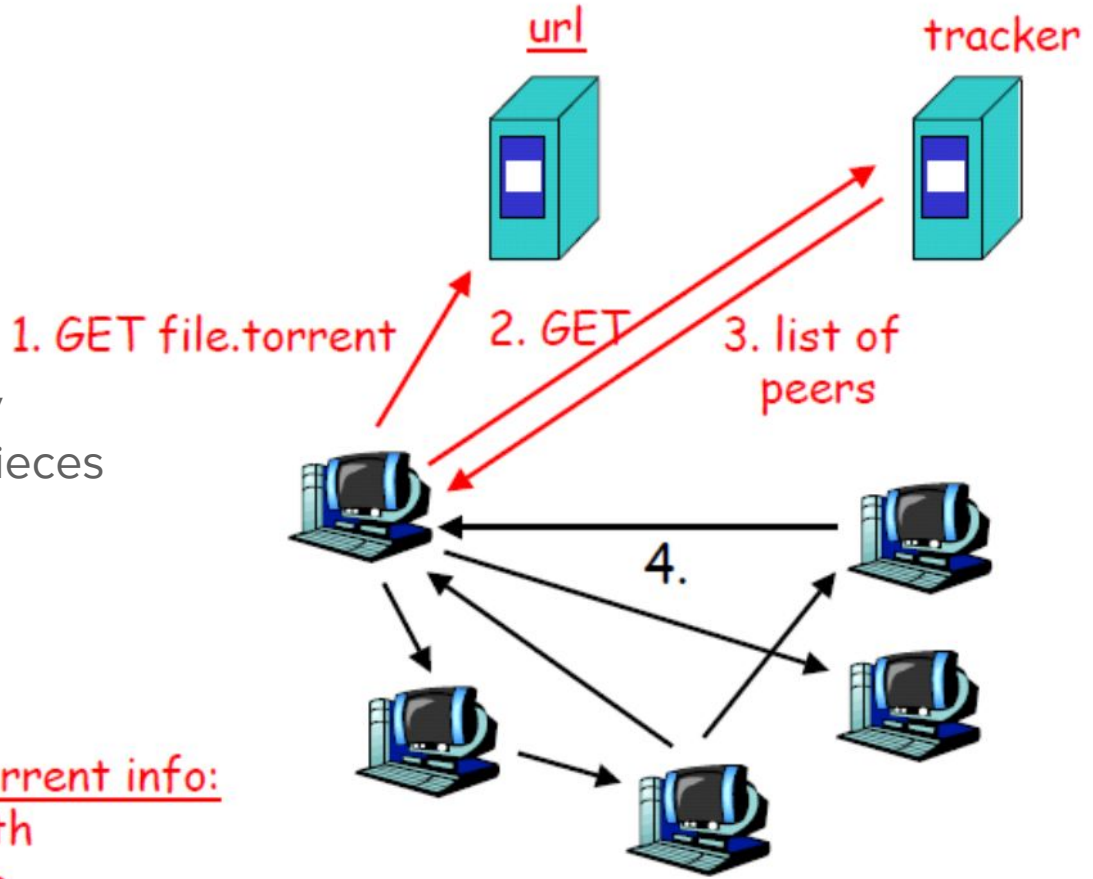

insert(X,
123.2.21.23)
...
Publish
I have X!
123.2.21.23

BitTorrent

Share large files efficiently
Maximize bandwidth utilization

Files downloaded non-sequentially
Typically broken down into small pieces
(256kb, 1mb, etc.)

Random or rarest-first downloads



Structured P2P

Structured P2P

Add routing to system
- No question about how to get to data
- Routing protocol governs lookup

Simple example:
- Each node gets an incrementing ID
- Route towards destination ID

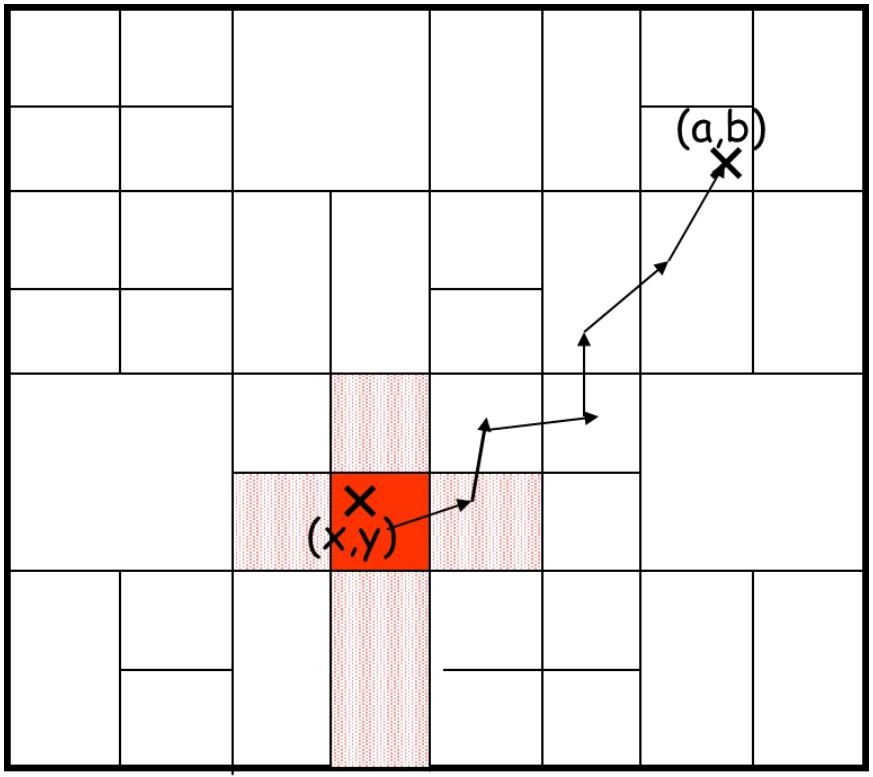
Multiple examples (Chord, Tapestry, CAN (content addressable network), ...)

Content-addressable networks (CAN)

Network represented in 2D space

Nodes "own" partition

Routing goes through neighbors until destination reached



https://sites.cs.ucsb.edu/~ravenben/talks/intro-SP2P.pdf

CAN

Node maintains state for neighboring node

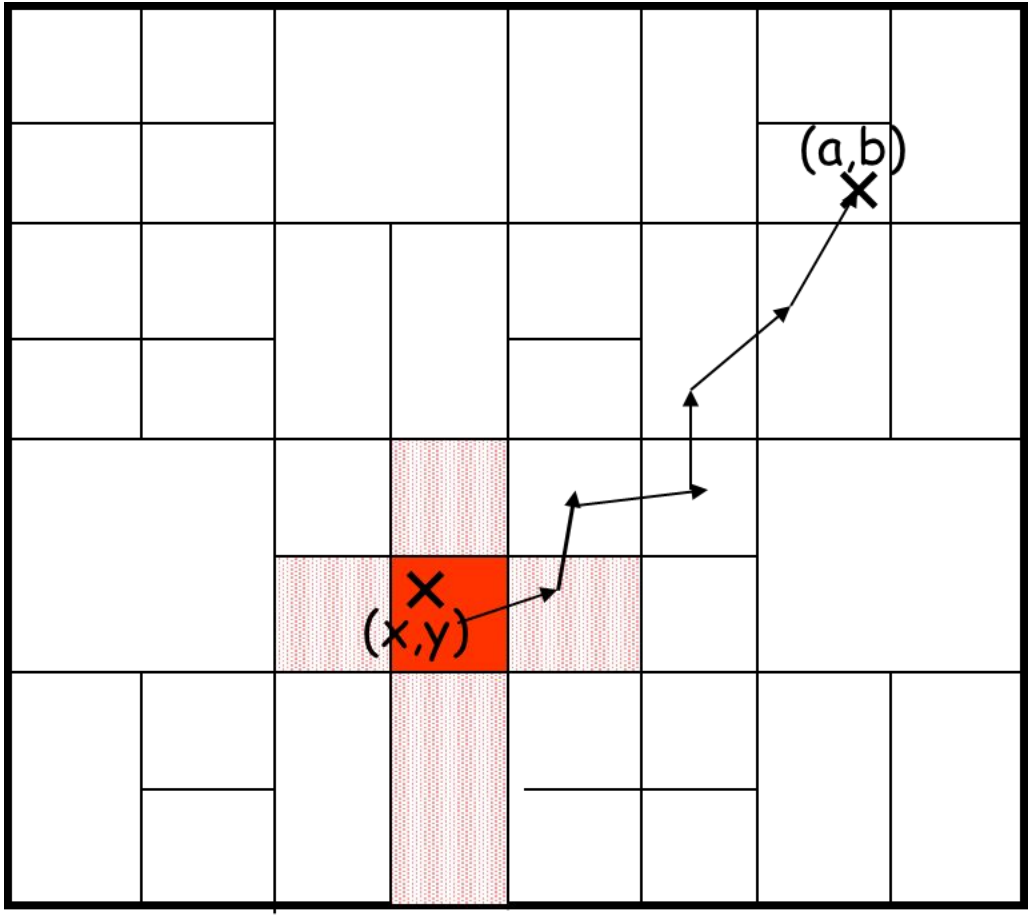
- O ($n^{1/d}$) lookup

Pros:

- Easily map multi-field data to namespace
- Low routing state

Cons:

- Static neighbors - no flexibility
- No network distance optimization



Structured P2P - What are protocols?

Definition of name-proximity
- Hops to get to destination ID

Size of routing table
- How much state tracked by nodes (i.e., network size)

Number of overlay hops
- Worst case performance to destination

Network locality
- Which neighbor do we pick?

Application interface
- How do we calculate IDs?

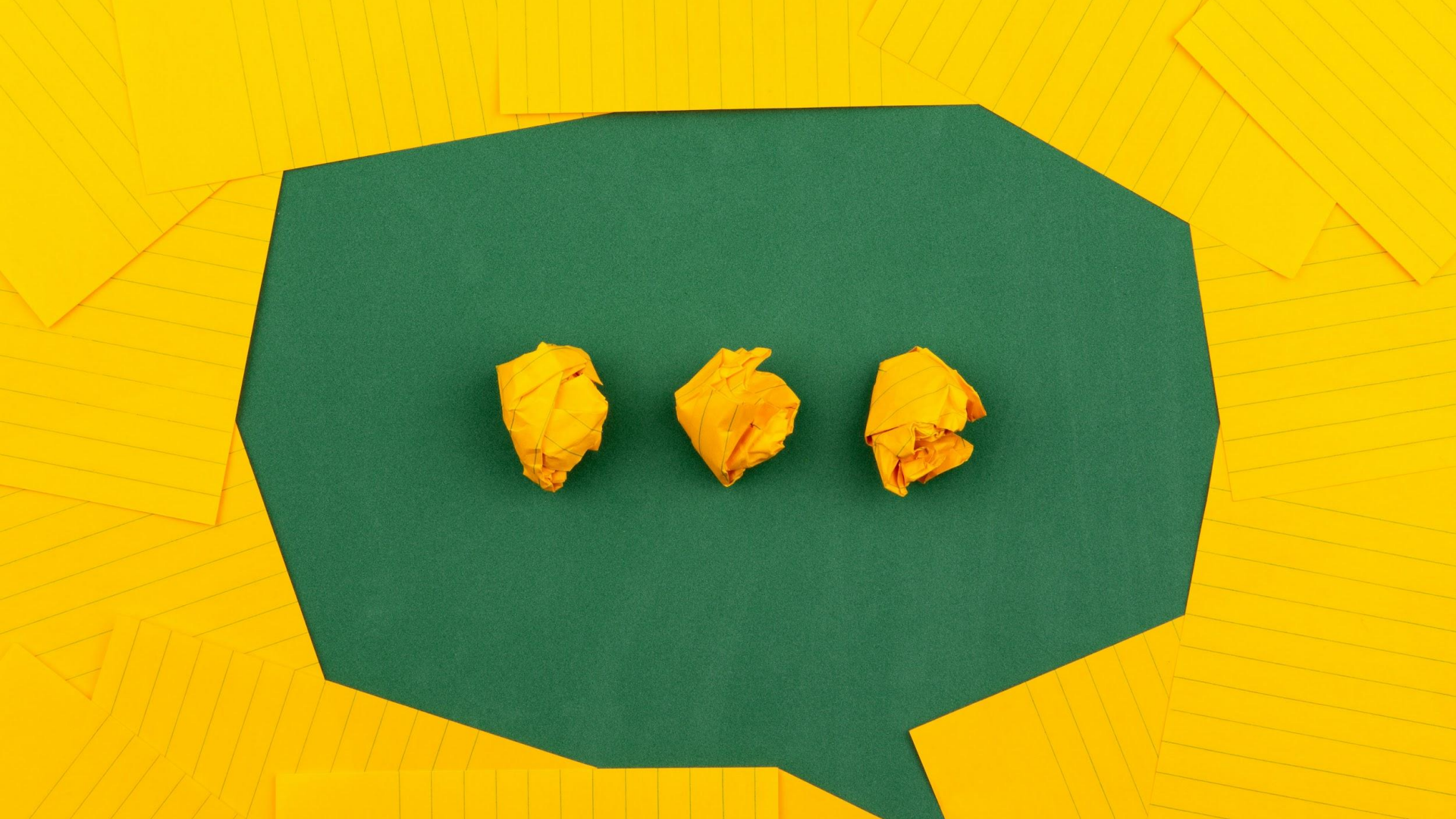
Structured P2P

Given a piece of data, the system knows exactly where it is

Each node should maintain its own local routing table to quickly find out the destination of the data

Not for file sharing

Usually for decentralized data storage in a data center



Discussion (1)

What types of applications would be appropriate for a client/server architecture?

And what types of applications would be appropriate for a P2P architecture?

- Client/server: applications which can be deferred, need a server -- you wouldn’t want communication to fail because a peer was turned off
- P2P: Immediate applications can be done peer-to-peer -- if a client is turned off, other peers can take over. Also avoids server bottlenecks.

-

Discussion (2)

Based on the introduction of P2P, do you think P2P can work well without the concept of server?

In other words, do you think P2P can work well only with clients?

- No. Somehow at least, we still need some kind of a server to set up an initial connection for clients.

Discussion (3)

What kind of legal issue could be there for P2P system? And why?

- P2P system may violate copyright laws. Different from C/S model, when a peer is downloading, it may also be uploading at the same time.

Lab