



ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN  
Vietnam - Korea University of Information and Communication Technology

# BÀI GIẢNG LẬP TRÌNH MẠNG

**PGS.TS.Huỳnh Công Pháp; Nguyễn Anh Tuấn; Lê Tân;  
Nguyễn Thanh Cẩm; Hoàng Hữu Đức**

Khoa Khoa học máy tính



## **Bài 3. Các mô hình ứng dụng mạng**



## Mô hình ứng dụng client/server



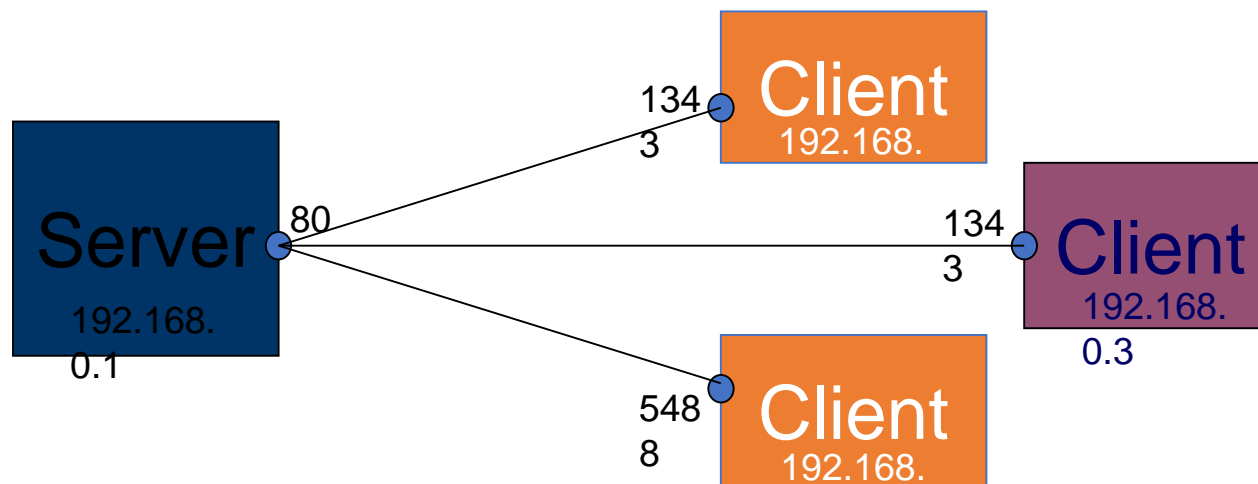
## Tổng quan

- Mô hình mạng cơ bản nhất hiện nay.
- Dạng phổ biến của mô hình ứng dụng phân tán.
- Đa số các ứng dụng mạng dựa theo mô hình này.
- Thuật ngữ client/server xuất hiện vào đầu thập niên 80.
- Một số ứng dụng client/server phổ biến
  - Email
  - FTP
  - Web



# Thành phần

- Một tiến trình Server và một hoặc nhiều tiến trình client.



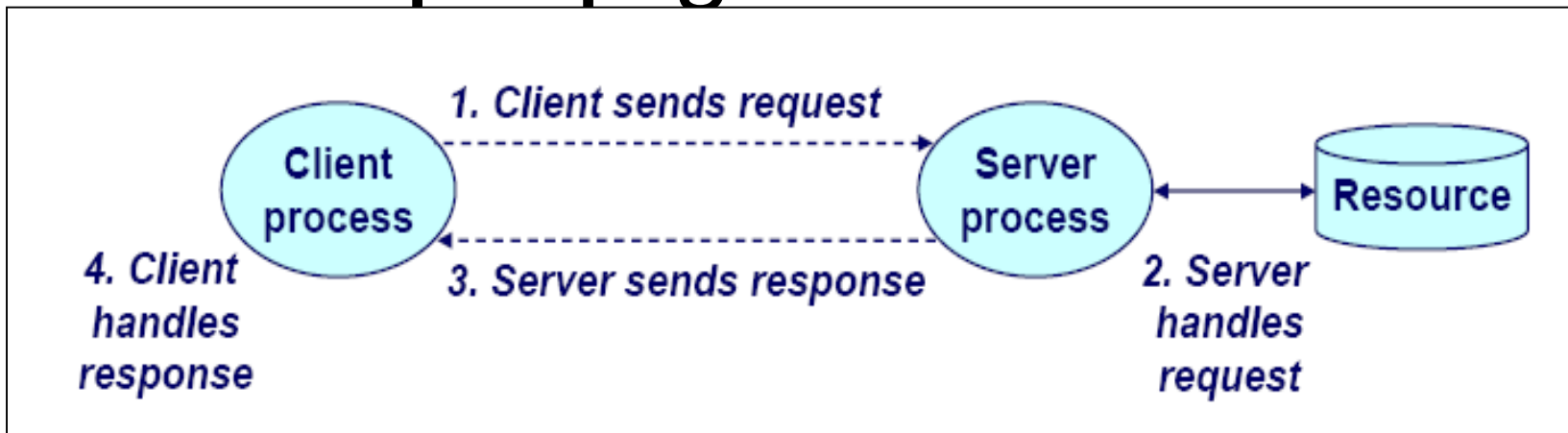
- Các tiến trình clients và servers có thể chạy trên cùng trạm (host) hoặc khác trạm.
- Là các đối tượng logic tách biệt và liên lạc với nhau qua mạng cùng thực hiện một công việc.



## Chức năng từng thành phần

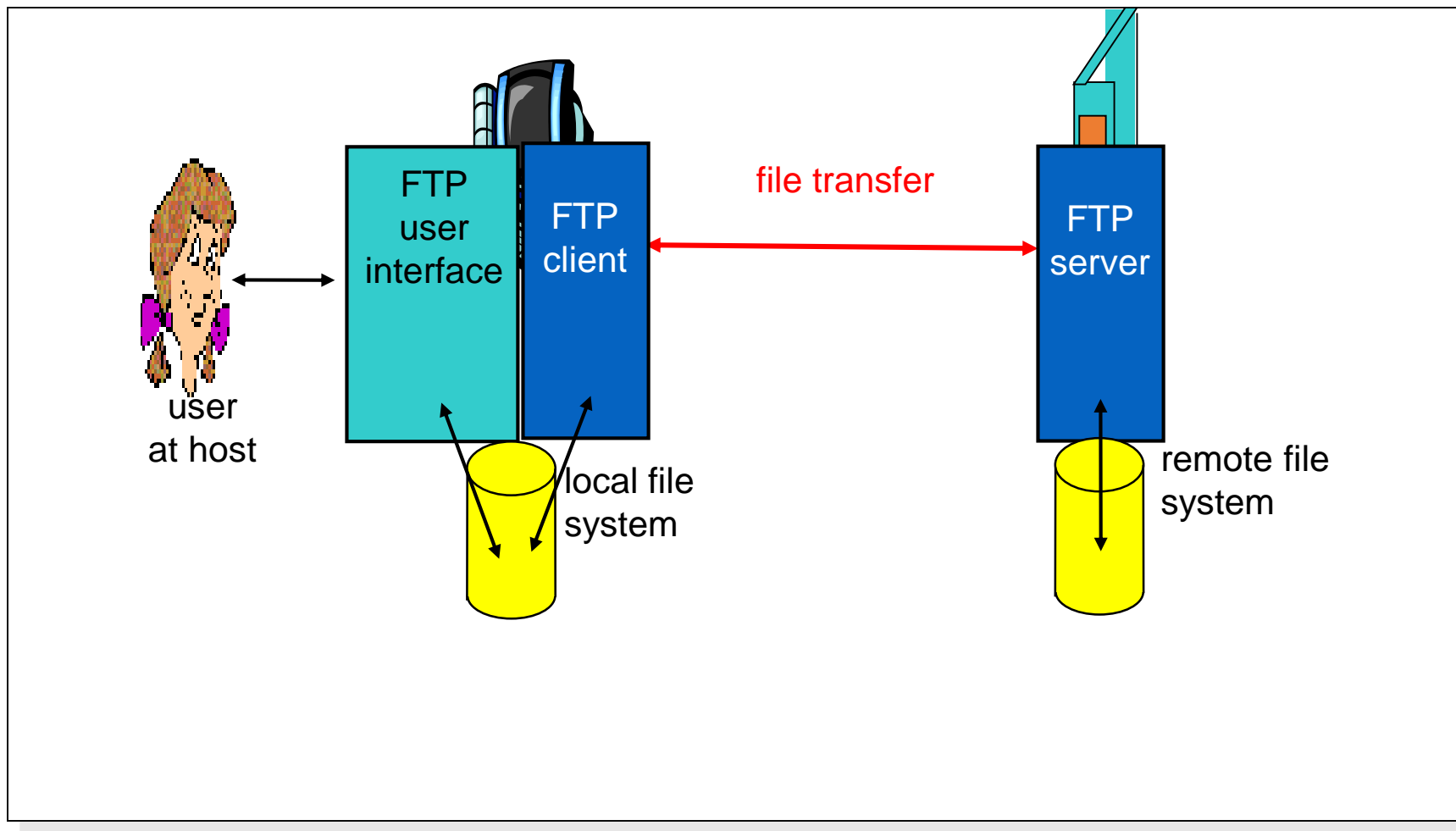
- Server
  - Quản lý nguồn tài nguyên nào đó.
  - Cung cấp dịch vụ và phân phối tài nguyên.
- Client
  - Chương trình giao tiếp với người sử dụng,
  - Cần yêu cầu về tài nguyên
- Một tiến trình có thể vừa là server vừa là client.

# Cách hoạt động



- Client:
  - Khởi tạo liên lạc với server (“speaks first”)
  - Yêu cầu dịch vụ nào đó từ server,
  - Đối với Web, client được hiện thực trong browser,
- Server:
  - Cung cấp dịch vụ yêu cầu cho client
  - Chặn hạn, Web server gửi trang Web yêu cầu, hay mail server phân phát e-mail

# Ví dụ Ftp: The File Transfer Protocol







## Đặc trưng của mô hình ứng dụng client/server

- Hoạt động theo kiểu giao thức bất đối xứng
  - Thể hiện quan hệ một chiều giữa các client và một server.
  - Client bắt đầu phiên hội thoại bằng cách yêu cầu dịch vụ.
  - Server sẵn sàng chờ các yêu cầu từ client.



# Đặc trưng của mô hình ươd client/server (tt)

- Đóng gói dịch vụ
  - Server như một chuyên gia,
    - Hoàn thành tác vụ đáp ứng lại các yêu cầu từ client.
  - Server có thể được nâng cấp mà không ảnh hưởng đến client
- Tính toàn vẹn
  - Mã và dữ liệu đối với một server được bảo trì tập trung
    - Giảm chi phí và bảo vệ sự toàn vẹn của dữ liệu chung.
  - Trong khi đó, client duy trì tính cá nhân và độc lập.



# Đặc trưng của mô hình ươ client/server (tt)

- Trong suốt định vị
  - Server và client có thể trên một máy hoặc khác.
- Tính modun
  - Một ứng dụng client/server có thể thiết kế thành nhiều modun.
    - Chia để trị
    - Khả năng chịu lỗi (lỗi xảy ra tại modun nào đó không làm sập toàn bộ hệ thống)
- Tính không phụ thuộc nền
  - Một ươ client/server lý tưởng là độc lập nền
    - phần cứng, hệ điều hành



## Đặc trưng của mô hình ươ client/server (tt)

- Tính sử dụng lại mã
  - Các đoạn mã trong tiến trình server phải được sử dụng trong các tiến trình server khác.
- Tách biệt chức năng client và server
  - 2 bên phải đảm nhận chức năng riêng
- Chia sẻ tài nguyên
  - Một server có thể chia sẻ tài nguyên cho nhiều client cùng một lúc.
  - Server cũng có khả năng điều phối truy nhập của các client đến các tài nguyên dùng chung.



# Ưu điểm của mô hình ươd client/server

- Tính tập trung (Centralization)
  - Truy cập tài nguyên và bảo mật dữ liệu
    - tập trung thông qua server
- Tính co giãn (Scalability)
  - Nâng cấp bất cứ thành phần nào khi cần thiết
- Tính mềm dẻo (Flexibility)
  - Công nghệ mới có thể dễ dàng tích hợp vào hệ thống
- Tính trao đổi tương tác (Interoperability)
  - Tất cả các thành phần (clients, mạng, servers) cùng nhau làm việc.



# Nhược điểm của mô hình ươd client/server

- Quản trị hệ thống khó khăn
  - Duy trì thông tin cấu hình luôn cập nhật và nhất quán giữa tất cả các thiết bị.
- Nâng cấp phiên bản mới khó đồng bộ.
- Phụ thuộc **độ tin cậy của mạng**
- Chi phí thiết kế, cài đặt, quản trị và bảo trì rất lớn.
- Phải giải quyết
  - sự xung đột trong hệ thống,
  - tính tương thích của các thành phần,
  - việc cấu hình hệ thống.



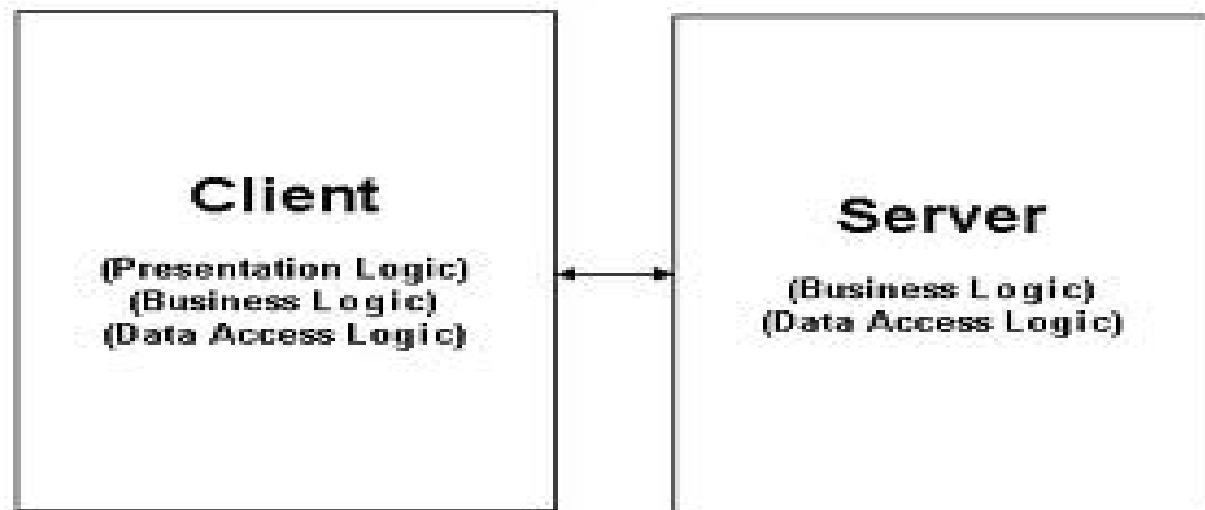
## Sự phân lớp trong mô hình ứng dụng client/server

- Mọi ứng dụng mạng client/server đều có 3 khối chức năng:
  - Khối logic biểu diễn hay giao diện người dùng
  - Khối logic nghiệp vụ (business logic)
  - Khối dữ liệu (cơ sở dữ liệu lưu trữ)



# Kiến trúc ươd client/server 2 lớp (2-tier)

- Khối nghiệp vụ (business logic)
  - Được đặt bên trong lớp giao diện người dùng tại client hoặc
  - Được đặt bên trong lớp csdl.
  - Cũng có thể được chia ra đặt tại cả client và server.



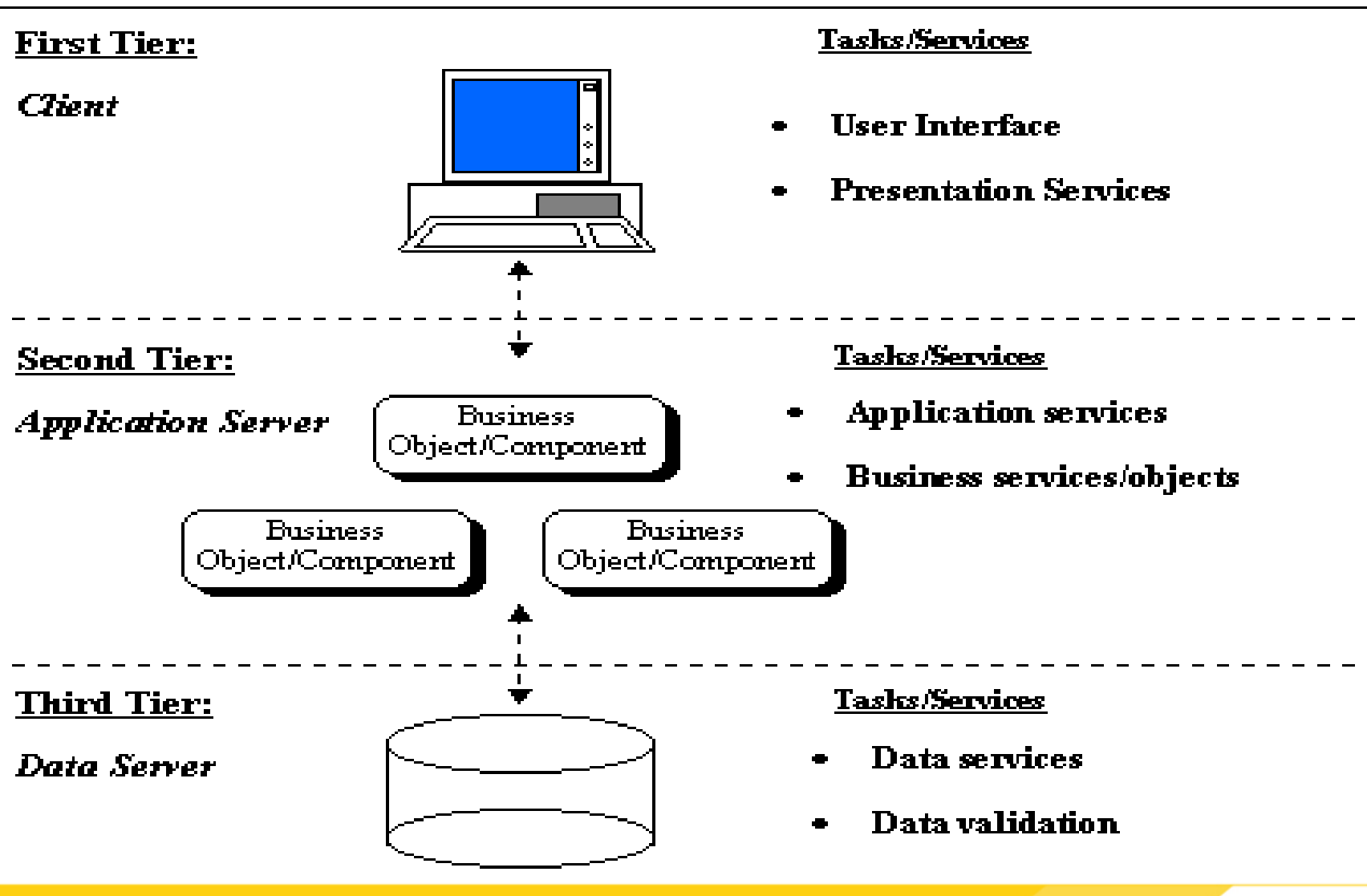




# Kiến trúc ươd client/server 3 lớp (3-tier)

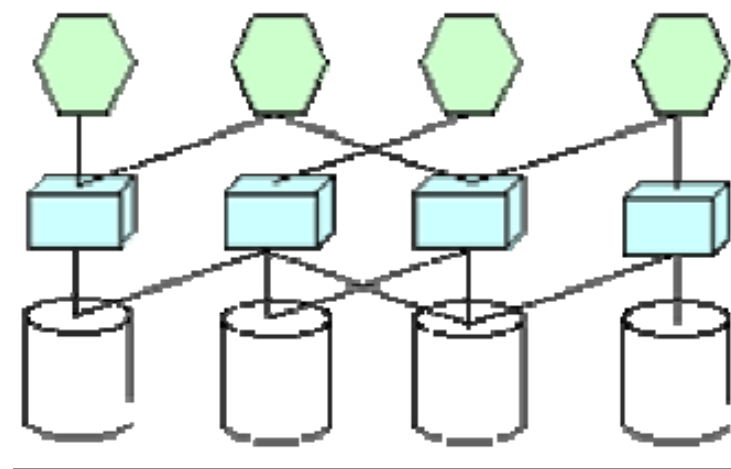
- Phát triển vào thập niên 90'
  - Khi nhu cầu đối với những hệ thống lớn hơn và ổn định hơn.
- Mở rộng từ mô hình 2 lớp: Tách biệt khối nghiệp vụ
  - Nâng cao hiệu năng (performance),
  - Tính linh hoạt (flexibility),
  - Khả năng bảo trì (maintainability),
  - Khả năng dùng lại (reusability)

# Kiến trúc ươ client/server 3 lớp (3-tier)



# Kiến trúc ươd client/server 3 lớp (3-tier)

- Lớp trên cùng
  - Chứa giao diện dịch vụ cho người dùng
- Lớp dưới cùng
  - Chứa chức năng quản trị csdl
- Lớp trung gian (lớp thứ 3)
  - Chứa khối nghiệp vụ
  - Các tiến trình xử lý
  - Điều khiển các giao dịch và các hàng đợi.
  - Gửi các yêu cầu từ client đến server csdl,
  - Được xem như proxy server.



Mô hình 3 tầng



# Ví dụ: ứng dụng client/server 3 lớp

- Ứng dụng Web
  - Lớp trên cùng: Web browser
  - Lớp trung gian: Web server engine (tomcat..), các trang Web (JSP,..)
  - Lớp dưới cùng: Hệ csdl
- Ứng dụng Struts hoặc JSF (trong Java)
  - Lớp trên cùng: Views
  - Lớp trung gian: Controllers
  - Lớp dưới cùng: Models
- Kiến trúc client/server n lớp (n-tier)
  - Lớp trung gian được chia thành nhiều đơn vị nhỏ



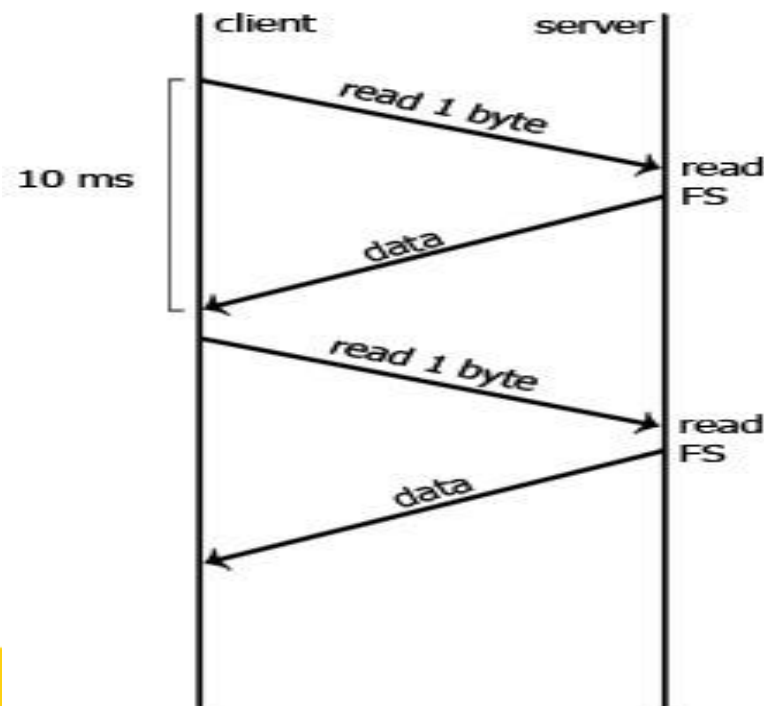
# Giao thức cho ứng dụng client/server

- Giao thức?
  - Là tập các khuôn dạng bản tin, tập các trạng thái, qui tắc, quy ước trong truyền thông giữa client và server
- Khi tạo một ứng dụng client/server,
  - Phải thiết kế giao thức.
  - Các giao thức phổ biến FTP, HTTP, SMTP, RPC,...
    - đã được IETF (Internet Engineering Task Force) chuẩn hóa thành các giao thức chuẩn.
- Lập trình csdl
  - Giao thức trao đổi các câu truy vấn
    - Được hỗ trợ và qui định bởi hệ quản trị csdl và các thư viện lập trình.

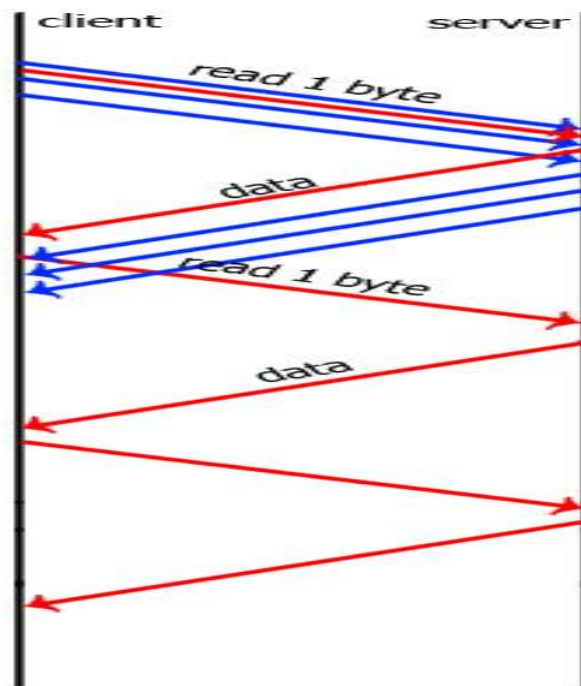


# Phân loại giao thức

- Giao thức đồng bộ (Synchronous protocol)
  - Truyền thông giữa client và server diễn ra theo 2 chiều nhưng không đồng thời.
  - Được thực hiện lần lượt.
  - Các giao thức kiểu này là : HTTP, SMTP, POP3...



- Client và server có thể đồng thời gửi thông tin.
- Các giao thức này như TELNET, RLOGIN,...



- Ngoài ra, còn có loại giao thức hybrid kết hợp giữa 2 giao thức trên



# Thảo luận

- 2 SV lập thành một nhóm để thảo luận cách xây dựng ứng dụng client/server nào đó mà GV yêu cầu:
  - Sơ đồ hoạt động
  - Các chức năng phía server
  - Các chức năng phía client
  - Các tình huống
  - Các thuật toán
  - Cách tổ chức và quản lý dữ liệu
  - ...





## Mô hình ứng dụng P2P



# P2P- Mạng ngang hàng?

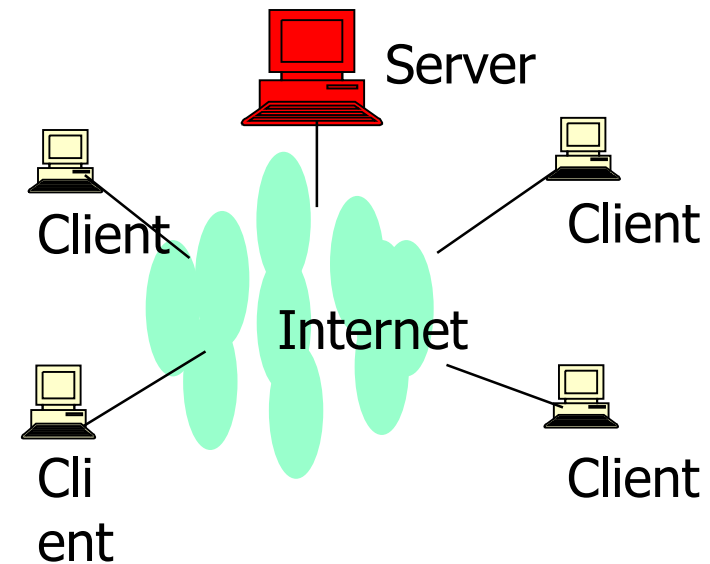
- Hoạt động của mạng chủ yếu dựa vào
  - khả năng tính toán và băng thông của các máy tham gia
  - không tập trung vào một số nhỏ các máy chủ trung tâm
- Một Peer?
  - “...một thực thể có khả năng tương tự như các thực thể khác trong hệ thống mạng”



# Tại sao P2P?

- Mô hình client/server
  - Server là nguồn cung cấp tài nguyên
  - Client yêu cầu dữ liệu từ server
  - Rất thành công (www, FTP, Web services,...)
- Tuy nhiên, hạn chế
  - Khả năng mở rộng
  - Hoạt động phụ thuộc vào server
  - Yêu cầu quản trị
  - Không tận dụng tối đa tài nguyên mạng

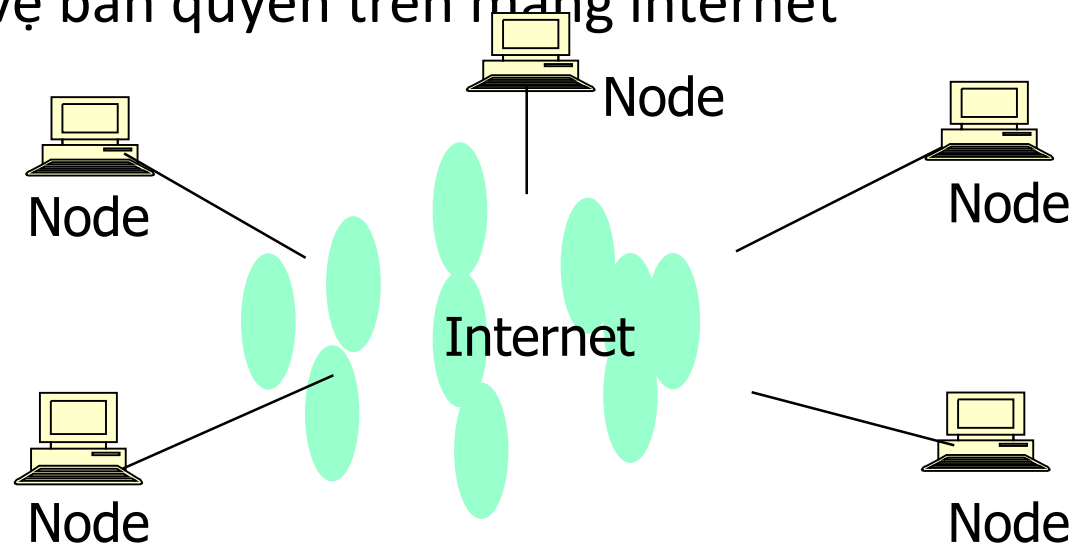
→ Các hệ thống P2P khắc phục các hạn chế trên





# Kiến trúc P2P

- Mỗi node vừa là client vừa là server
  - Cung cấp và tiêu thụ dữ liệu
  - Có thể khởi tạo kết nối
- Dữ liệu không tập trung
  - Vấn đề về bảo vệ bản quyền trên mạng internet





# Đặc tính của P2P

- Clients cũng là servers và routers
  - Các nút đều đóng góp băng thông, tài nguyên, bộ nhớ, CPU,...
- Các nút đều độc lập
- Mạng không cố định
  - nodes tham gia hoặc ra khỏi mạng thường xuyên
- Các nút tự cộng tác trực tiếp với nhau
- Các nút có khả năng khác nhau



# Lợi ích của P2P

- Sử dụng hiệu quả nguồn tài nguyên
  - Băng thông, không gian lưu trữ, sức mạnh xử lý
- Khả năng mở rộng
  - Không có hiện tượng “thắt cổ chai” (truyền thông, tính toán, lưu trữ)
  - Càng có nhiều người sử dụng, khả năng của mạng càng tăng
- Tin cậy
  - Được sao lưu
  - Phân phối theo địa lý
  - Không phụ thuộc vào điểm nào trong mạng



## Ưu/nhược của P2P

Ưu điểm	Nhược điểm
Mạng P2P cài đặt dễ và đơn giản, chỉ yêu cầu một Hub hay một Switch để kết nối tất cả máy tính với nhau	Vấn đề về bản quyền truy cập tài nguyên
Người dùng có thể truy cập bất kỳ file nào trên máy tính miễn là file đó được đặt trong thư mục chia sẻ.	Hoạt động không hiệu quả khi số lượng kết nối ít
Topology đơn giản	Các máy tính tham gia có thể thoát khỏi mạng một cách đột ngột
Nếu một máy tính lỗi, các máy tính còn lại vẫn làm việc	



# Một số ứng dụng P2P

- Chia sẻ File (Napster, Gnutella, Kazaa,...)
- Multiplayer games (Unreal Tournament, DOOM)
- Ứng dụng cộng tác (ICQ, shared whiteboard)
- Tính toán phân tán (Seti@home)
- Mạng Ad-hoc





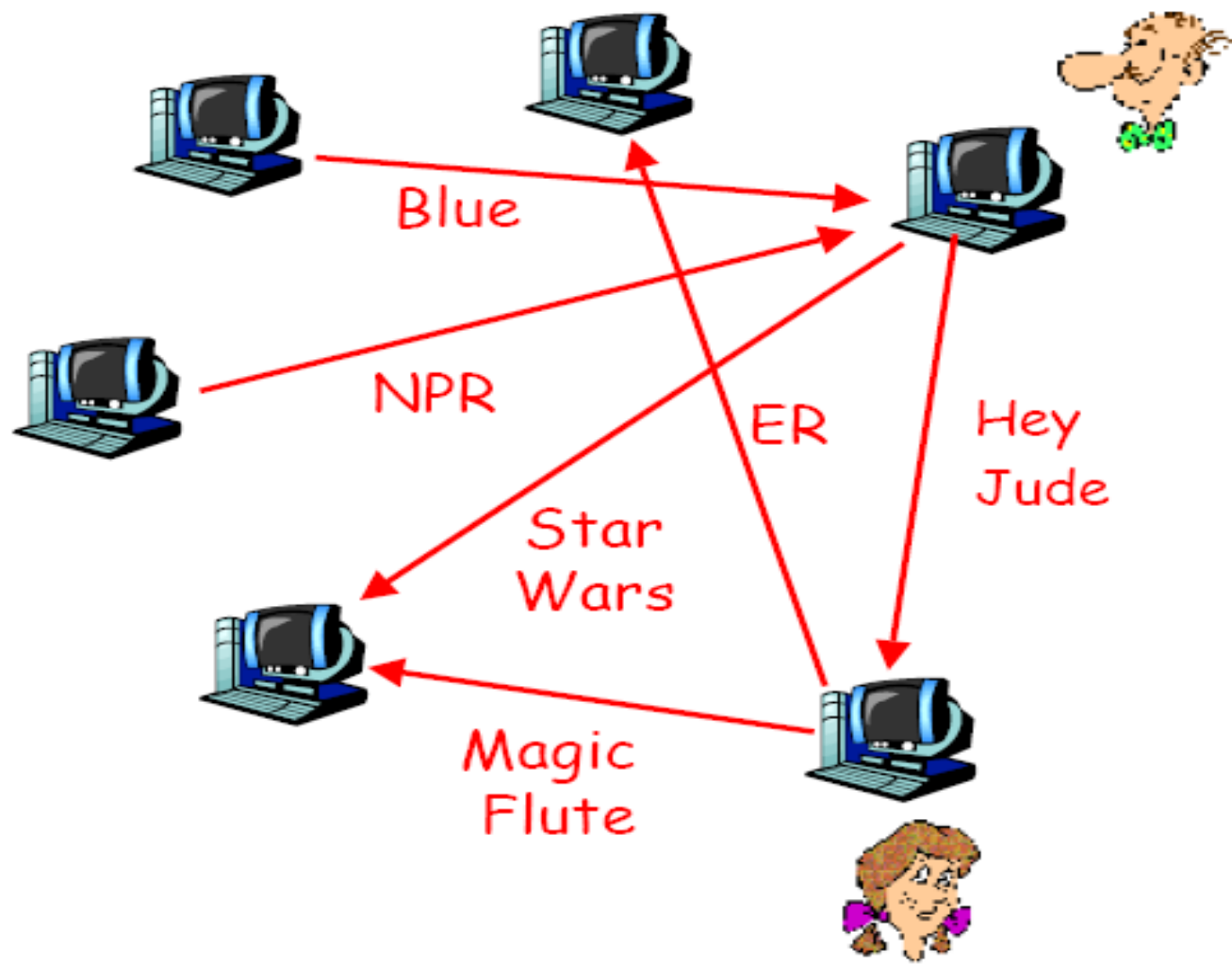
# Phân loại mạng P2P

- Mạng P2P thuần túy:
  - Các máy trạm có vai trò vừa là máy chủ vừa là máy khách
  - Không có máy chủ trung tâm quản lý mạng
  - Không có máy định tuyến (bộ định tuyến) trung tâm, các máy trạm có khả năng tự định tuyến
- Mạng P2P lai:
  - Có một máy chủ trung tâm dùng để lưu trữ thông tin của các máy trạm và trả lời các truy vấn thông tin này.
  - Các máy trạm có vai trò lưu trữ thông tin, tài nguyên được chia sẻ, cung cấp các thông tin về chia sẻ tài nguyên của nó cho máy chủ.
  - Sử dụng các trạm định tuyến để xác định địa chỉ IP của các máy trạm.



# Cơ chế làm việc P2P chia sẻ file

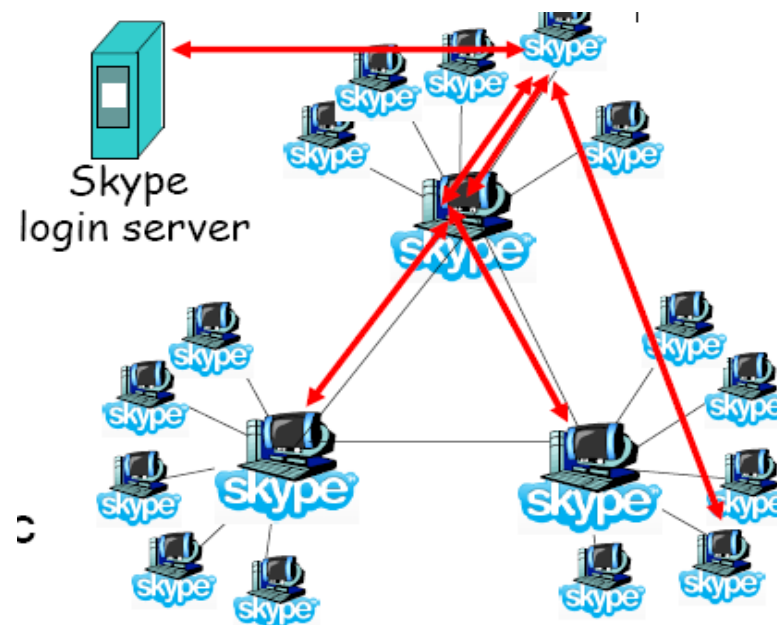
- A chạy một ứng dụng client chia sẻ file trên máy tính đã kết nối vào Internet
  - Tìm kiếm một file nào đó
  - Ứng dụng hiển thị các máy (peer) có chứa một bản sao file đó
  - A chọn một máy trong số đó, B.
- File được truyền từ máy B đến máy A
  - Khi máy A đang download file, các máy khác có thể copy từ máy A
- A vừa là client, vừa là server



# Ứng dụng chat P2P

- A chạy ứng dụng chat client trên máy có kết nối internet
  - Ứng dụng hiển thị danh sách friends có chứa B
  - A khởi tạo một kết nối TCP trực tiếp đến B
  - A và B bắt đầu chat với nhau trực tiếp

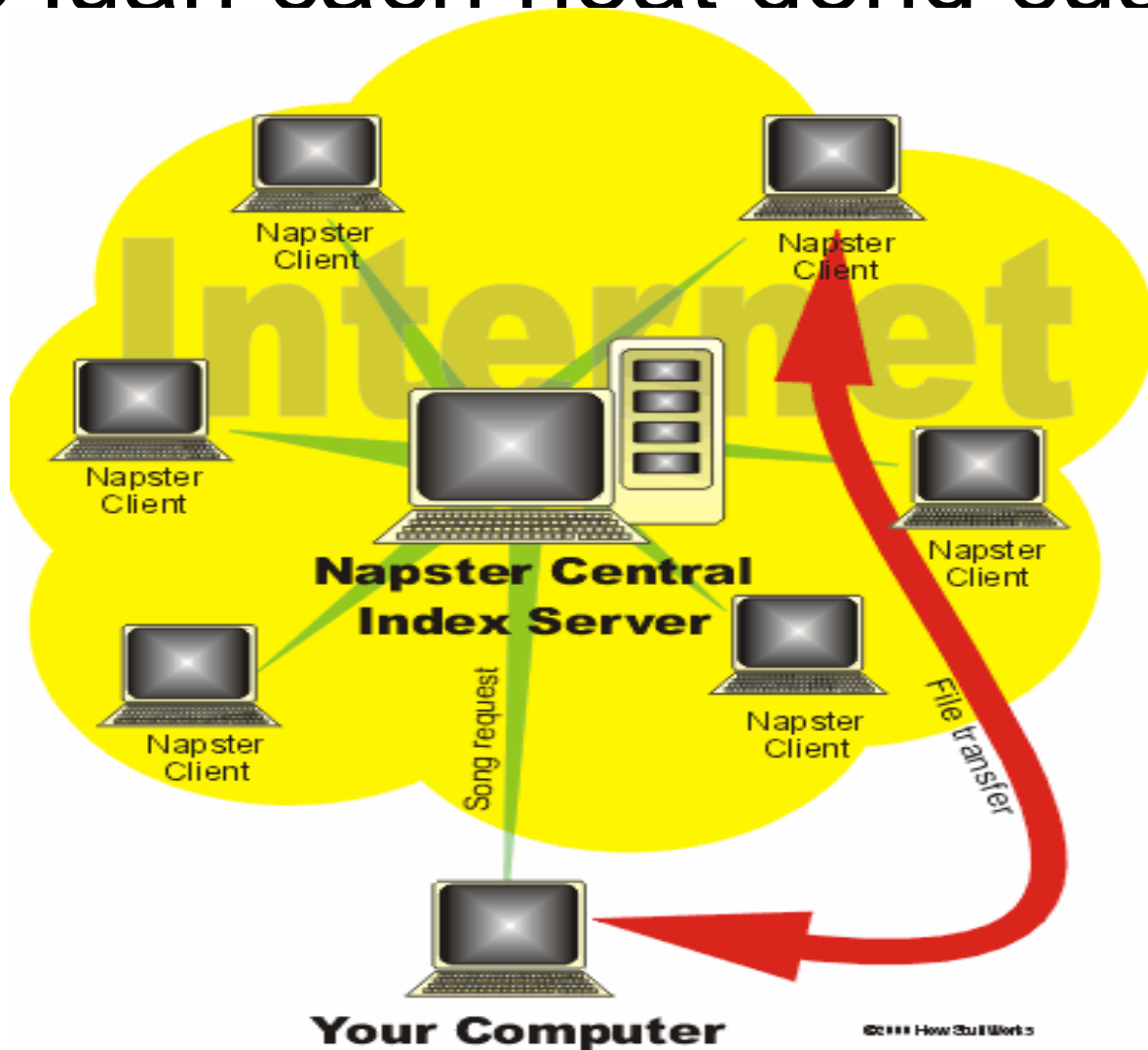
**Skype là hệ thống VoIP P2P**





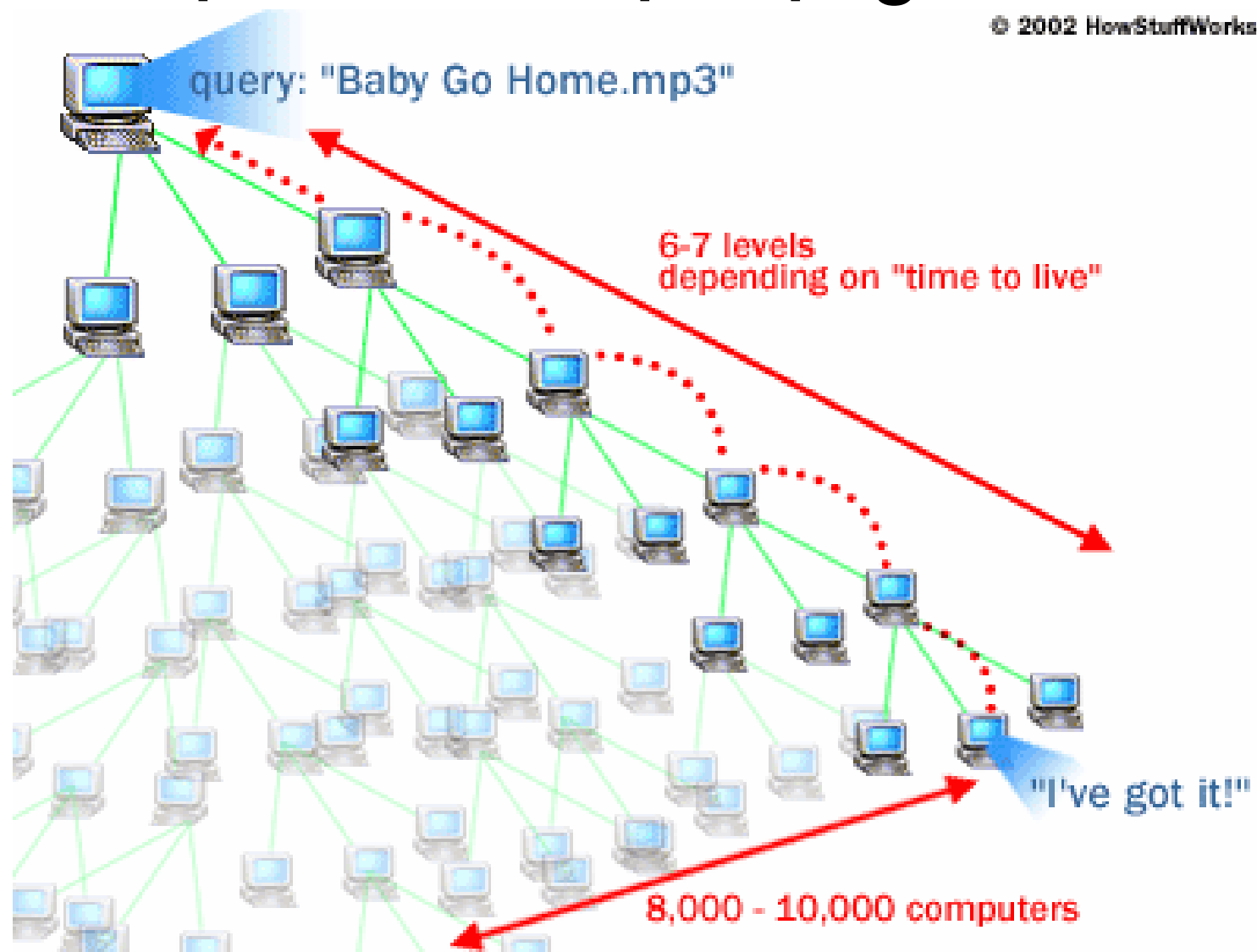
- Mô hình phân tán

# Thảo luận cách hoạt động của Napster

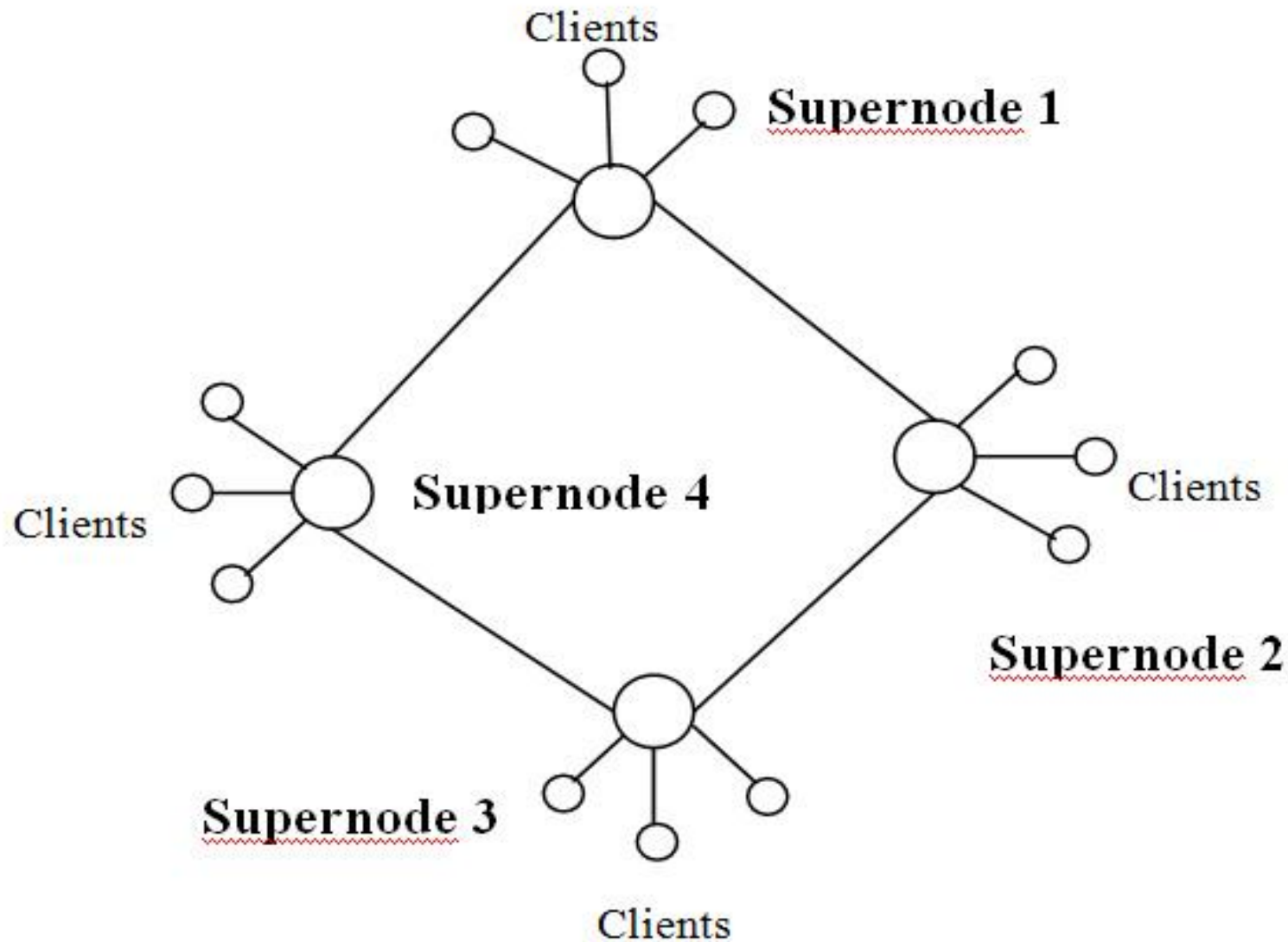


# Thảo luận cách hoạt động của Gnutella

© 2002 HowStuffWorks



# Thảo luận cách hoạt động của Kazaa







*Thank your listenning*