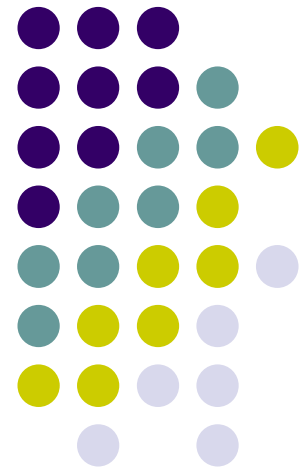


Ứng dụng P2P

cuu duong than cong . com

cuu duong than cong . com





Chia sẻ file ngang hàng

Ví dụ

- A chạy một ứng dụng chia sẻ file trên máy tính
- kết nối vào Internet
- tìm kiếm bài hát “Hello”
- Ứng dụng hiển thị các máy (peer) có chứa một copy của bài hát đó.
- A chọn một máy trong số đó, B.
- File được truyền từ máy B đến máy A: HTTP
- khi A đang load file, các máy khác có thể copy từ A
- A vừa là client, vừa là server.

Tất cả các máy có thể là server = Tính mở rộng cao!

P2P: Thư mục tập trung

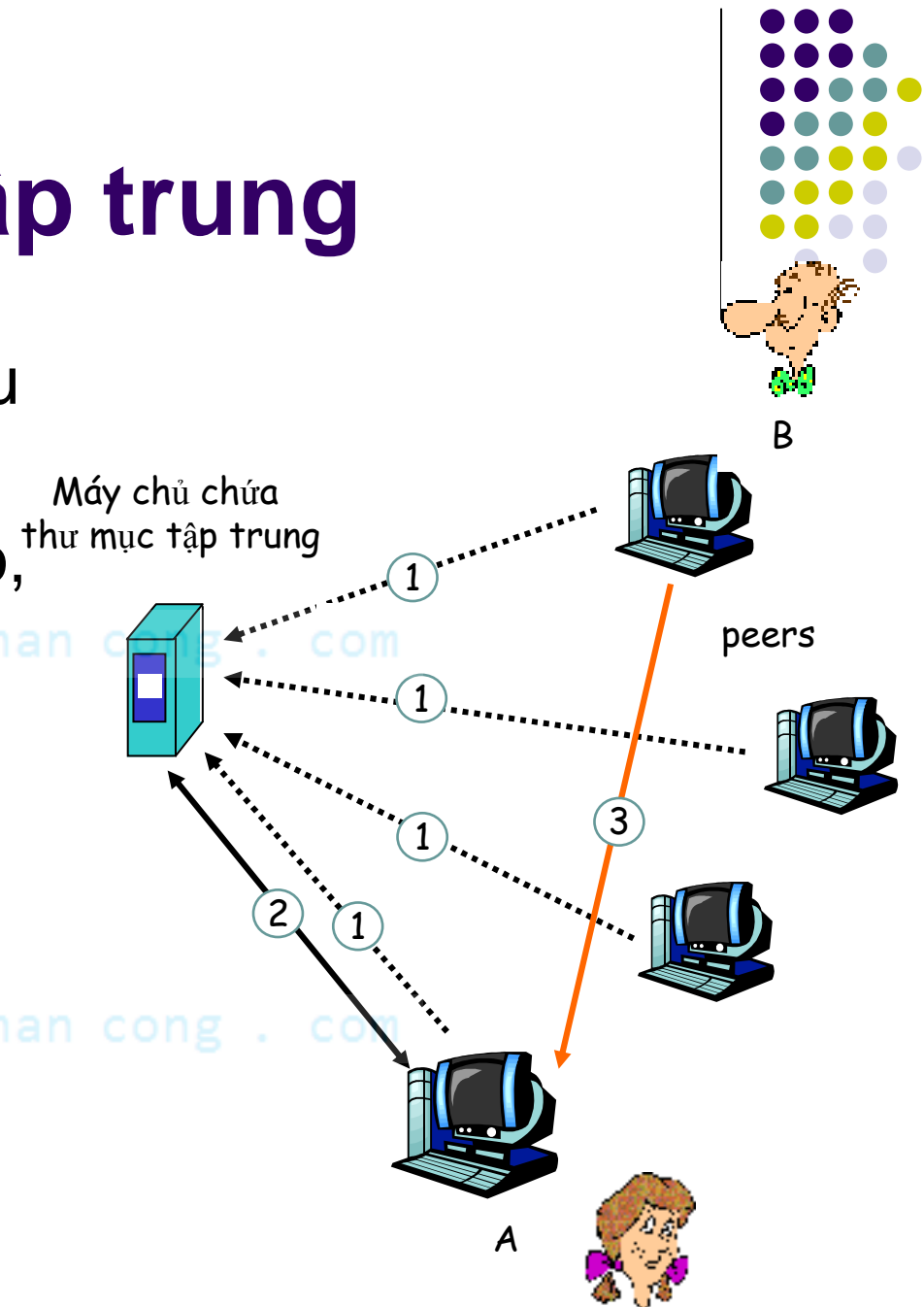
Phiên bản gốc thiết kế kiểu
“Napster”

1) Khi một máy kết nối vào,
nó sẽ khai báo:

- IP address
- nội dung

2) A tìm kiếm “Hello”

3) A yêu cầu file từ B



P2P: Hạn chế của thư mục tập trung



- Tính chịu lỗi
- Hiệu năng
- Bản quyền: sẽ bị truy tố

Chia sẻ file là phân tán song thông tin về nội dung là tập trung

cuu duong than cong . com

cuu duong than cong . com



Query flooding: Gnutella

- hoàn toàn phân tán
 - không có máy chủ tập trung
- giao thức public
- có nhiều phiên bản client Gnutella cùng cài đặt giao thức này

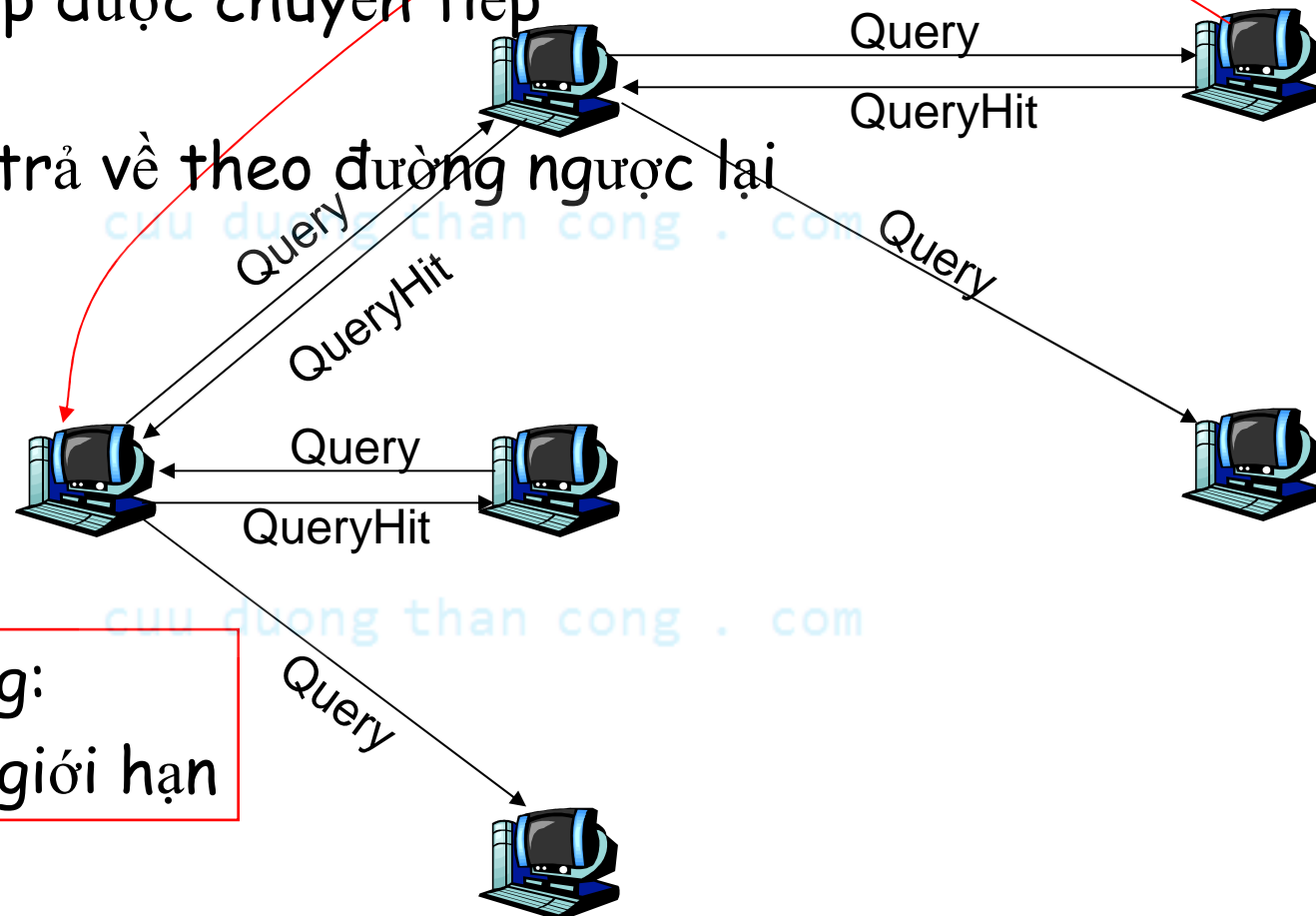
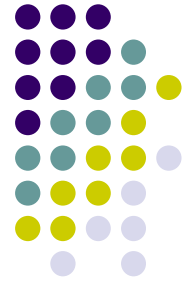
overlay network: Đồ thị

- X và Y có một cạnh nếu giữa chúng có một liên kết TCP
- Các máy đang hoạt động và các cạnh tạo nên đồ thị
- cạnh: liên kết logic
- thông thường có ít hơn 10 liên kết cho mỗi peer

Gnutella: protocol

- ❑ Thông điệp yêu cầu được quảng bá trên tất cả các cạnh
- ❑ Thông điệp được chuyển tiếp
- ❑ QueryHit sẽ được gửi trả về theo đường ngược lại

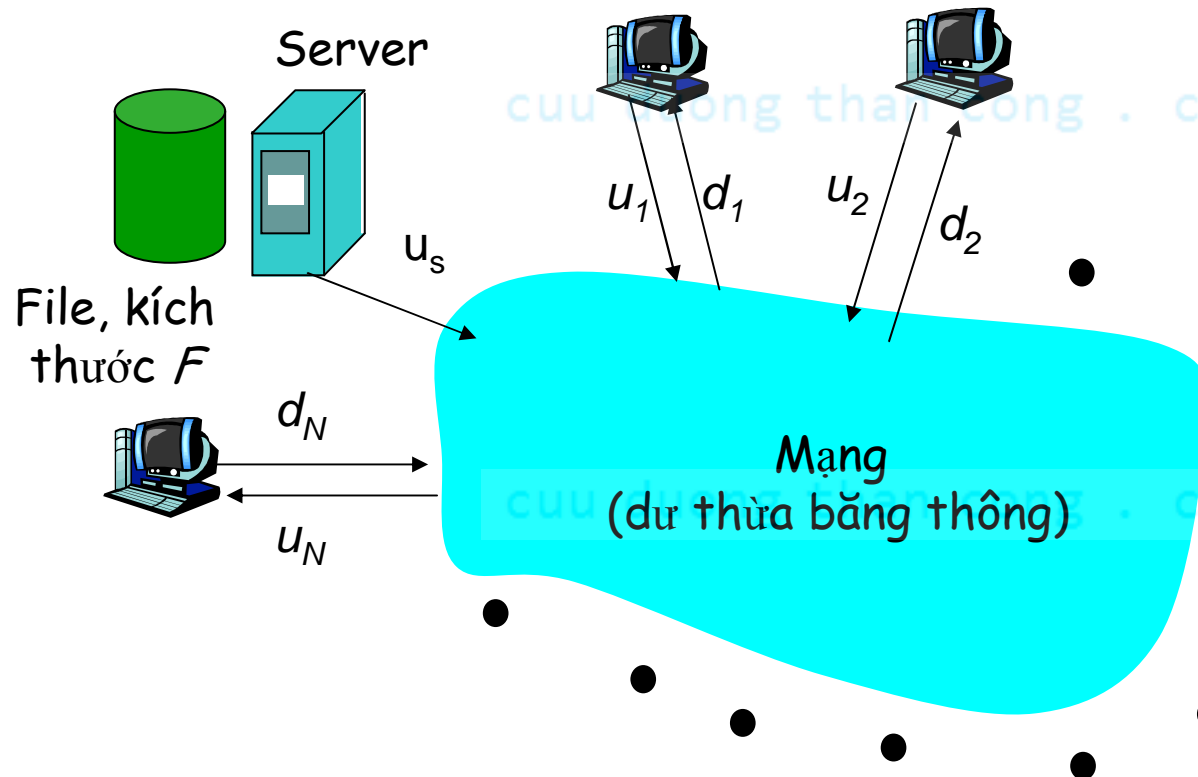
File transfer:
HTTP



Tính mở rộng:
Flooding có giới hạn

So sánh kiến trúc client-server và P2P

Vấn đề : Thời gian để gửi một file tới N máy tính khác?



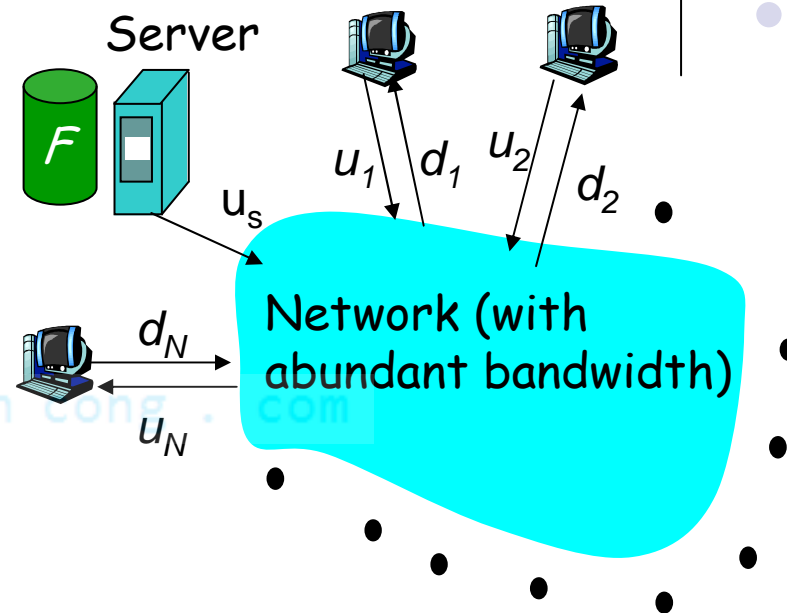
u_s : băng thông upload của server

u_i : băng thông upload của client/peer i

d_i : băng thông upload của client/peer i

Client-server: thời gian chia sẻ file

- server lần lượt gửi N copies:
 - NF/u_s
- client i mất F/d_i để download



Thời gian gửi tới N clients $= d_{cs} = \max \{ NF/u_s, F/\min_i(d_i) \}$

Tỉ lệ tuyến tính với N
(với N lớn)

P2P: thời gian chia sẻ file

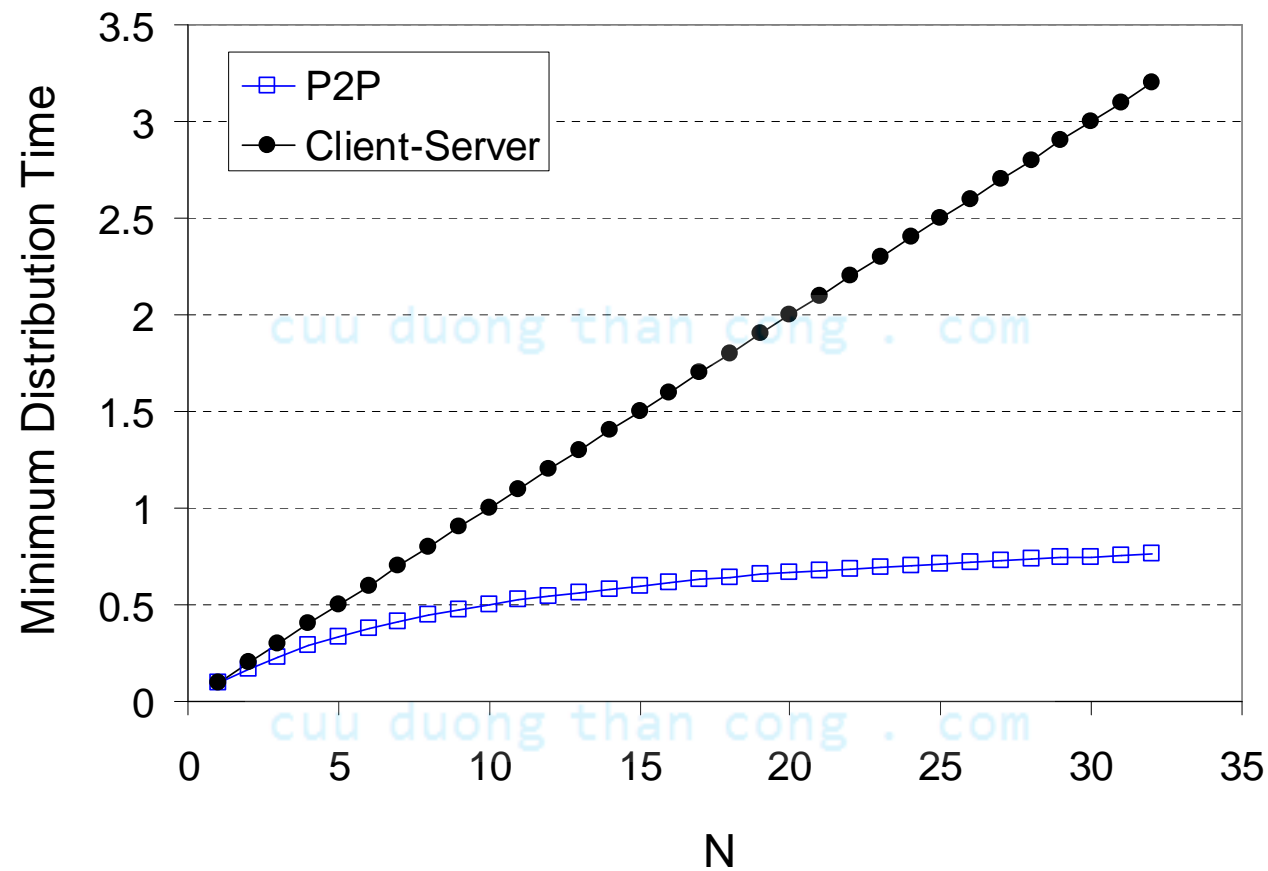
- server gửi một copy: F/u_s time
- client i mất F/d_i để tải xuống
- NF bits phải được tải xuống (tổng cộng)



- tốc độ upload cao nhất có thể (giả sử tất cả các nút gửi cùng các đoạn file tới cùng một máy): $u_s + \sum_{i=1, N} u_i$

$$d_{P2P} = \max \left\{ F/u_s, F/\min(d_i), NF/(u_s + \sum_{i=1, N} u_i) \right\}$$

So sánh kiến trúc Client-server, P2P



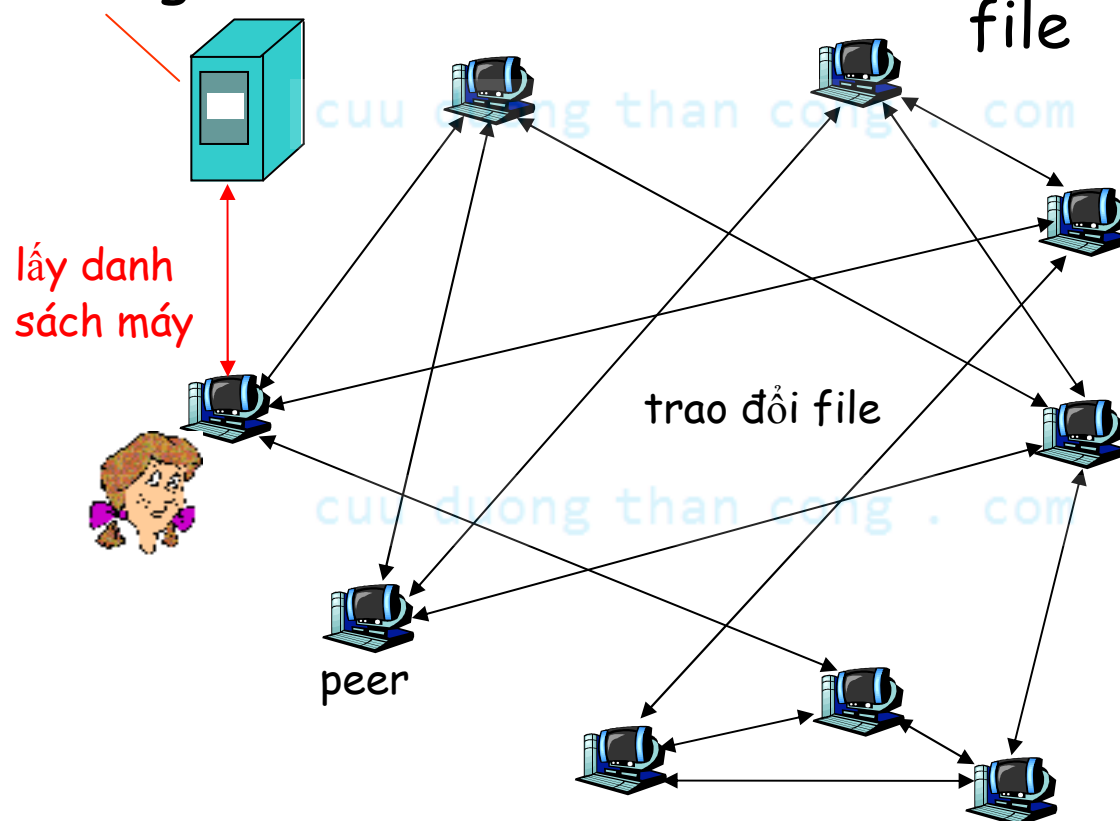


P2P Case Study: BitTorrent

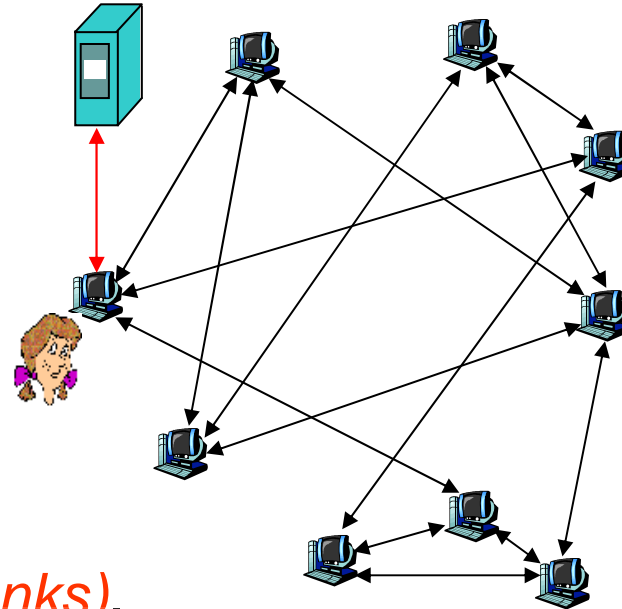
- Chia sẻ file P2P

tracker: tìm tất cả các máy trong torrent

torrent: nhóm các máy chia sẻ cùng một nhóm file



BitTorrent (1)



- file được chia thành 256KB (*chunks*).
- một máy tham gia vào torrent:
 - không có chunks, nhưng sẽ tích lũy dần theo thời gian
 - đăng ký với tracker để lấy danh sách các máy và connect đến các máy bên cạnh
- khi download sẽ đồng thời upload đến các máy khác .
- các máy có thể gia nhập/rời bỏ mạng
- Khi máy lấy file xong, nó có thể rời mạng 😞 hoặc tiếp tục tham gia chia sẻ 😊



BitTorrent (2)

Lấy chunks

- Tại mỗi thời điểm, các máy khác nhau sẽ có các đoạn file khác nhau
- thường xuyên hỏi các máy bên cạnh.
- A tìm các đoạn còn thiếu
 - rarest first

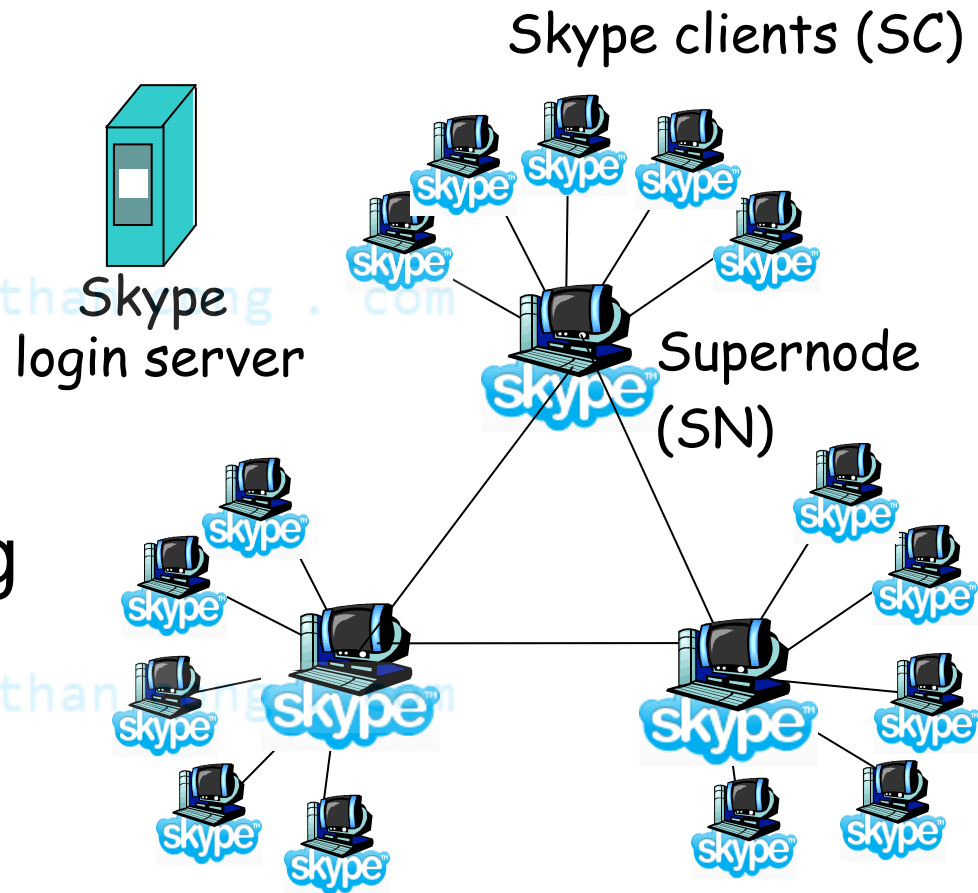
Gửi chunks: “Bánh ít- bánh quy”

- A gửi các đoạn file tới top 4 máy đã cho A với tốc độ cao nhất
 - Đánh giá lại top 4, cứ 10 secs
- Khoảng 30 secs: chọn ngẫu nhiên 1 máy khác và gửi đoạn file
 - cho phép các máy mới tham gia vào top 4

P2P Case study: Skype



- P2P (pc-to-pc, pc-to-phone, phone-to-pc) Voice-Over-IP (VoIP)
 - IM
- Giao thức ứng dụng riêng
- Mô hình phân cấp



Skype: thiết lập cuộc gọi

- User khởi tạo Skype
- SC đăng ký với SN
 - Danh sách SNs
- SC đăng nhập
- Call: SC liên lạc với SN, gửi ID người được gọi
- SN liên lạc với SNs khác để tìm ID của người kia, sau đó trả lại cho SC
- SC liên lạc trực tiếp với nhau, qua TCP

