



BÀI GIẢNG MÔN

***Internet và giao thức
(Internet and Protocols)***

Giảng viên:

TS. Phạm Anh Thu

Điện thoại/E-mail:

0912528188, thupaptit@gmail.com, thupa@ptit.edu.vn

Bộ môn:

Mạng viễn thông - Khoa Viễn thông 1

Học kỳ/Năm biên soạn: I/ 2022-2023

Chương 2: Các ứng dụng và giao thức

2.1 Tổng quan về các ứng dụng và dịch vụ hạ tầng

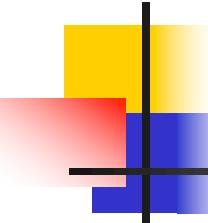
2.2 Ứng dụng WEB và các giao thức

2.3 Ứng dụng truyền tệp và các giao thức

2.4 Ứng dụng thư điện tử và các giao thức

2.5 Ứng dụng hệ thống tên miền

2.6 Các ứng dụng mạng ngang hàng



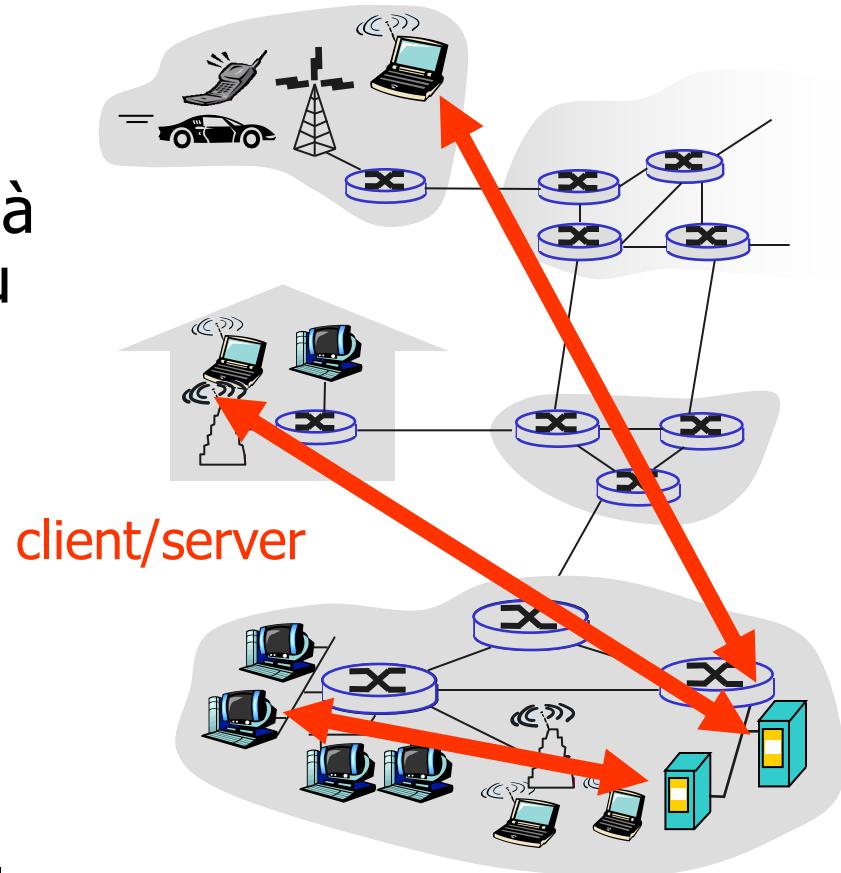
Kiến trúc lớp ứng dụng mạng Internet

- Kiến trúc ứng dụng: được các nhà phát triển ứng dụng thiết kế và chỉ rõ cách cấu trúc ứng dụng đó trên nhiều hệ thống cuối khác nhau
- Kiến trúc ứng dụng khác hoàn toàn với kiến trúc mạng (như kiến trúc phân lớp của Internet)
- Mô hình nổi bật được dùng trong các ứng dụng mạng hiện tại:
 - Kiến trúc khách/chủ (client/server)
 - Kiến trúc ngang hàng (P2P - peer-to-peer)

Kiến trúc lớp ứng dụng mạng Internet

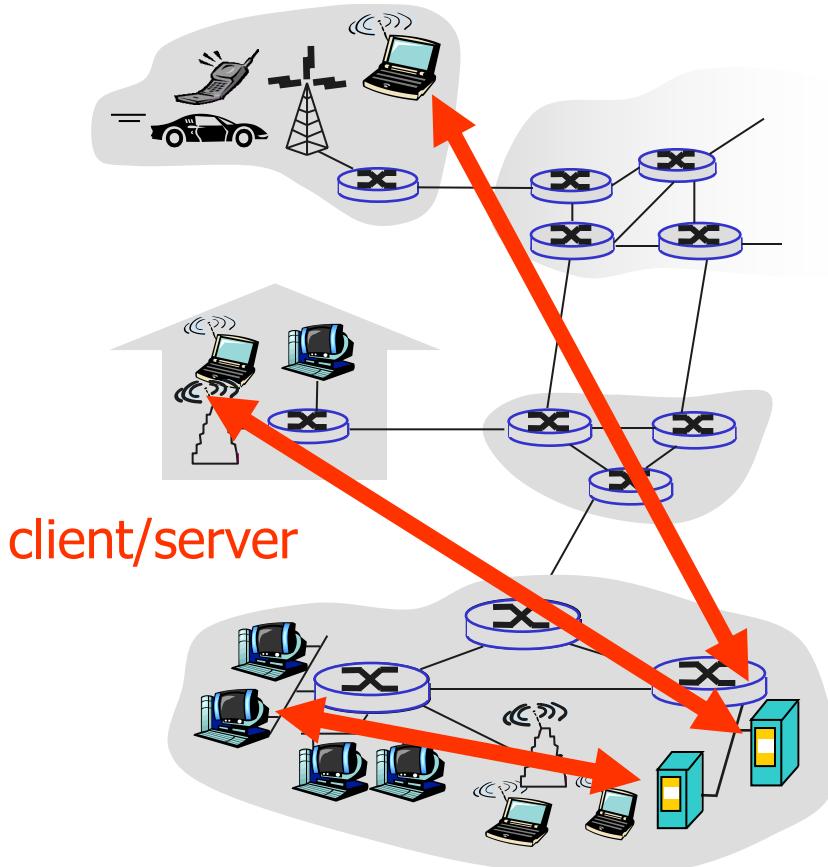
❖ Kiến trúc khách/chủ (client/server)

- ❑ Luôn có một máy trạm hoạt động gọi là máy chủ (server), phục vụ các yêu cầu từ nhiều máy trạm khác (client).
- ❑ Máy chủ có địa chỉ cụ thể và cố định (địa chỉ IP)
- ❑ Các máy khách có thể hoạt động liên tục hoặc không.
- ❑ Các máy khách không truyền thông trực tiếp với nhau
- ❑ Ví dụ về ứng dụng web, FTP, và e-mail



Kiến trúc lớp ứng dụng mạng Internet

Kiến trúc Client-server



Clients (các máy khách):

- Truyền thông với máy chủ
- Có thể được kết nối không liên tục
- Có thể có các địa chỉ IP động
- Không truyền thông trực tiếp với nhau

Server (máy chủ):

- Là trạm luôn ở trạng thái hoạt động
- Có địa chỉ IP cố định
- Mở rộng tạo thành các server farms

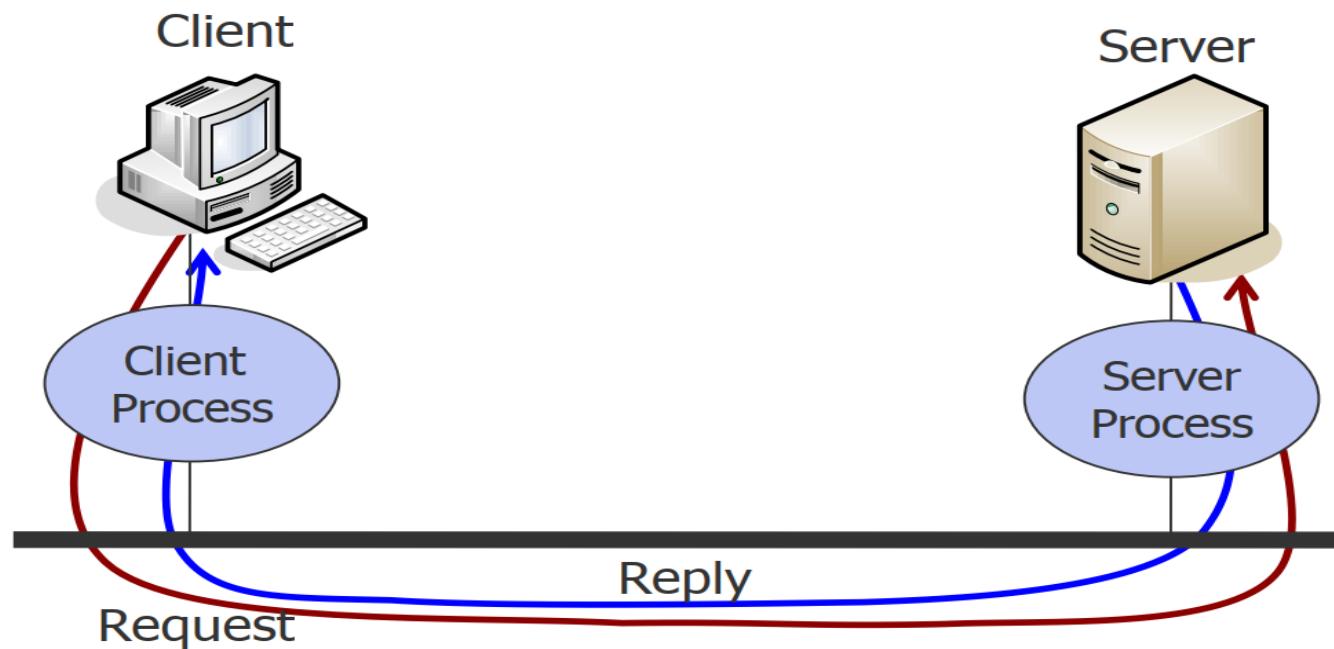
Clients và Servers (Khách/chủ)

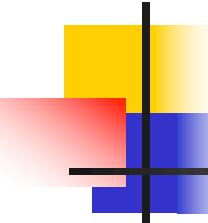
■ Chương trình Client

- Chạy trên trạm cuối
- Yêu cầu dịch vụ
- Ví dụ: Web browser

■ Chương trình Server

- Chạy trên trạm cuối
- Cung cấp dịch vụ
- Ví dụ: Web server



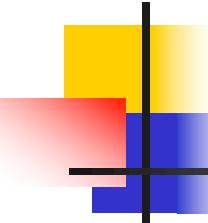


Kiến trúc lớp ứng dụng mạng Internet

- ❖ Kiến trúc khách/chủ (client/server)

- ❑ Ưu điểm:

- Điều khiển tập trung: mọi thông tin cần thiết đều ở một vị trí nên dễ dàng cho việc quản lý, giám sát
 - Bảo mật: dữ liệu được bảo mật tốt do nó có kiến trúc tập trung
 - Dễ dàng mở rộng
 - Dễ dàng quản lý dữ liệu
 - Dễ dàng truy cập



Kiến trúc lớp ứng dụng mạng Internet

❖ Kiến trúc khách/chủ (client/server)

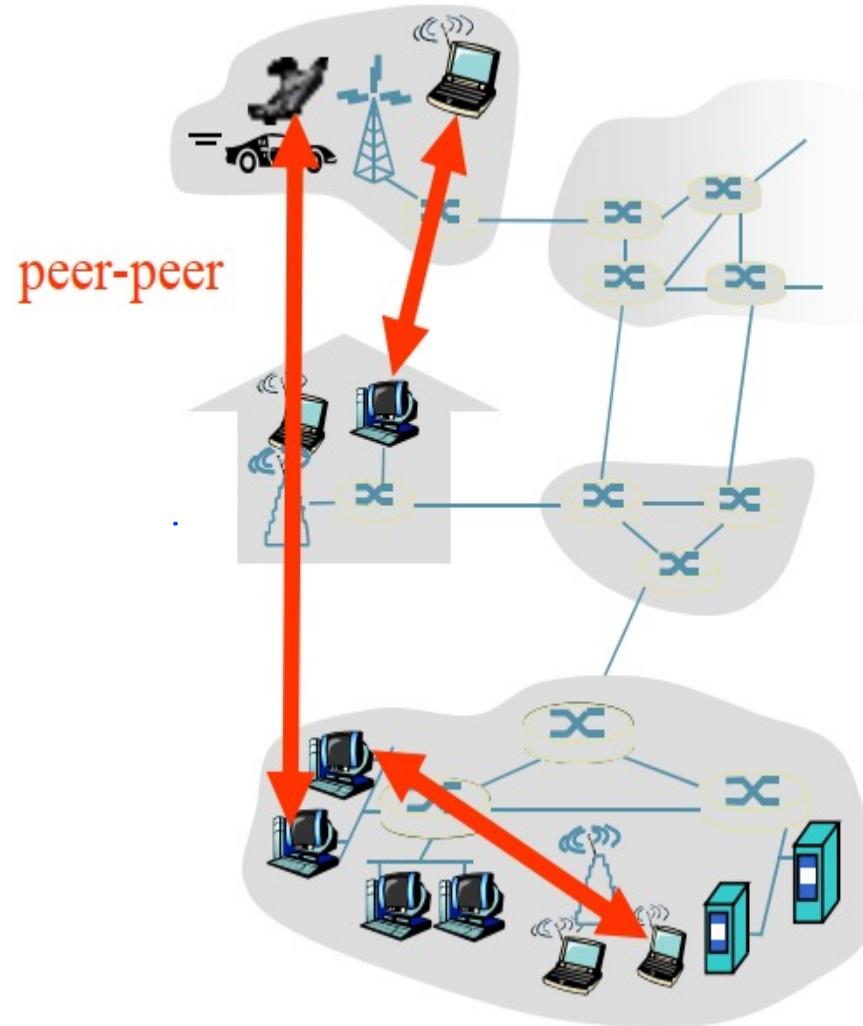
❑ Nhược điểm:

- Một server đơn lẻ không có khả năng đáp ứng kịp những yêu cầu từ các máy khách của nó
- Server có thể nhanh chóng bị sập nếu nó chỉ có một máy chủ xử lý tất cả các yêu cầu
- Một nhóm trạm lớn – đôi khi được gọi là trung tâm dữ liệu – được dùng để tạo ra một máy chủ ảo mạnh trong kiến trúc khách/chủ
- Cần nhiều cơ sở hạ tầng vì chúng yêu cầu nhà cung cấp dịch vụ phải mua, cài đặt và bảo dưỡng cụm máy chủ

Kiến trúc lớp ứng dụng mạng Internet

❖ Kiến trúc ngang hàng (P2P)

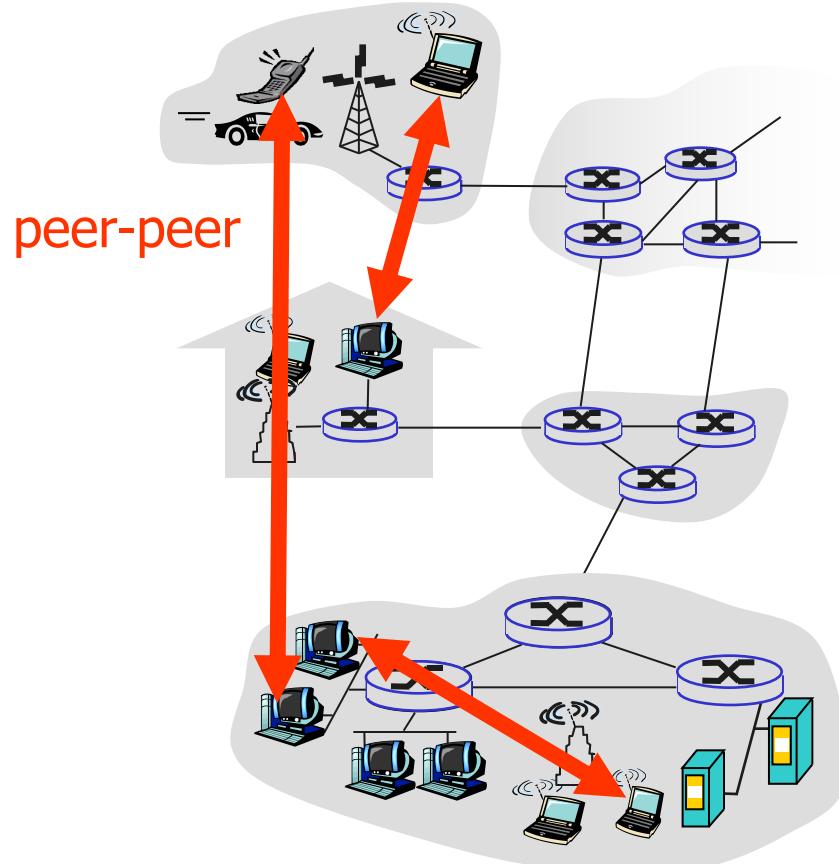
- ❑ Ứng dụng khai thác truyền thông trực tiếp giữa cặp trạm kết nối liên tục, gọi là các thiết bị ngang hàng (peer)
- ❑ Các thiết bị ngang hàng không phải của nhà cung cấp dịch vụ mà là các thiết bị đầu cuối người sử dụng ở trong nhà, trong trường học hay trong các công sở.
- ❑ Truyền thông giữa các thiết bị ngang hàng không cần qua máy chủ dành riêng

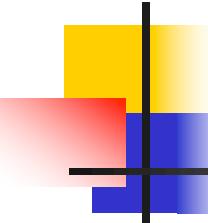


Kiến trúc lớp ứng dụng mạng Internet

- *Server không phải luôn hoạt động*
- Các hệ thống đầu cuối bất kỳ truyền thông trực tiếp
- Các peer được kết nối tùy ý và có địa chỉ IP thay đổi

➡ Dễ mở rộng nhưng khó quản lý





Kiến trúc lớp ứng dụng mạng Internet

❖ Kiến trúc ngang hàng

❑ Ưu điểm:

- Khả năng tự mở rộng (self-scalability)
- Hiệu quả về chi phí vì thường không cần hạ tầng máy chủ và băng thông máy chủ

❑ Nhược điểm:

- Vì có đặc tính mở và phân bổ rộng khắp nên các ứng dụng P2P sẽ có thách thức về vấn đề an toàn
- Thành công của các ứng dụng P2P còn phụ thuộc vào việc thuyết phục được người sử dụng tin tưởng chia sẻ nguồn lực như băng thông, bộ nhớ và khả năng tính toán cho các ứng dụng

Kiến trúc lớp ứng dụng mạng Internet

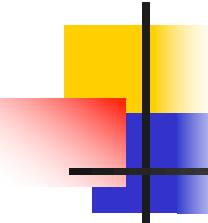
❖ Mô hình lai ghép client-server và P2P

Skype

- Ứng dụng voice-over-IP P2P
- Server tập trung: tìm địa chỉ đầu xa
- Kết nối client-client: trực tiếp (không qua server)

Nhắn tin ngắn trên mạng

- Chatting giữa hai người dùng là P2P
- Dịch vụ tập trung: phát hiện vị trí/hiện diện của client
 - Người dùng đăng ký địa chỉ IP của mình với server trung tâm khi lên online
 - Người sử dụng kết nối tới server trung tâm để tìm địa chỉ của bạn chat

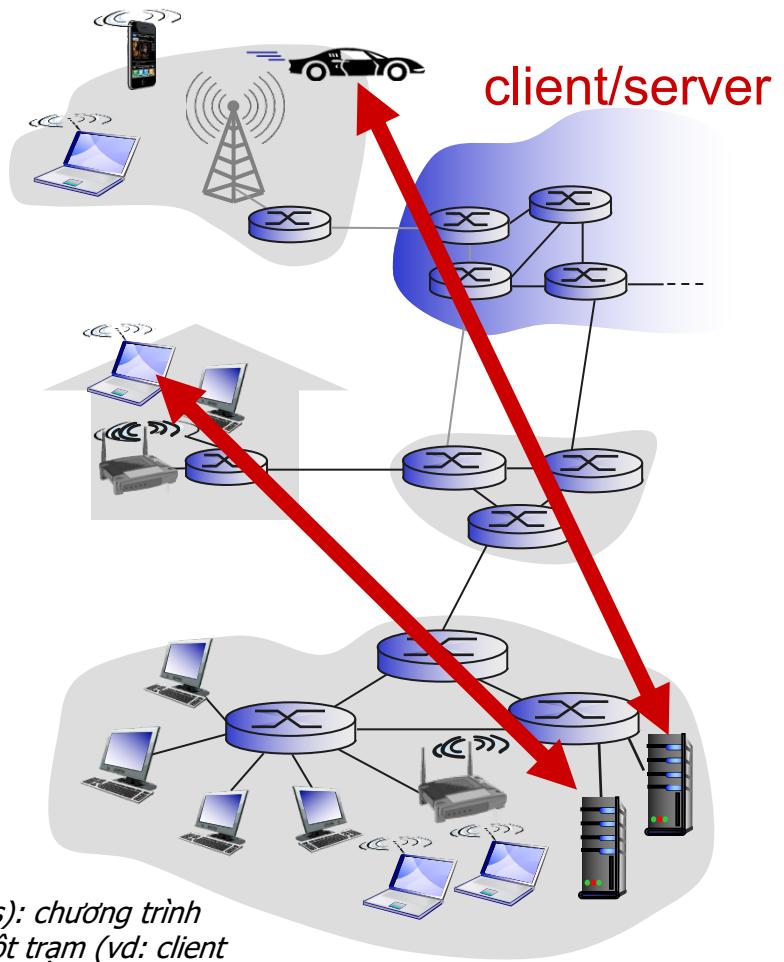
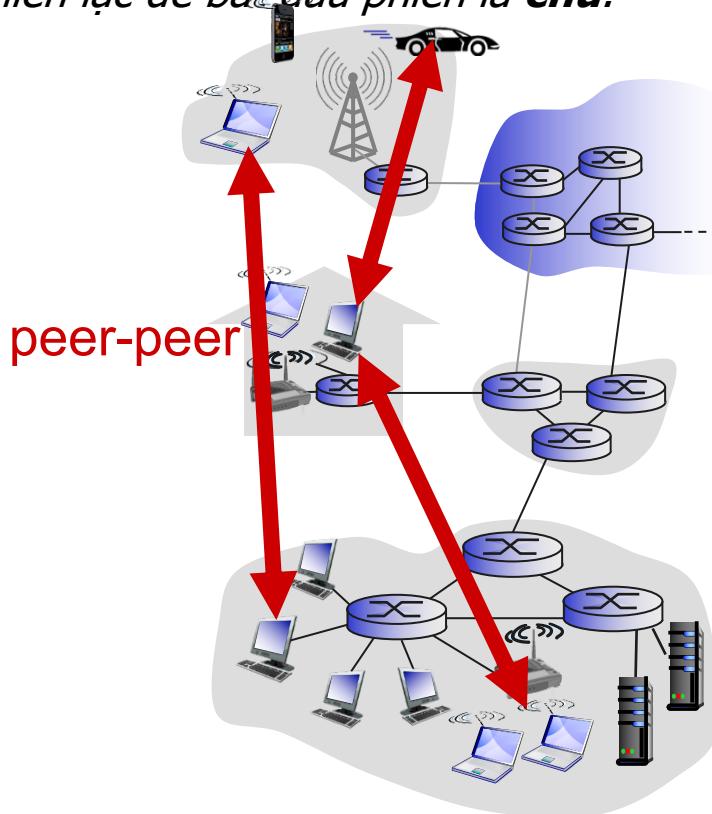


Quá trình truyền thông trên mạng

- Trước khi xây dựng ứng dụng, cần phải nắm được cách thức để chương trình chạy trên các hệ thống đầu cuối truyền thông với nhau.
- Chương trình chạy trong một hệ thống cuối: tiến trình
- Tiến trình khách chủ: là một cặp tiến trình truyền thông
 - ✓ Với ứng dụng Web, trình duyệt là tiến trình khách, còn máy chủ Web là tiến trình chủ
 - ✓ Với ứng dụng chia sẻ tệp P2P thì thiết bị ngang hàng tải tệp dữ liệu xuống được gọi là khách còn thiết bị ngang hàng tải dữ liệu lên được gọi là chủ.

Quá trình truyền thông trên mạng

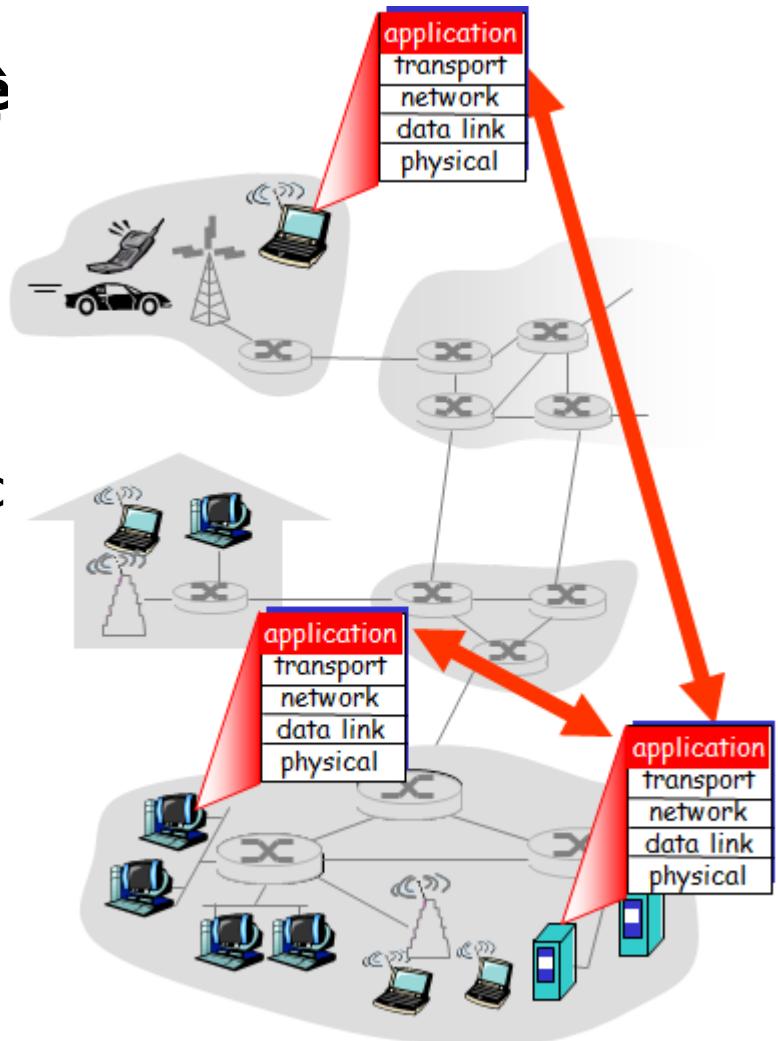
- Định nghĩa:** "Trong ngữ cảnh phiên truyền thông giữa một cặp tiến trình, tiến trình kích hoạt truyền thông (nghĩa là khởi đầu liên hệ với tiến trình khác tại đầu phiên) được gọi là **khách**. Còn tiến trình chờ được liên lạc để bắt đầu phiên là **chủ**."



Tiến trình (process): chương trình chạy bên trong một trạm (vd: client process, server process)
 Truyền thông (communication)
 Ứng dụng (application)
 Giao diện (interface)

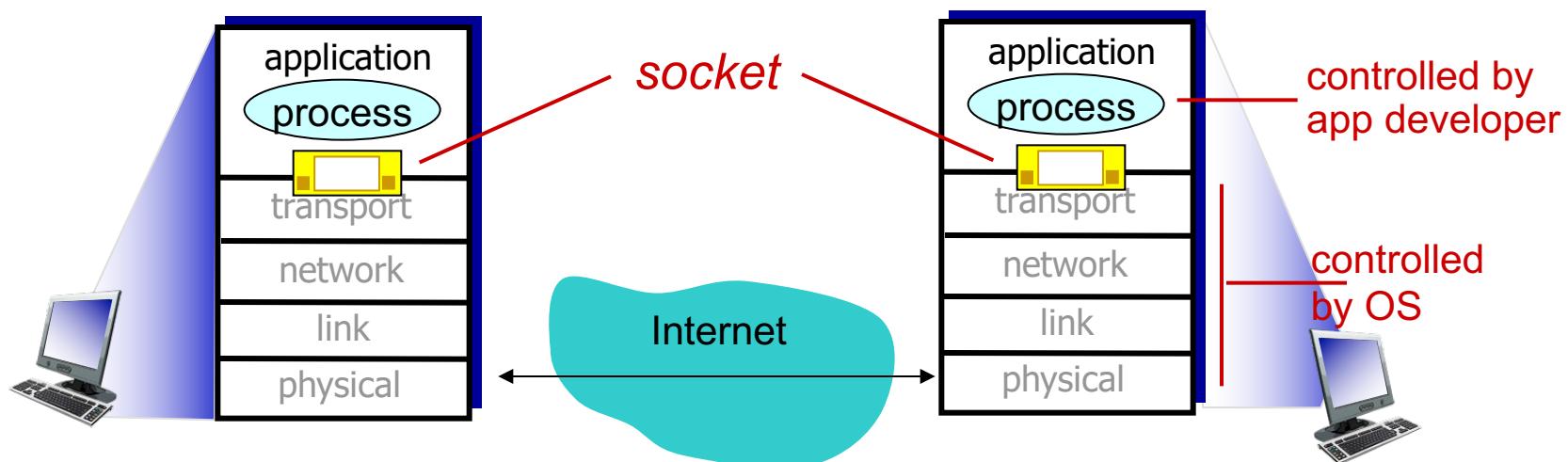
Quá trình truyền thông trên mạng

- Khi các tiến trình chạy **trên cùng một hệ thống cuối** thì chúng truyền thông với nhau theo kiểu **liên tiến trình**, sử dụng những quy tắc do HĐH của hệ thống đầu cuối đó điều khiển.
- Tiến trình truyền thông trên hai/nhiều trạm khác nhau (có thể các trạm chạy các HĐH khác nhau): gửi các bản tin xuyên qua mạng máy tính.
 - Tiến trình gửi tạo ra và gửi các bản tin vào mạng.
 - Tiến trình nhận sẽ nhận các bản tin này và có thể đáp lại bằng việc gửi bản tin ngược lại.



Quá trình truyền thông trên mạng

- Chức năng giao diện giữa tiến trình và mạng
 - Một tiến trình gửi bản tin vào mạng và nhận bản tin từ mạng qua một giao diện phần mềm gọi là **socket**.
 - Một tiến trình có số nhận dạng (identifier) gồm địa chỉ IP và số cổng gắn với tiến trình trong trạm. Ví dụ gửi bản tin HTTP tới máy của web gaia.cs.umass.edu thì
 - IP address: 128.119.245.12
 - port number: 80



Dịch vụ truyền tải cho ứng dụng

Các dịch vụ mà giao thức lớp giao vận cung cấp cho ứng dụng

Toàn vẹn dữ liệu (data integrity)

- some apps (e.g., file transfer, web transactions) require 100% reliable data transfer
- other apps (e.g., audio) can tolerate some loss

Thời gian (timing)

- some apps (e.g., Internet telephony, interactive games) require low delay to be “effective”

Thông lượng (throughput)

- ❖ some apps (e.g., multimedia) require minimum amount of throughput to be “effective”
- ❖ other apps (“elastic apps”) make use of whatever throughput they get

Bảo mật (security)

- ❖ encryption, data integrity,

...

Dịch vụ truyền tải cho ứng dụng

Yêu cầu của một số ứng dụng

| Ứng dụng | Tổn thất dữ liệu | Băng thông | Độ nhạy về thời gian |
|-------------------------------------------|--------------------|---------------------------------------------------|-----------------------|
| Truyền tệp | Không tổn thất | Thay đổi | Không |
| E-mail (thư điện tử) | Không tổn thất | Thay đổi | Không |
| Web | Không tổn thất | Thay đổi (vài kb/s) | Không |
| Điện thoại Internet/ hội nghị Video | Chịu được tổn thất | Audio: vài kb/s – 1 Mb/s Video: 10kb/s – 5Mb/s | Có: $n \times 100$ ms |
| Audio/video lưu trữ | Chịu được tổn thất | | Có: $n \times s$ |
| Trò chơi tương tác | Chịu được tổn thất | Vài kb/s-10kb/s | Có: $n \times 100ms$ |
| Nhắn tin thức thời | Không tổn thất | Thay đổi | Có và không |

Các dịch vụ truyền tải

TCP service:

- *reliable transport* between sending and receiving process
- *flow control*: sender won't overwhelm receiver
- *congestion control*: throttle sender when network overloaded
- *does not provide*: timing, minimum throughput guarantee, security
- *connection-oriented*: setup required between client and server processes

UDP service:

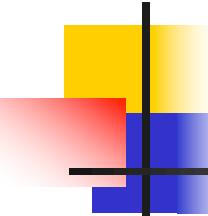
- *unreliable data transfer* between sending and receiving process
- *does not provide*: reliability, flow control, congestion control, timing, throughput guarantee, security, or connection setup

→ **Which applications rely on TCP and/or UDP**

Các dịch vụ truyền tải

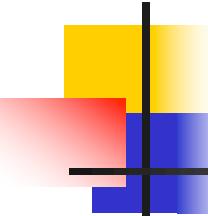
- Các ứng dụng Internet điển hình, giao thức lớp ứng dụng và giao vận

| Ứng dụng | Giao thức lớp ứng dụng | Giao thức lớp giao vận |
|---------------------------|-----------------------------------|------------------------|
| E-mail (thư điện tử) | SMTP [RFC 5321] | TCP |
| Truy nhập đầu cuối từ xa | Telnet [RFC 854] | TCP |
| Web | HTTP [RFC 2616] | TCP |
| Truyền tệp | FTP [RFC 959] | TCP |
| Trực tuyến đa phương tiện | HTTP (ví dụ YouTube), RTP | TCP hoặc UDP |
| Điện thoại Internet | SIP, RTP hoặc riêng (ví dụ Skype) | Thường là UDP |



Các giao thức lớp ứng dụng

- Giao thức lớp ứng dụng định nghĩa các thủ tục của ứng dụng, chạy trên các hệ thống cuối khác nhau, qui định cách thức chuyển các bản tin cho nhau như thế nào. Một giao thức lớp ứng dụng định nghĩa các vấn đề sau:
 - Loại bản tin trao đổi, ví dụ bản tin yêu cầu hay bản tin phản hồi;
 - Cú pháp của các loại bản tin khác nhau, như các trường trong bản tin và cách mô tả các trường này;
 - Ngữ nghĩa của các trường, tức là ý nghĩa của trường thông tin;
 - Quy tắc xác định một tiến trình gửi và phản hồi bản tin khi nào và như thế nào.
- Giao thức lớp ứng dụng có thể công khai (RFC) hoặc không công khai (1 số hệ thống P2P)



Các giao thức lớp ứng dụng

- Giao thức lớp ứng dụng chỉ là một phần của ứng dụng mạng:
 - Web là một ứng dụng khách-chủ cho phép người sử dụng lấy dữ liệu từ máy chủ Web theo yêu cầu.
 - Ứng dụng Web gồm rất nhiều phần tử:các khuôn dạng tài liệu chuẩn (HTML), các trình duyệt Web (ví dụ Firefox và Microsoft Internet Explorer), các máy chủ Web (ví dụ máy chủ Apache và Microsoft) và một giao thức lớp ứng dụng.
 - Giao thức lớp ứng dụng của Web là HTTP: định nghĩa khuôn dạng và trình tự của các bản tin chuyển giữa phần mềm trình duyệt và máy chủ Web.
- Một số giao thức lớp ứng dụng: HTTP, FTP, DNS, ...

Một số giao thức lớp ứng dụng

- Một số giao thức lớp ứng dụng: HTTP, FTP, DNS, ...

| Application / Service | Acronym | Port |
|-------------------------------------|---------|--------|
| Domain Name System | DNS | 53 |
| Hypertext Transfer Protocol | HTTP | 80 |
| Simple Mail Transfer Protocol | SMTP | 25 |
| Post Office Protocol | POP3 | 110 |
| Telnet | Telnet | 23 |
| Dynamic Host Configuration Protocol | DHCP | 67 |
| File Transfer Protocol | FTP | 20, 21 |

Chương 2: Các ứng dụng và giao thức

2.1 Tổng quan về các ứng dụng và dịch vụ hạ tầng

2.2 Ứng dụng WEB và các giao thức

2.3 Ứng dụng truyền tệp và các giao thức

2.4 Ứng dụng thư điện tử và các giao thức

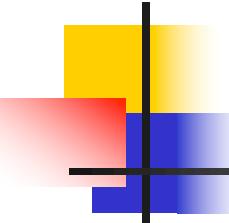
2.5 Ứng dụng hệ thống tên miền

2.6 Các ứng dụng mạng ngang hàng

Chương 2: Các ứng dụng và giao thức

- Nội dung chính
 - Tổng quan về HTTP
 - Các kết nối HTTP
 - Khuôn dạng bản tin HTTP
 - Tương tác người sử dụng-máy chủ: Cookie
 - Lưu đệm Web (Web caching)
 - Cơ chế GET có điều kiện





Tổng quan về WEB và HTTP

- **Đầu những năm 1990, Internet chủ yếu được dùng** để truy nhập vào các trạm ở xa, truyền tệp từ trạm nội bộ tới trạm từ xa và ngược lại, để nhận và gửi tin tức, thư điện tử.
- **Sau đó WEB ra đời**, là ứng dụng Internet đầu tiên có được sự quan tâm của toàn công chúng
- Web đã thúc đẩy Internet từ vị trí là một trong nhiều mạng truyền dữ liệu thành mạng dữ liệu duy nhất.
- Web vận hành *theo yêu cầu*, khác với truyền thanh và truyền hình quảng bá
- Tạo thông tin trên Web khá dễ dàng – tất cả mọi người đều có thể trở thành người phát hành thông tin với chi phí thấp

Tổng quan về WEB và HTTP

Mô hình hoạt động của WEB

Web Server /
Application server

Storage System

Web Browser



Tổng quan về WEB và HTTP

Trang Web chứa các **đối tượng (object)**, sử dụng **HTML (HyperText Markup Language)** làm ngôn ngữ để tạo và cấu trúc các thành phần trong trang Web.

- **Đối tượng** có thể là file HTML, hình ảnh JPEG, ứng dụng Java, file audio,...
- **Trang web** chứa **file HTML cơ bản** bao gồm nhiều đối tượng tham chiếu
- Mỗi đối tượng được xác định bằng địa chỉ **URL (Uniform Resource Locator)**

- Ví dụ về URL:

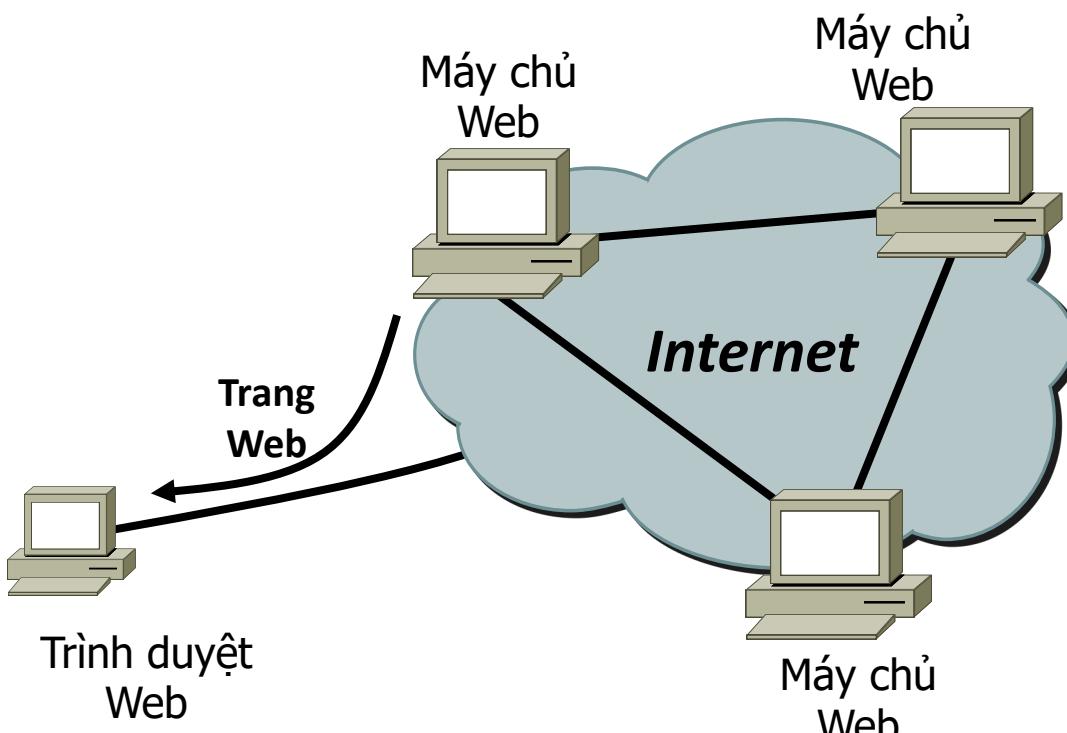
www.someschool.edu/someDepartment/pic.gif

host name

path name

Tổng quan về WEB và HTTP

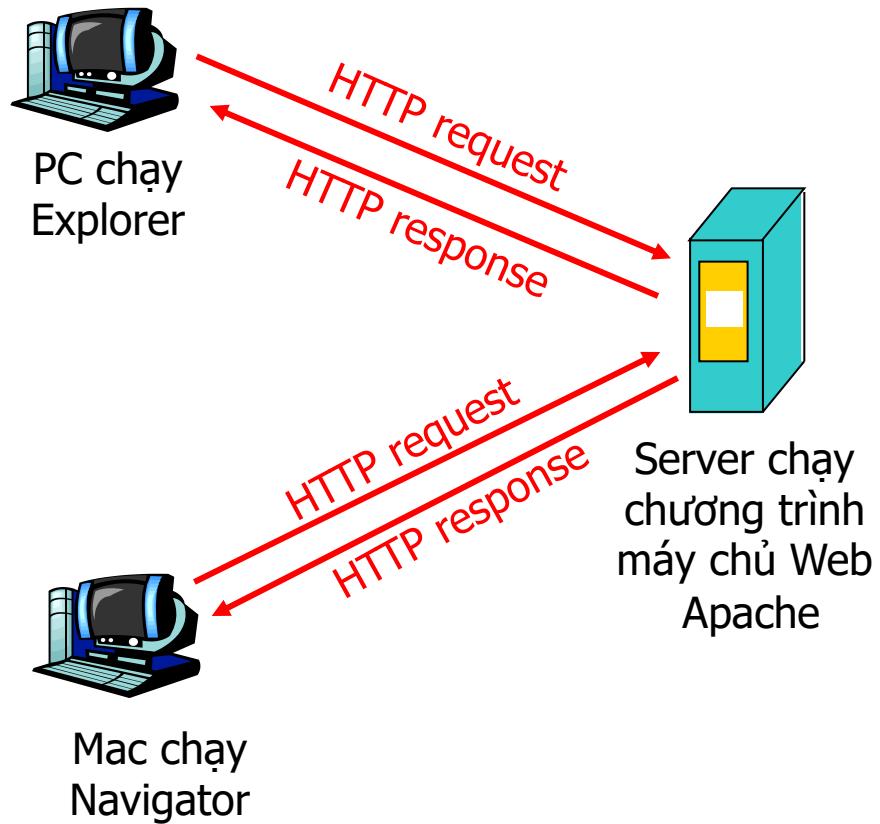
- **HTTP (Hypertext Transfer Protocol) là một giao thức lớp ứng dụng của WEB**, được định nghĩa trong RFC 1945 và RFC 2616
- HTTP được thực hiện trong hai chương trình: chương trình máy khách và chương trình máy chủ. Chúng giao tiếp với nhau bằng cách trao đổi các bản tin HTTP.



Tổng quan về WEB và HTTP

HTTP: hypertext transfer protocol (giao thức truyền siêu văn bản)

- Là giao thức lớp ứng dụng của Web
- Mô hình khách chủ (client/server)
 - *client*: trình duyệt yêu cầu (request), nhận, hiển thị các đối tượng Web
 - *server*: Server Web gửi các đối tượng để đáp ứng (response) lại yêu cầu của client



Tổng quan về WEB và HTTP

HTTP Sử dụng TCP:

1. Client khởi tạo kết nối TCP (tạo socket) tới server, cổng **80**
2. Server chấp nhận kết nối TCP từ client
3. Các bản tin HTTP (bản tin giao thức lớp ứng dụng) được trao đổi giữa trình duyệt (HTTP client) và server web (HTTP server)
4. Đóng kết nối TCP

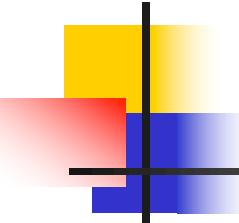
HTTP không có trạng thái

- Server không duy trì thông tin về những yêu cầu trước đó của client

Bên cạnh đó

Các giao thức có duy trì trạng thái thì rất phức tạp!

- Trạng thái cũ (lịch sử) phải được lưu trữ
- Nếu server/client bị phá hỏng, trạng thái có thể không đồng nhất và phải thống nhất lại



Kết nối HTTP

- Trong mô hình client-server: client tạo ra các **request** tới server, server sẽ **response** mỗi request
 - Chuỗi các request có thể được tạo ra **liên tục** hoặc **định kỳ** theo thời gian
 - Do đó, nhà phát triển ứng dụng cần tạo ra các quyết định quan trọng:
 - ❖ Các cặp request/response nên được gửi riêng trong mỗi kết nối TCP?
 - ❖ Tất cả request/response được gửi qua cùng một kết nối TCP?
- **Kết nối liên tục** và **kết nối không liên tục**

Kết nối HTTP

1: Kết nối không liên tục

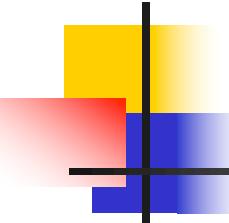
Non-persistent HTTP

- Chỉ **một** đối tượng được gửi qua **một** kết nối TCP .
- HTTP/1.0 sử dụng kết nối không liên tục

2: Kết nối liên tục

Persistent HTTP

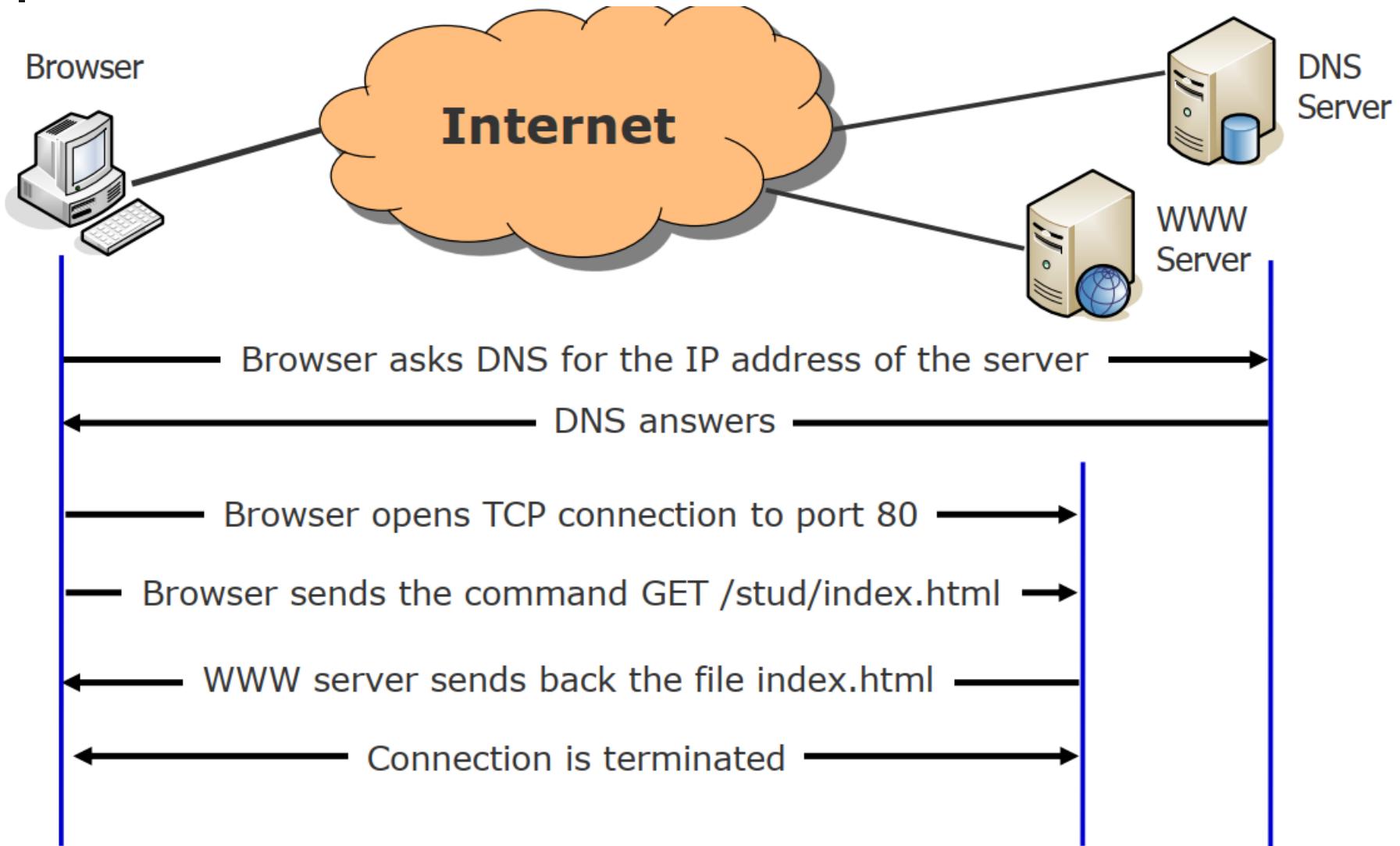
- **Nhiều** đối tượng có thể được gửi trên **một** kết nối TCP giữa client và server.
- HTTP/1.1 sử dụng kết nối liên tục trong chế độ mặc định

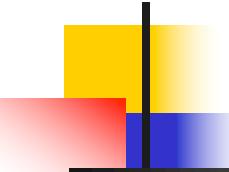


Kết nối HTTP

- Một kết nối HTTP bao gồm các thành phần:
 - ❖ Connection: client tạo kết nối tới server
 - Biên dịch tên miền sang địa chỉ IP nhờ DNS
 - Client yêu cầu mở kết nối TCP với server
 - ❖ Request: client yêu cầu thông tin từ server
 - ❖ Response: server đáp ứng thông tin được yêu cầu hoặc từ chối thông tin được yêu cầu
 - ❖ Close: client hoặc server hoặc cả hai kết thúc phiên kết nối TCP

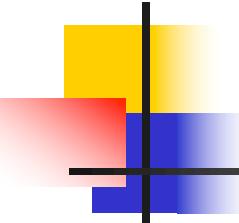
Kết nối HTTP





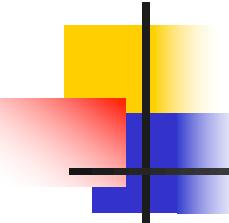
Kết nối HTTP

- Example: Call of the URL <http://cst.mi.fu-berlin.de/teaching/WS0708/19540-V/index.html>
 1. The Browser determines the URL (which was clicked or typed)
 2. The Browser asks the DNS for the IP address of the server cst.mi.fu-berlin.de
 3. DNS answers with 160.45.117.167
 4. The browser opens a TCP connection to port 80 of the computer 160.45.117.167
 5. Afterwards, the browser sends the command GET [/teaching/WS0708/19540-V/index.html](http://cst.mi.fu-berlin.de/teaching/WS0708/19540-V/index.html)
 6. The WWW server sends back the file index.html
 7. The connection is terminated
 8. The browser analyzes the WWW page index.html and presents it
 9. If necessary, each picture is reloaded over a new connection to the server
(The address is included in the page index.html in form of an URL)
- Note!
 - Step 9 applies only to HTTP/1.0! With the newer version HTTP/1.1 all referenced pictures are loaded before the connection termination (more efficiently for pages with many pictures)



Kết nối HTTP không liên tục

- Kết nối HTTP không liên tục sẽ được **đóng** sau khi server gửi đáp ứng cho client
- **Một kết nối** chỉ được sử dụng cho **1 cặp request/response**
- Để download **nhiều đối tượng** thì cần **nhiều kết nối**, như:
 - ❖ Truyền 1 webpage từ server đến client, webpage gồm 1 file HTML cơ bản và 10 hình ảnh JPEG
 - ❖ Tổng có 11 đối tượng nằm ở server, cần 11 kết nối TCP tương ứng



Kết nối HTTP không liên tục

Giả sử người sử dụng vào URL

www.someSchool.edu/someDepartment/home.index

11 đối tượng: Tập HTML cơ bản và 10 hình ảnh jpeg

- 1a. Client HTTP **khởi tạo** kết nối TCP tới server HTTP (process-tiến trình) tại www.someSchool.edu trên cổng 80

2. Client HTTP gửi bản tin yêu cầu HTTP (chứa URL) vào socket kết nối TCP. Bản in này cho biết client muốn đối tượng someDepartment/home.index

5. Client HTTP nhận bản tin đáp ứng chứa file html, hiển thị html. Phân tích cú pháp file html, tìm 10 đối tượng jpeg tham chiếu.

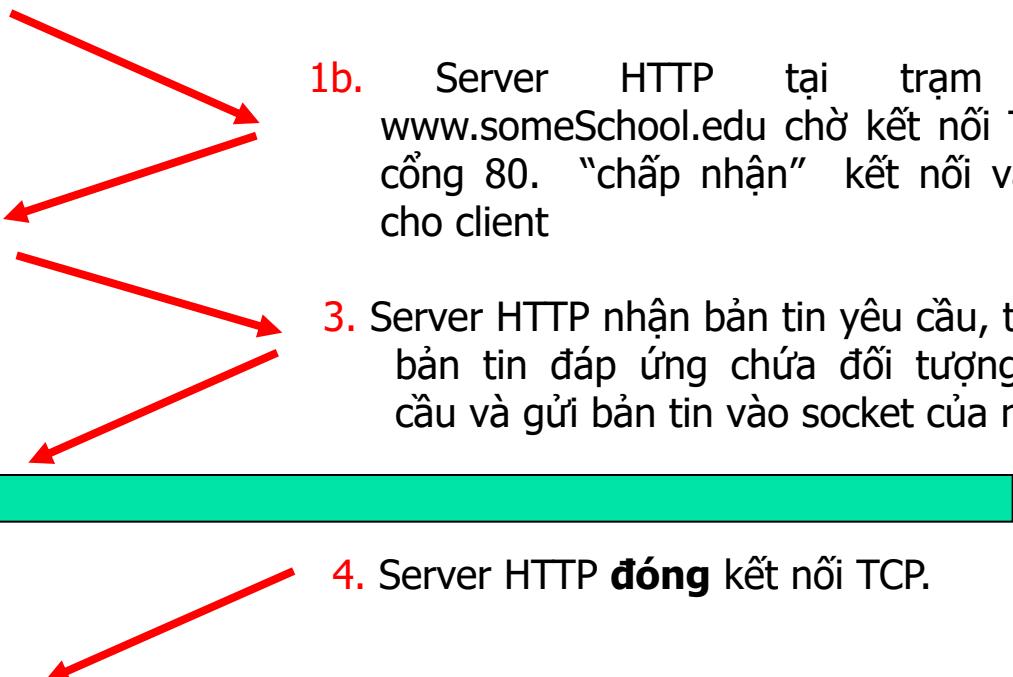
6. **Lặp lại** các bước từ 1-5 với từng đối tượng jpeg đó.

- 1b. Server HTTP tại trạm chủ www.someSchool.edu chờ kết nối TCP ở cổng 80. "chấp nhận" kết nối và báo cho client

3. Server HTTP nhận bản tin yêu cầu, tạo ra bản tin đáp ứng chứa đối tượng yêu cầu và gửi bản tin vào socket của nó.

4. Server HTTP **đóng** kết nối TCP.

time



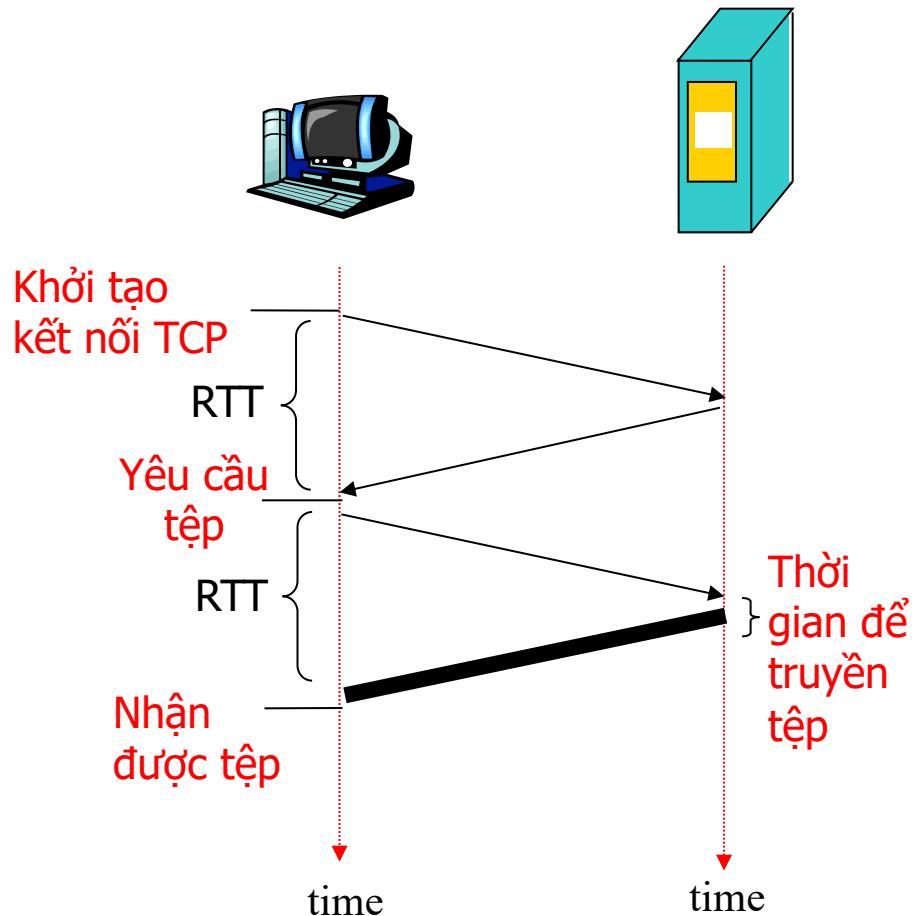
Kết nối HTTP không liên tục

Định nghĩa RTT: thời gian cho một gói tin đi từ client đến server và phản hồi trở lại.

Thời gian đáp ứng:

- Một RTT dùng để khởi tạo kết nối TCP
- Một RTT cho yêu cầu HTTP và một vài byte đầu của đáp ứng HTTP được phản hồi lại
- Thời gian truyền file

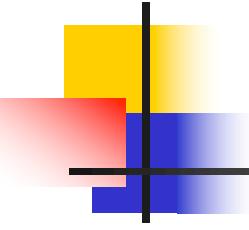
$$\text{Tổng} = 2\text{RTT} + \text{thời gian truyền file}$$



Kết nối HTTP không liên tục

Nhược điểm của kết nối HTTP không liên tục:

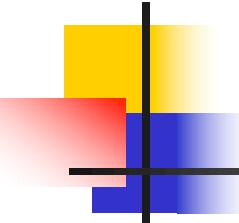
- Non-persistent connections have some **shortcomings**.
- **Firstly**, Each of new connections established, **TCP buffers** allocated and **TCP variables** must **kept** in both client and server side.
- This can a **significant burden** on web server, which may be increase in case of serving hundred of different clients requests simultaneously.
- **Second**, Each object **suffers** a delivery **delay of two RTTs**;
 - ✓ **one RTT** to establish TCP connection.
 - ✓ **one RTT** to request/receive an object.



Kết nối HTTP không liên tục

Nhược điểm của kết nối HTTP không liên tục:

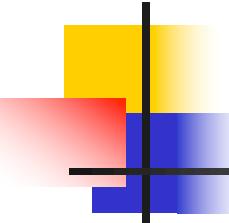
- Phải thiết lập và duy trì kết nối mới cho mỗi đối tượng được yêu cầu.
- Với mỗi kết nối TCP, phải cấp phát bộ đệm TCP và phải duy trì các biến TCP trên cả máy khách và máy chủ. Điều này có thể áp đặt tải trọng lên máy chủ Web, do nó có thể phải phục vụ yêu cầu cho hàng trăm máy khách khác nhau đồng thời.
- Mỗi đối tượng sẽ chịu một thời gian trễ chuyển phát khoảng 2 RTT: một RTT để thiết lập kết nối TCP và một RTT để yêu cầu và nhận đối tượng.



Kết nối HTTP liên tục

Persistent HTTP

- Server để kết nối mở sau khi gửi đáp ứng
- Các bản tin HTTP của cùng cặp client/server được gửi trên kết nối mở đó
- Client gửi các yêu cầu bất cứ khi nào nó gặp đối tượng tham chiếu
- Chỉ cần một RTT cho tất cả các đối tượng tham chiếu

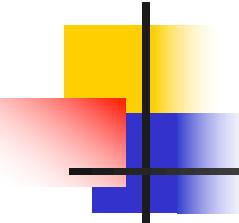


Bài tập kết nối HTTP

Bài tập 1:

Suppose within your Web browser you click on a link to obtain a Web page.

The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that n DNS servers are visited before your host receives the IP address from DNS; the successive visits incur an RTT of RTT_1, \dots, RTT_n . Further suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Let RTT_0 denote the RTT between the local host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the object?

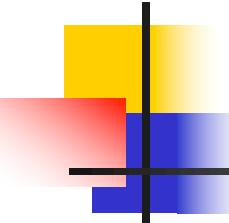


Bài tập kết nối HTTP

Bài tập 2:

Referring to Problem 1, suppose the HTML file references eight very small objects on the same server. Neglecting transmission times, how much time elapses with

- a. Non-persistent HTTP with no parallel TCP connections?
- b. Non-persistent HTTP with the browser configured for 5 parallel connections?
- c. Persistent HTTP?



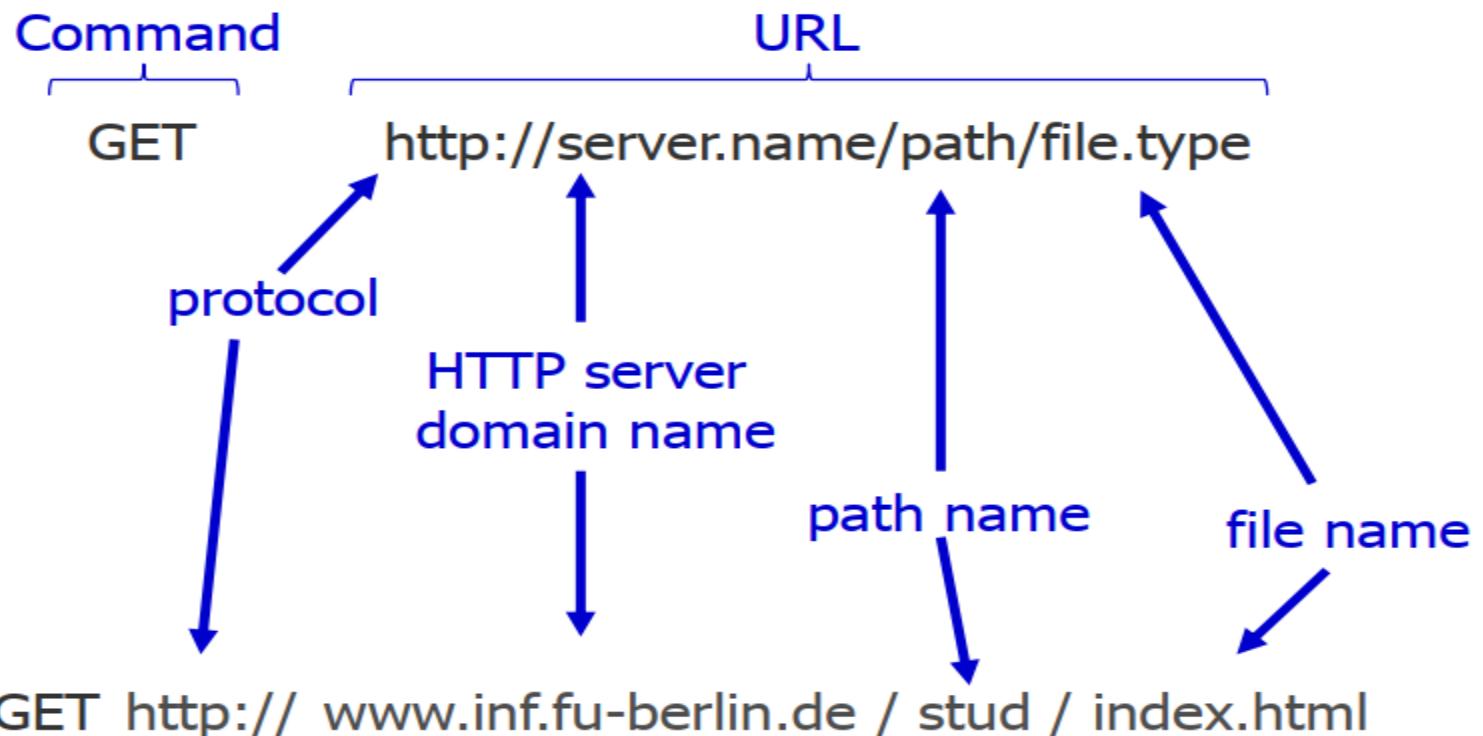
Khuôn dạng bản tin HTTP

■ Hai loại bản tin HTTP:

- ❖ *Request (yêu cầu)*: sử dụng định dạng ASCII
 - Request line: gồm 3 trường: Method, URL, HTTP version
 - Header lines
 - Blank line
 - Entity body
- ❖ *Response (đáp ứng)*
 - Status line
 - Header lines
 - Blank line
 - Entity body

Khuôn dạng bản tin HTTP

- Hai loại bản tin HTTP:
request (yêu cầu),
response (đáp ứng)



- Instructions on a URL are:
 - GET: Load a web page
 - HEAD: Load only the header of a web page
 - PUT: Store a web page on the server
 - POST: Append something to the request passed to the web server
 - DELETE: Delete a web page

Khuôn dạng bản tin yêu cầu HTTP

- Bản tin **yêu cầu** HTTP: ASCII (khuôn dạng mà con người đọc được)

| method | sp | URL | sp | version | cr | If |
|-------------------|----|-------|----|---------|----|----|
| header field name | : | value | cr | If | | |
| header field name | : | value | cr | If | | |
| : | | | | | | |
| header field name | : | value | cr | If | | |
| cr | If | | | | | |
| Data | | | | | | |

sp: space

cr/lf: carriage return/line feed

Request line: necessary part, e.g.,

GET server.name/path/file.type

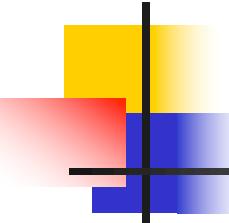
Header lines: optionally, further information to the host/document, e.g.

Host: www.fu-berlin.de

Accept-language: fr

User-agent: Opera /5.0

Entity Body: optionally. Further data, if the Client transmits data (POST method)



Khuôn dạng bản tin yêu cầu HTTP

- GET: được sử dụng khi trình duyệt yêu cầu một đối tượng, đối tượng yêu cầu được xác định trong trường URL
- POST: được sử dụng khi người sử dụng điền vào khuôn mẫu (form), ví dụ khi người sử dụng cung cấp từ khoá tìm kiếm cho công cụ tìm kiếm
- HEAD: tương tự như GET. Máy chủ nhận đáp ứng bằng bản tin HTTP nhưng bỏ qua đối tượng yêu cầu, thường dùng để gỡ rối (debugging)
- PUT: được sử dụng kết hợp với các công cụ phát hành Web, cho phép người dùng tải lên một đối tượng tới một thư mục cụ thể trên một máy chủ Web
- DELETE: cho phép người sử dụng hoặc ứng dụng xoá một đối tượng trên máy chủ Web

Khuôn dạng bản tin HTTP

HTTP/1.0

- GET
- POST
- HEAD

HTTP/1.1

- GET, POST, HEAD
- PUT
- DELETE

Khuôn dạng bản tin đáp ứng HTTP

| version | sp | status code | sp | phrase | cr | If |
|-------------------|----|-------------|----|--------|----|----|
| header field name | : | value | cr | If | | |
| header field name | : | value | cr | If | | |
| | : | | | | | |
| | : | | | | | |
| header field name | : | value | cr | If | | |
| cr | If | | | | | |
| Data | | | | | | |

Entity Body: inquired data

HEAD method: the server answers, but does not transmit the inquired data (debugging)

Status LINE: status code and phrase indicate the result of an inquiry and an associated message, e.g.

200 OK

400 Bad Request

404 Not Found

Groups of status messages:

1xx: Only for information

2xx: Successful inquiry

3xx: Further activities are necessary

4xx: Client error (syntax)

5xx: Server error

Khuôn dạng bản tin đáp ứng HTTP

status line

(protocol status code status phrase)

HTTP/1.1 200 OK

header
lines

Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998
Content-Length: 6821
Content-Type: text/html

data, e.g.,
requested
HTML file

data data data data data ...

Khuôn dạng bản tin đáp ứng HTTP

Trong dòng đầu bản tin phản hồi server → client
Một vài mã mẫu:

200 OK

- Yêu cầu thành công, đối tượng yêu cầu xuất hiện phía sau trong bản tin này.

301 Moved Permanently

- Đã bỏ đổi tượng yêu cầu, vị trí mới sẽ được đặc tả phía sau trong bản tin này (Location:)

400 Bad Request

- Server không hiểu bản tin yêu cầu

404 Not Found

- Tài liệu yêu cầu không tìm thấy trong server này

505 HTTP Version Not Supported

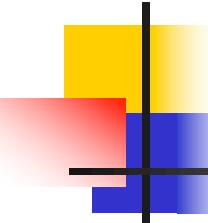
- Phiên bản giao thức HTTP yêu cầu không được máy chủ hỗ trợ.

Bài tập bản tin HTTP 1

Consider the following string of ASCII characters that were captured by Wireshark when the browser sent an HTTP GET message (i.e., this is the actual content of an HTTP GET message). The characters *<cr><lf>* are carriage return and line-feed characters (that is, the italicized character string *<cr>* in the text below represents the single carriage-return character that was contained at that point in the HTTP header). Answer the following questions, indicating where in the HTTP GET message below you find the answer.

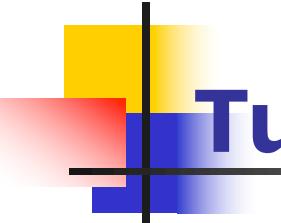
```
GET /cs453/index.html HTTP/1.1<cr><lf>Host: gai  
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (  
Windows; U; Windows NT 5.1; en-US; rv:1.7.2) Gec  
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept: ex  
t/xml, application/xml, application/xhtml+xml, text  
/html;q=0.9, text/plain;q=0.8, image/png,*/*;q=0.5  
<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>Accept-  
Encoding: zip,deflate<cr><lf>Accept-Charset: ISO  
-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>  
<lf>Connection:keep-alive<cr><lf><cr><lf>
```

- a. What is the URL of the document requested by the browser?
- b. What version of HTTP is the browser running?
- c. Does the browser request a non-persistent or a persistent connection?
- d. What is the IP address of the host on which the browser is running?
- e. What type of browser initiates this message? Why is the browser type needed in an HTTP request message?



Tương tác người dùng-máy chủ: Cookie

- ❖ Máy chủ HTTP không lưu trạng thái, làm đơn giản hóa thiết kế và cho phép các máy chủ Web có hiệu năng cao, có thể xử lý đồng thời hàng nghìn kết nối TCP.
- ❖ Tuy nhiên, không phù hợp với các ứng dụng Web như online shopping: yêu cầu lưu giữ thông tin của người dùng như ID, các mặt hàng mà người dùng đã chọn trước đó,...
- ❖ Với những mục đích này, HTTP sử dụng cookie. Các cookie được định nghĩa trong RFC 2965.



Tương tác người dùng-máy chủ: Cookie

- ❖ Cookie là **1 file text nhỏ** được **server tạo ra** và **gửi tới browser**, được lưu trữ trong máy tính người dùng một cách tạm thời hoặc cố định, nó cho phép các trang web **nhận ra** và **bám vết người sử dụng**.
- ❖ Cookie có **chứa URL của trang web** đã đặt cookie, không chứa bất cứ thông tin nào về người dùng hoặc máy của người dùng.
- ❖ Khi người dùng quay lại trang web, trình duyệt sẽ tìm kiếm cookies. Nếu **URL của cookies khớp với URL của trang web**, trang web sẽ truy xuất thông tin máy chủ bằng cách sử dụng thông tin lấy được từ cookies.

Tương tác người dùng-máy chủ: Cookie

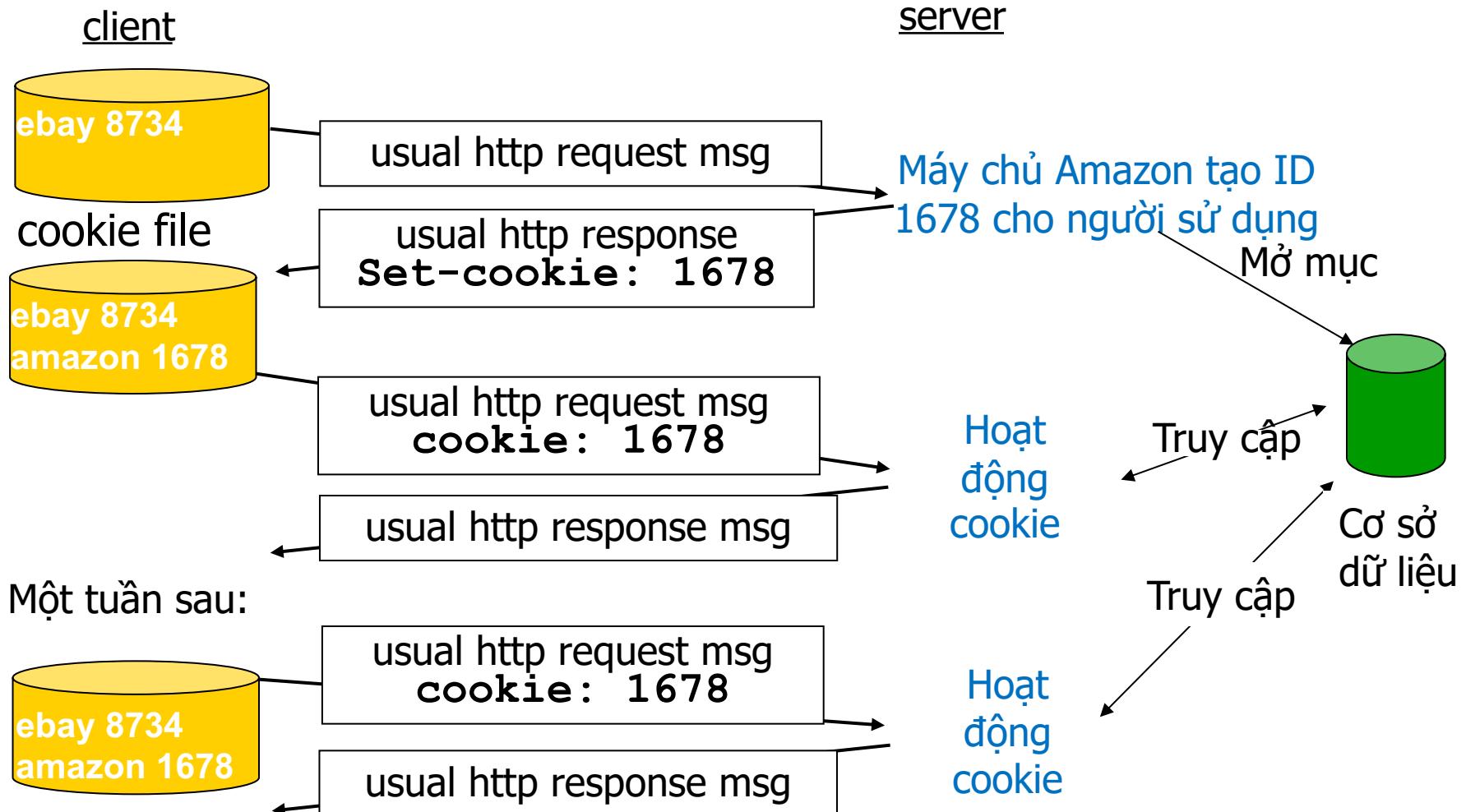
Ví dụ:

- Susan thường truy nhập Internet từ PC
- Lần đầu vào một trang thương mại điện tử cụ thể
- Khi khởi động yêu cầu HTTP tới trang này thì trang này sẽ tạo ra:
 - ID duy nhất
 - Một mục trong cơ sở dữ liệu đầu xa cho ID đó

Cách xem cookie: Mở Google **Chrome** > Nhấn vào biểu tượng 3 chấm dọc ở góc trên bên phải > Nhấn vào Cài đặt > Nhấn vào Quyền riêng tư và bảo mật > Nhấn vào Cài đặt trang web > Kéo xuống phía dưới > Nhấn vào **Cookie** và dữ liệu trang web > Kéo xuống phía dưới > Nhấn vào **Xem tất cả cookie** và dữ liệu trang web

Tương tác người dùng-máy chủ: Cookie

Cookies: giữ trạng thái người sử dụng



Tương tác người dùng-máy chủ: Cookie

Cookies có thể mang lai :

- Nhận dạng người dùng
- Mua bán chỉ qua một cú nhấp chuột
- Khuyến nghị người dùng
- Tạo lớp phiên người sử dụng trên đầu của HTTP phi trạng thái

Bên cạnh đó

Cookies và sự riêng tư:

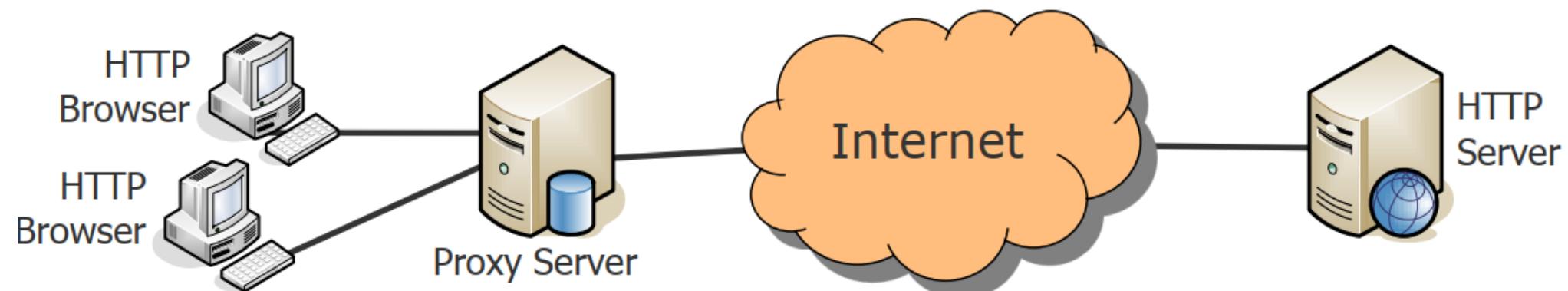
- cookies cho phép các trang web biết nhiều về bạn
- Có thể bạn sẽ cung cấp tên và địa chỉ email cho các trang này
- Xâm phạm tính riêng tư vì có thể bị bán thông tin cho bên thứ ba

Làm thế nào để giữ “trang thái”:

- Điểm cuối giao thức: duy trì trạng thái ở phía gửi/nhận qua các giao dịch
- cookies: bản tin http mang trạng thái

Lưu đệm Web (Web caching)

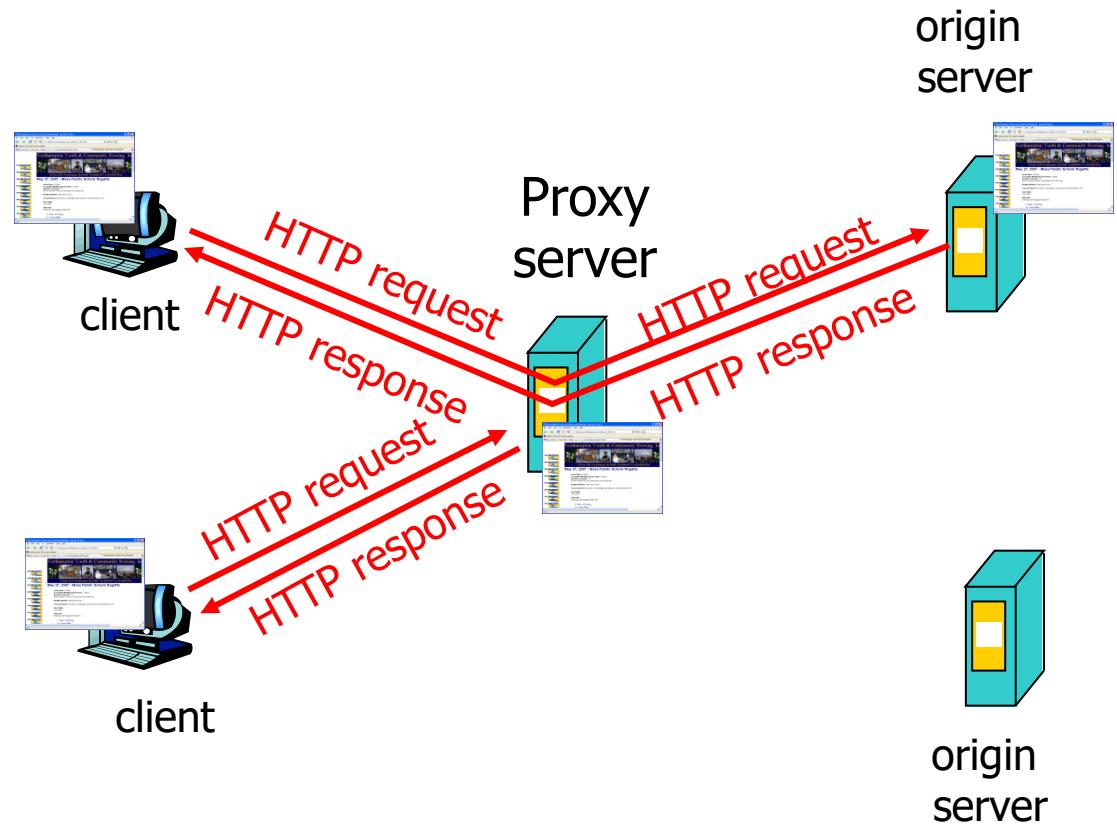
- **Máy chủ đệm Web** (Web Cache) hay **máy chủ proxy**: là một phần tử mạng thỏa mãn các yêu cầu HTTP khi đại diện cho máy chủ Web gốc.
- Máy chủ đệm Web có kho lưu trữ riêng và giữ các bản sao của các đối tượng được yêu cầu gần đây trong kho lưu trữ này.

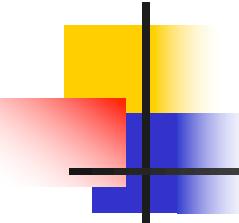


Lưu đệm Web (Web caching)

Mục đích/thành quả: đáp ứng yêu cầu của client mà không cần sự tham gia của server ban đầu

- Người sử dụng thiết lập trình duyệt: truy nhập web qua cache (máy chủ đệm là nơi lưu trữ/kho)
- Trình duyệt sẽ gửi toàn bộ yêu cầu HTTP tới cache
 - Đối tượng trong cache: cache sẽ trả về các đối tượng
 - Nếu không thì cache sẽ yêu cầu đối tượng từ server ban đầu, sau đó trả đối tượng cho client và lưu bản sao trong bộ lưu trữ nội bộ của mình





Lưu đệm Web (Web caching)

- Cache hoạt động như client và server
- Thường thì cache do ISP khởi tạo (trường đại học, công ty, VNPT,...)

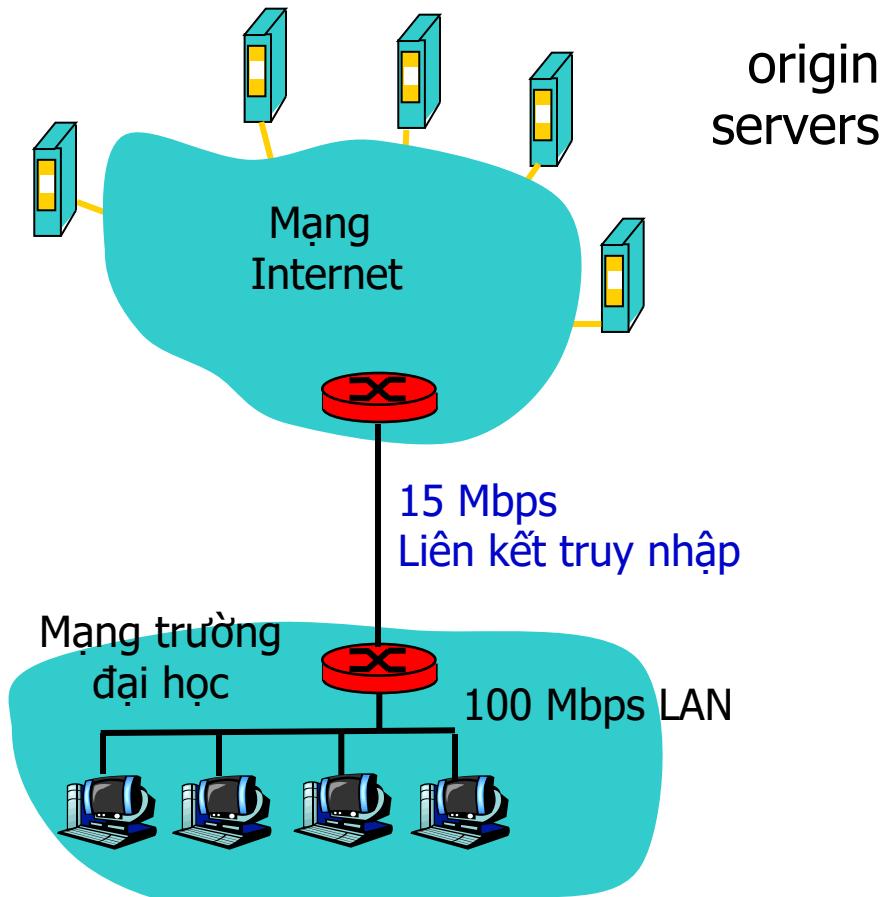
Vì sao lại lưu đệm Web?

- Để giảm thời gian đáp ứng yêu cầu của client
- Để làm giảm lưu lượng trên liên kết truy nhập của tổ chức tới Internet, giảm chi phí mua thêm băng thông kết nối.
- Các máy chủ đệm Web có thể làm giảm đáng kể lưu lượng Web trên tổng thể mạng Internet, do vậy nó sẽ cải thiện hiệu năng cho tất cả các ứng dụng.

Lưu đệm Web (Web caching)

Giả thiết

- Kích thước trung bình của đối tượng là 1Mb
- Tốc độ yêu cầu trung bình từ các trình duyệt của trường tới server gốc: 15 yêu cầu/giây
- Trễ Internet (từ router của trường tới bất kỳ server gốc nào và phản hồi về): 2 giây
- Tốc độ liên kết truy nhập: 15Mbps
- Tốc độ mạng LAN: 100 Mbps
- Giả sử các bản tin yêu cầu HTTP có kích thước rất nhỏ và sẽ không tạo lưu lượng đáng kể nào trong mạng



Lưu đệm Web (Web caching)

Kết quả là

- Tải trọng (mật độ) lưu lượng trên LAN là:

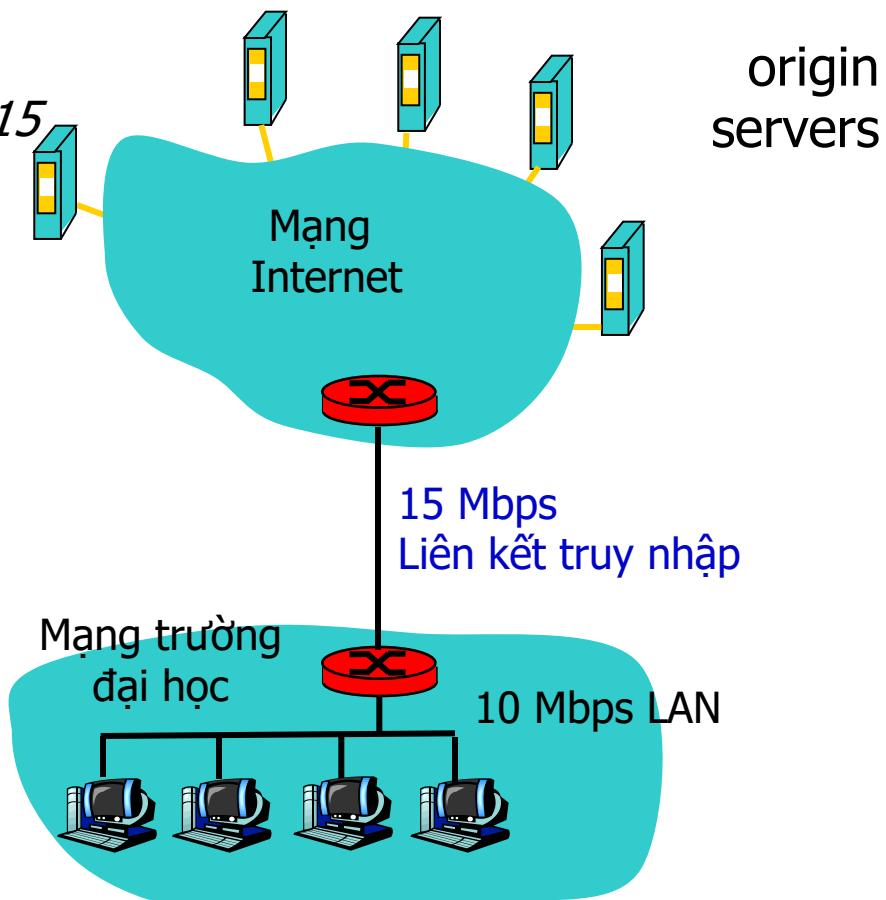
$(15 \text{ yêu cầu/giây}) \times (1\text{Mbit/ yêu cầu}) / (100\text{Mbit/s}) = 0,15$
(15% nên trễ mạng LAN là rất nhỏ, cỡ milli giây)

- Tải trọng lưu lượng trên liên kết truy nhập:

$(15 \text{ yêu cầu/giây}) \times (1\text{Mbit/ yêu cầu}) / (15\text{Mbit/s}) = 1$
(100%)

⇒ trễ trên liên kết trở nên rất lớn và có thể tăng không giới hạn

- Tổng trễ = trễ Internet + trễ truy nhập + trễ LAN
 $= 2 \text{ sec} + \text{minutes} + \text{milliseconds}$



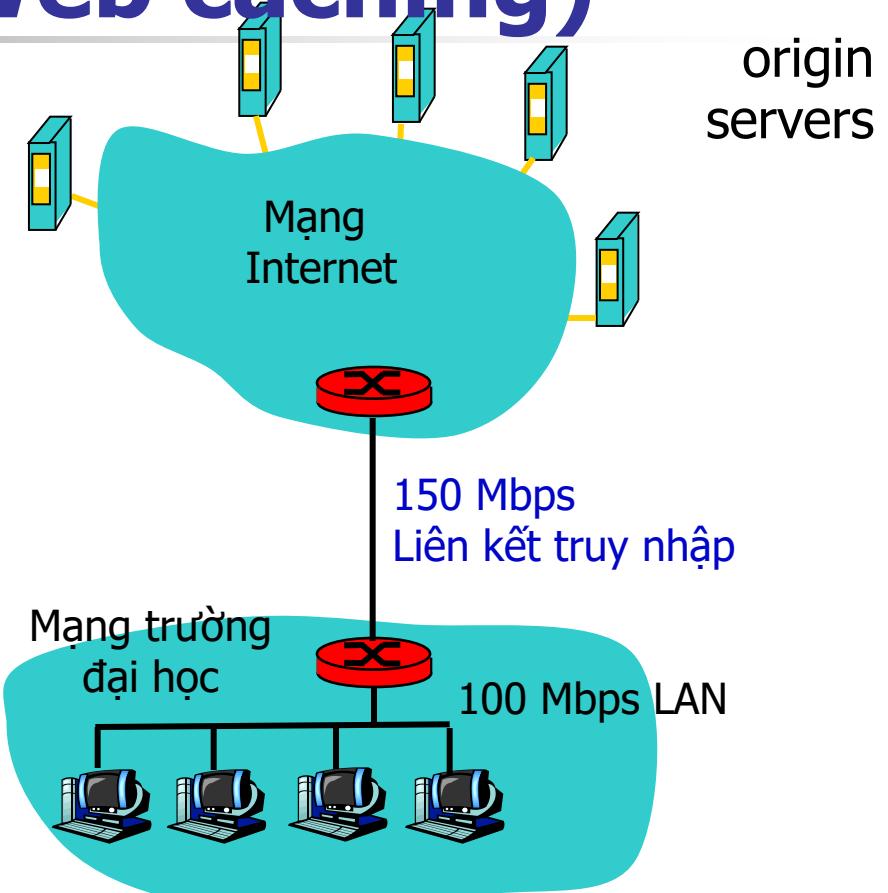
Lưu đệm Web (Web caching)

Giải pháp khả thi 1

- Tăng băng thông đường truy nhập, giả dụ là 150 Mbps

Kết quả

- Mật độ lưu lượng trên LAN = 15%
- Mật độ lưu lượng trên đường truy nhập= 10%
- Tổng trễ= Internet delay + access delay + LAN delay
= 2 sec + msec + msec
- Chi phí để nâng băng thông là khá đắt



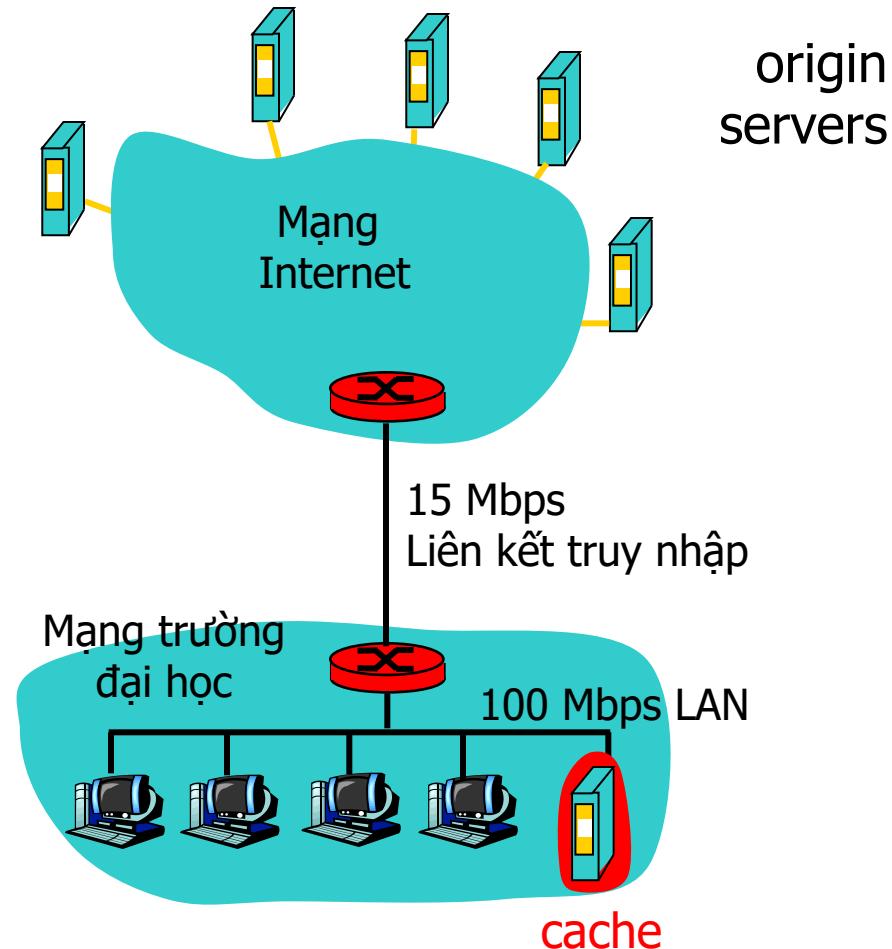
Lưu đệm Web (Web caching)

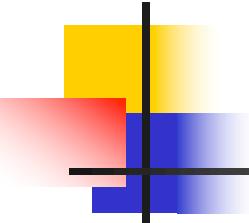
Giải pháp khả thi 2: lắp đặt cache

- Giả sử tỷ lệ truy vấn vào cache là 0,4 (40% yêu cầu được thực hiện tại cache, 60% yêu cầu do server gốc đáp ứng)

Kết quả

- 40% yêu cầu gần như thoả mãn tức thì
- 60% yêu cầu sử dụng liên kết truy nhập
- Độ sử dụng trên liên kết truy nhập giảm xuống còn 60% ($0.6 * 15 / 15$), dẫn đến trễ giảm xuống rất nhỏ (10 msec)
- Tổng trễ trung bình = Internet delay + access delay + LAN delay
 $= (0.6 * 2.01 \text{secs} + 0.4 * 0.01 \text{secs})$
 $\approx 1.21 \text{ secs}$



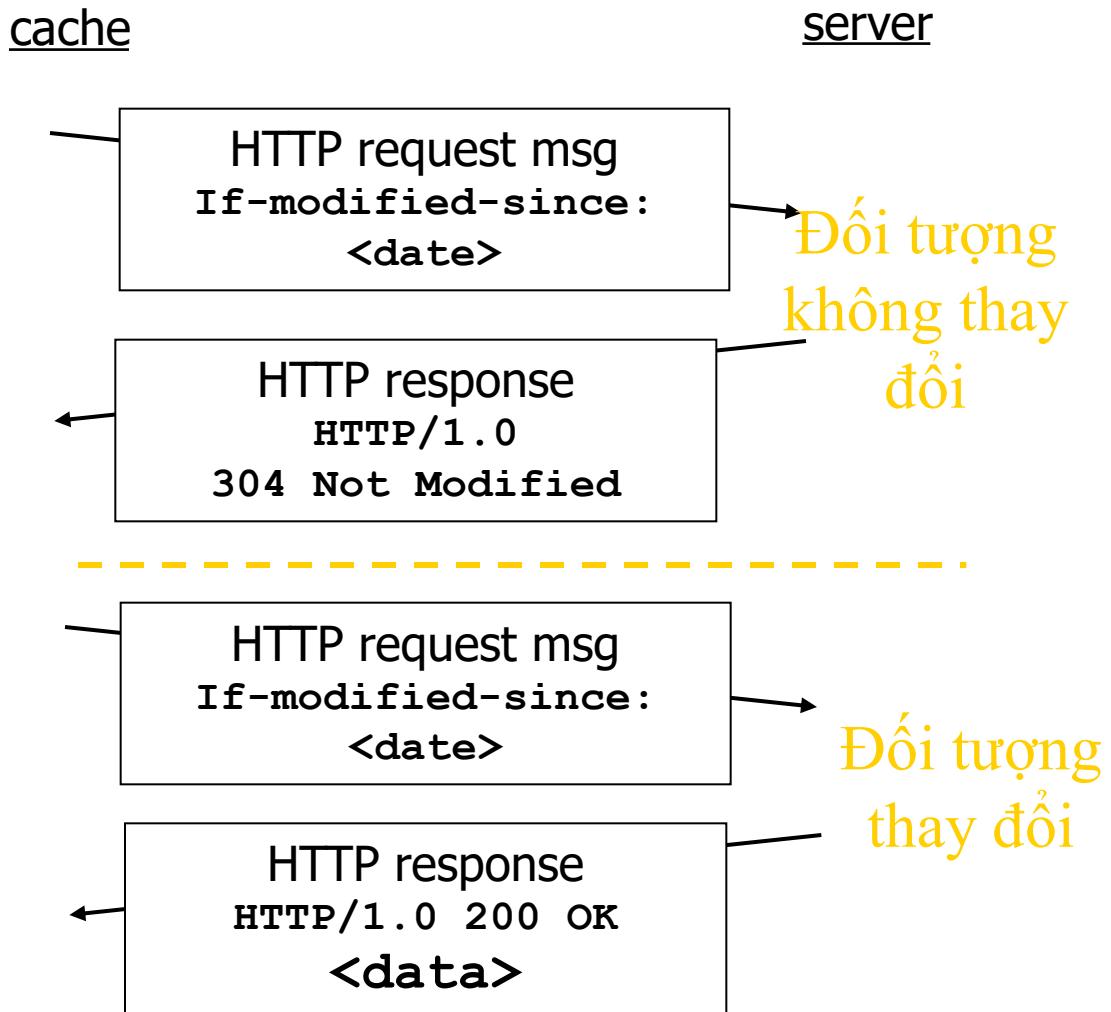


Bản tin GET có điều kiện

- Mặc dù việc đệm Web giảm được thời gian đáp ứng cho người sử dụng nhưng nó lại đặt ra một khó khăn mới, đó là bản sao của các đối tượng nằm trong các máy chủ đệm có thể bị cũ.
- Nói cách khác, đối tượng ở các máy chủ Web đã thay đổi so với thời điểm mà máy chủ đệm sao chép nó và chuyển tới máy khách.
- HTTP có cơ chế cho phép máy chủ đệm tra cứu việc đối tượng đã được cập nhật hay chưa. Cơ chế này được gọi là cơ chế GET có điều kiện (conditional GET).
- Một bản tin yêu cầu HTTP được gọi là bản tin conditional GET nếu (1) bản tin yêu cầu sử dụng phương thức GET và (2) bản tin yêu cầu bao gồm một dòng tiêu đề If-Modified-Since:..

Bản tin GET có điều kiện

- HTTP có cơ chế cho phép tra cứu việc đổi tượng trong cache đã được cập nhật hay chưa
- Nó sẽ không gửi đổi tượng nếu cache đã cập nhật phiên bản mới.
- cache: chỉ ra ngày của bản sao cached trong yêu cầu HTTP
If-modified-since: <date>
- server: phản hồi nhưng không gửi đổi tượng nào nếu copy của cache là cập nhật :
HTTP/1.0 304 Not Modified



Bản tin GET có điều kiện

cache

GET /fruit/kiwi.gif HTTP/1.1

Host: www.exotiquecuisine.com

1 tuần sau:

GET /fruit/kiwi.gif HTTP/1.1

Host: www.exotiquecuisine.com

If-Modified-Since: Wed, 4 Jul 2007 09:23:24

server

HTTP/1.1 200 OK

Date: Sat, 7 Jul 2007 15:39:29

Server: Apache/1.3.0 (Unix)

Last-Modified: Wed, 4 Jul 2007 09:23:24

Content-Type: image/gif

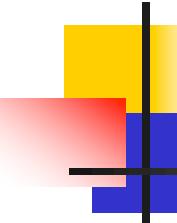
(data data data data data ...)

HTTP/1.1 304 Not Modified

Date: Sat, 14 Jul 2007 15:39:29

Server: Apache/1.3.0 (Unix)

(empty entity body)



Chương 2: Các ứng dụng và giao thức

- 2.1 Tổng quan về các ứng dụng và dịch vụ hạ tầng
- 2.2 Ứng dụng WEB và các giao thức

2.3 Ứng dụng truyền tệp và các giao thức

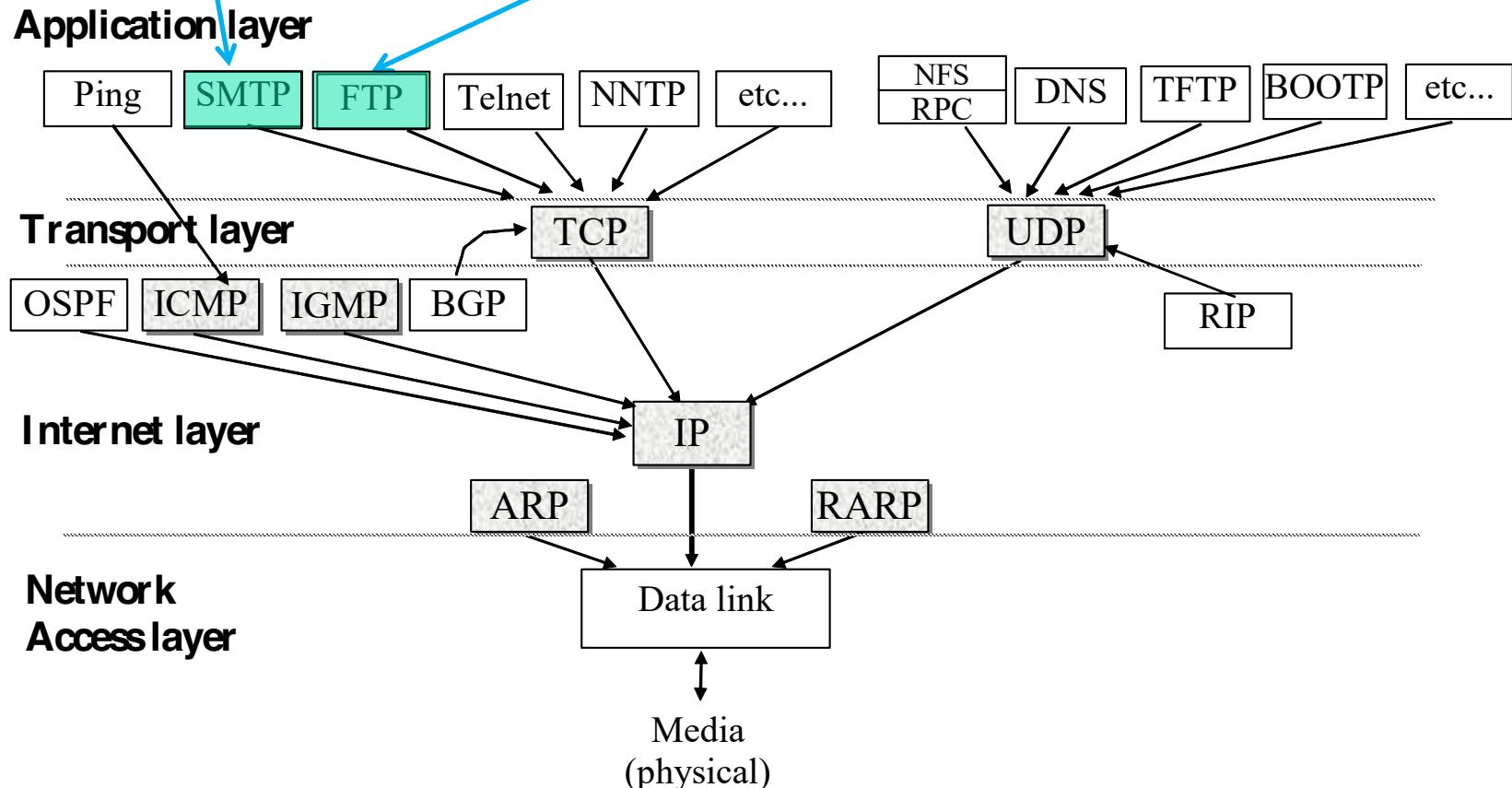
2.4 Ứng dụng thư điện tử và các giao thức

- 2.5 Ứng dụng hệ thống tên miền

- 2.6 Các ứng dụng mạng ngang hàng

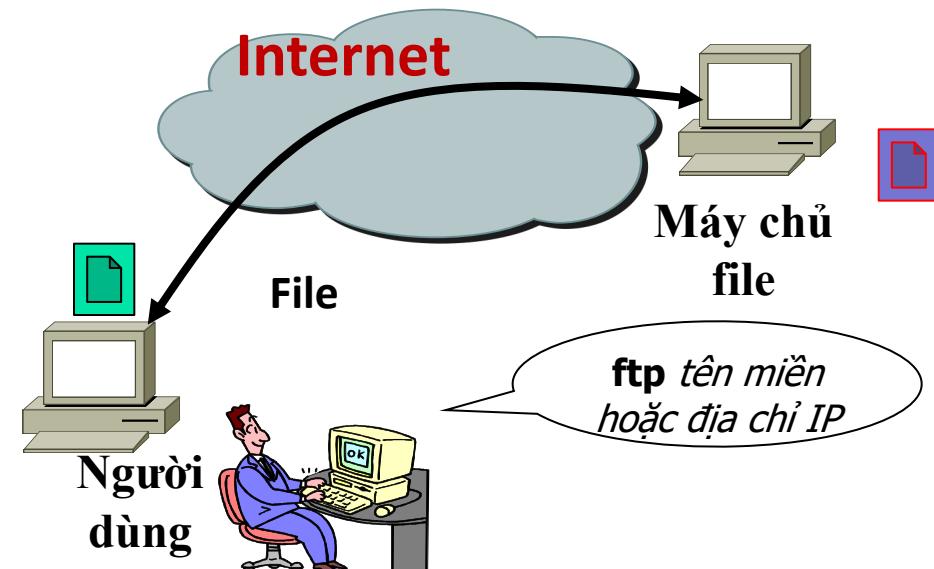
Truyền tệp và thư điện tử

- **SMTP: Simple Mail Transfer Protocol**
- **FTP: File Transfer Protocol**



Giao thức truyền tệp FTP

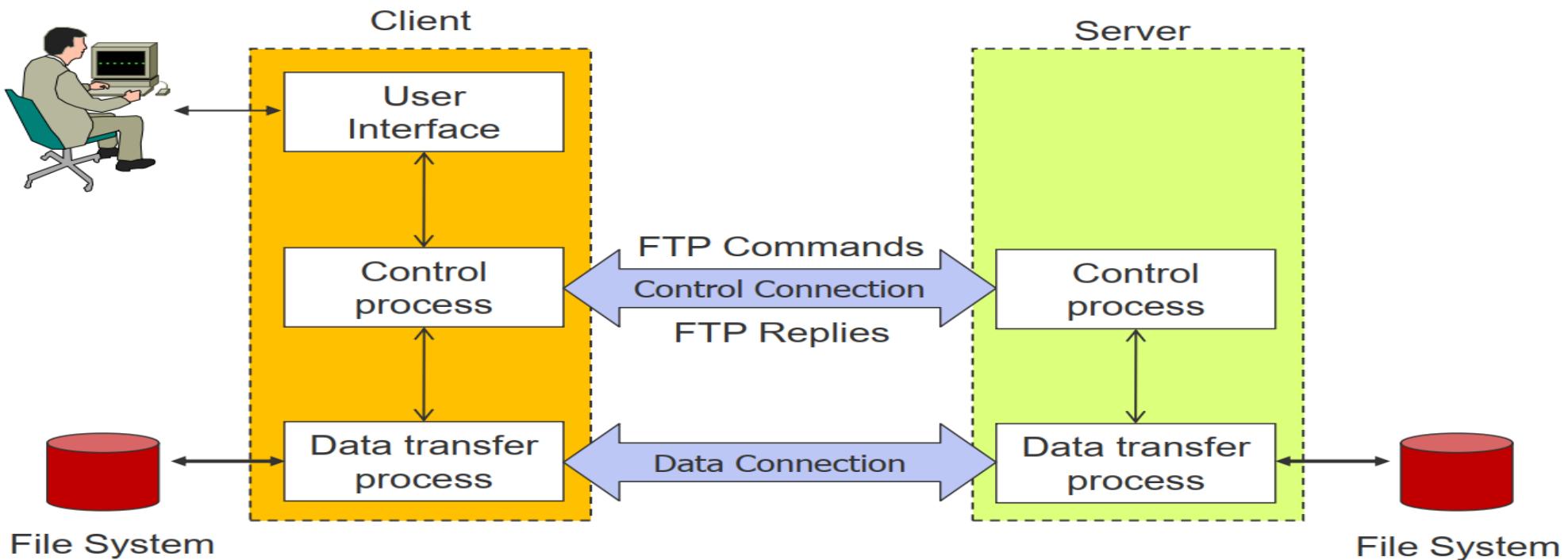
- Cho phép truyền file qua mạng (FTP-File Transfer Protocol)
- Người dùng ngồi trước màn hình (trạm cục bộ) và muốn truyền tệp tới một trạm chủ ở xa hoặc nhận tệp từ một trạm chủ ở xa.
- Để truy nhập vào tài khoản ở xa thì người sử dụng phải cung cấp nhận dạng cá nhân và mật khẩu.
- Sau đó, người dùng có thể truyền tệp từ hệ thống tệp cục bộ tới hệ thống tệp ở xa và ngược lại.

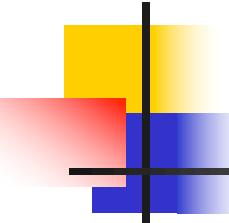


ftp: RFC 959
<http://www.ietf.org/rfc/rfc959.txt>

Giao thức truyền tệp FTP

- Hoạt động theo kiến trúc client/server, máy chủ chạy software cung cấp dịch vụ FTP, máy khách chạy software dùng cho người sử dụng
- FTP sử dụng điều khiển ngoài băng, một kết nối **điều khiển** (cổng 21) và một kết nối cho truyền **dữ liệu (data)** (cổng 20)

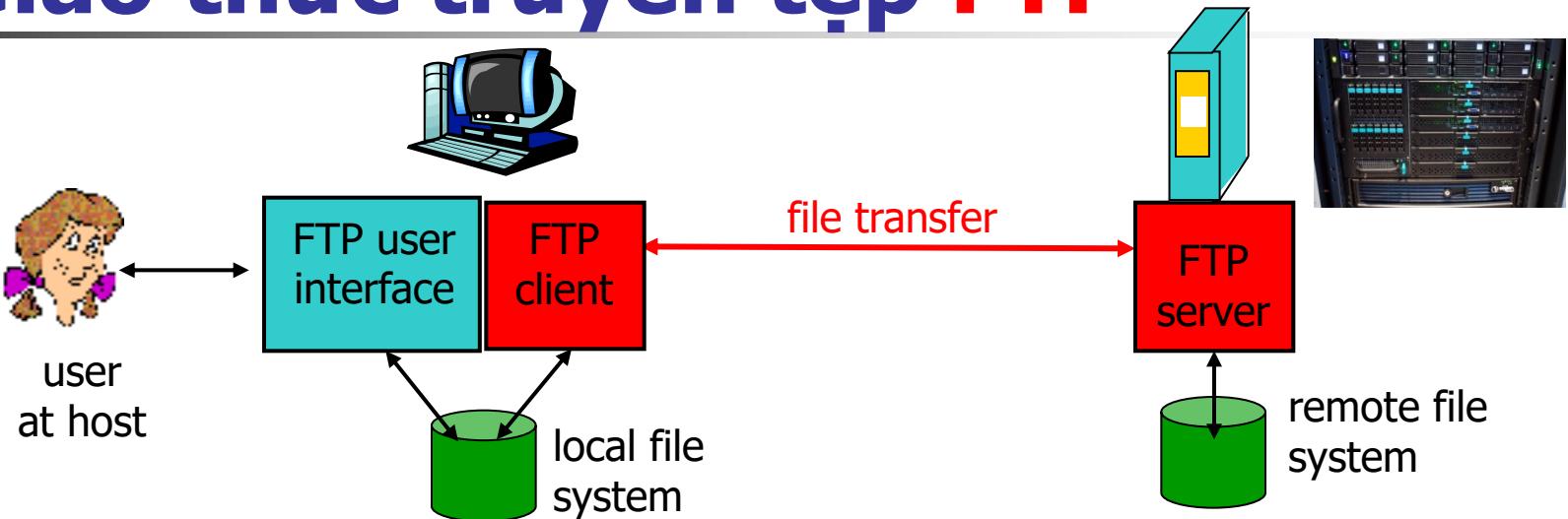




Giao thức truyền tệp FTP

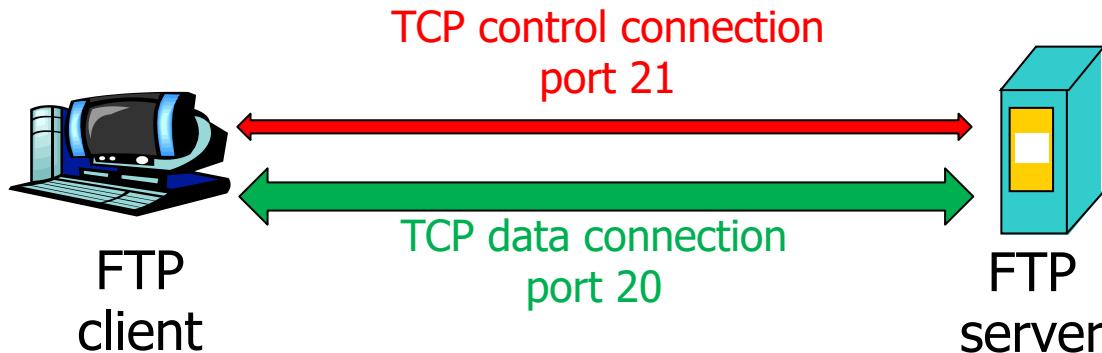
- Kết nối **điều khiển** (Control connection): truyền thông tin điều khiển, duy trì trong suốt phiên truyền dữ liệu
- Kết nối cho truyền **dữ liệu** (**data connection**): truyền dữ liệu, chỉ mở khi có lệnh truyền dữ liệu, và đóng khi truyền xong 1 file.
- Khi người sử dụng khởi tạo 1 phiên FTP, kết nối **điều khiển** được mở ra. Trong khi kết nối **điều khiển** được mở, kết nối **dữ liệu** có thể được mở và đóng nhiều lần nếu một vài file được truyền.

Giao thức truyền tệp FTP



- Truyền file từ/tới host ở xa: Người sử dụng có thể truy nhập file và các thư mục một cách tương tác trên máy chủ ở xa
 - ***Liệt kê (list)*** các file trong thư mục cục bộ ở xa
 - ***Đổi tên (rename)*** và ***xoá (delete)*** tập tin (nếu được phân quyền)
 - ***Chuyển file*** từ máy ở xa về máy cục bộ (***download***)
 - ***Truyền file*** từ máy cục bộ đến máy chủ ở xa (***upload***)
- Mô hình client/server
 - ***client:*** phía kích hoạt truyền (từ/ tới phía từ xa)
 - ***server:*** host ở xa

Giao thức truyền tệp FTP



- Người sử dụng cung cấp địa chỉ, mật khẩu. Máy khách FTP liên lạc với máy chủ FTP tại cổng 21, trên kết nối TCP.
- Máy khách được cấp phép trên kết nối điều khiển.
- Máy khách duyệt thư mục từ xa bằng lệnh gửi qua kết nối điều khiển.
- Khi máy chủ nhận được lệnh chuyển tệp, nó mở kết nối TCP thứ hai để truyền dữ liệu.
- Sau khi truyền xong 1 tệp, máy chủ đóng kết nối dữ liệu.
 - Máy chủ mở kết nối dữ liệu TCP khác để truyền tệp khác.
 - Kết nối điều khiển: ngoài dải "out of band"
 - Máy chủ FTP duy trì trạng thái: thư mục hiện thời, xác thực trước đó.

Giao thức truyền tệp FTP

Ví dụ các lệnh

- Gửi như văn bản ASCII qua kênh điều khiển
- **USER *username***
- **PASS *password***
- **LIST** trả lại danh sách file trong thư mục hiện thời
- **RETR *filename*** lấy một/nhiều file (get, mget)
- **STOR *filename*** lưu trữ (put, mput) một/nhiều file vào trạm chủ ở xa

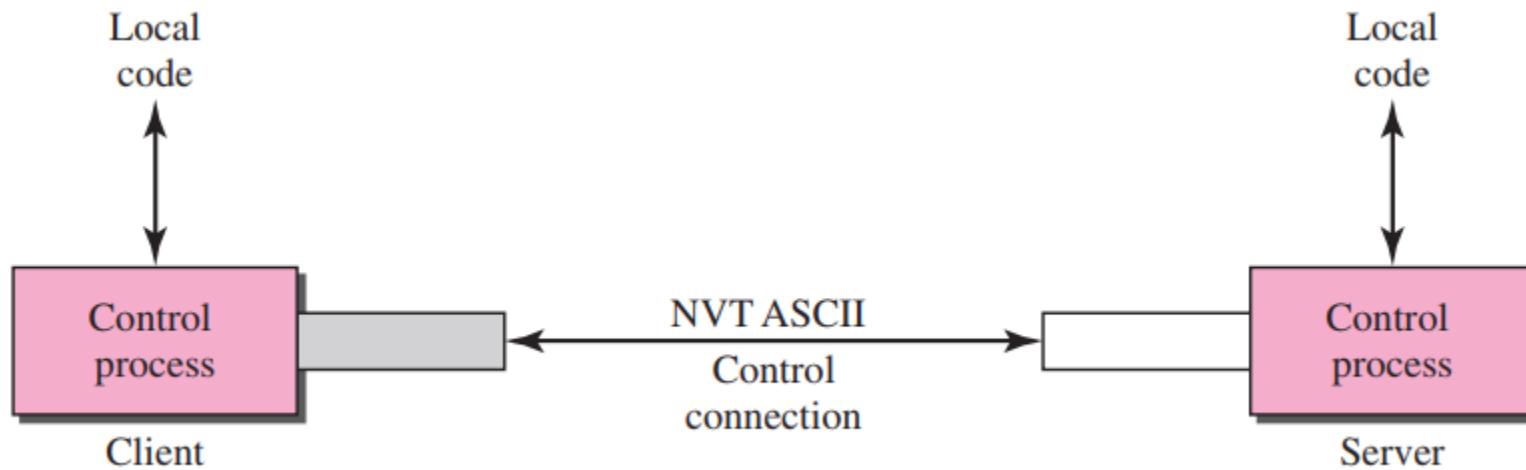
Ví dụ các mã trả về

- Mã trạng thái và câu giải thích (như HTTP)
 - **331 Username OK, password required**
 - **125 data connection already open; transfer starting**
 - **425 Can't open data connection**
 - **452 Error writing file**

Giao thức truyền tệp FTP

■ Kết nối điều khiển:

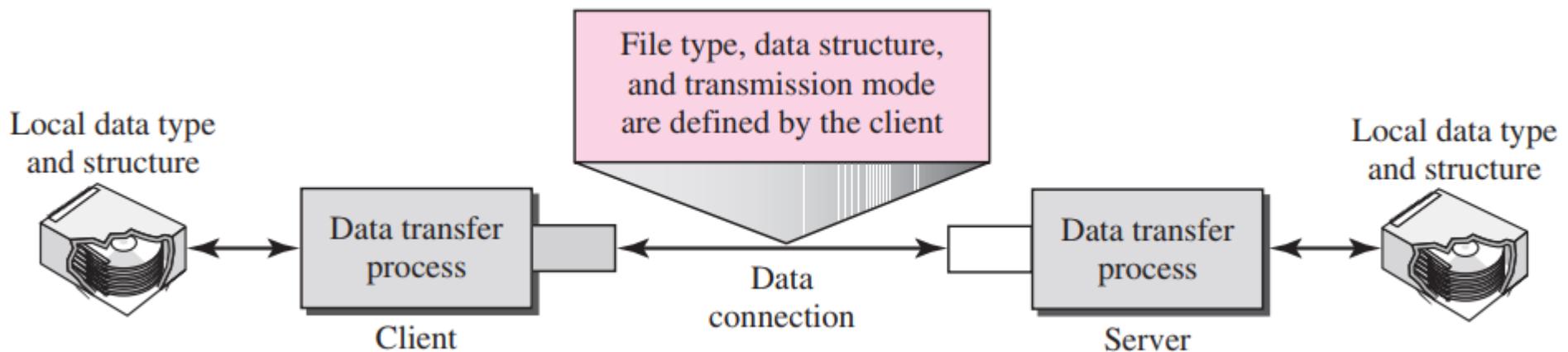
- Sử dụng các command và response, là dòng lệnh ngắn, đơn giản
- Một lệnh hoặc một bản tin đáp ứng được gửi tại 1 thời điểm
- Mỗi dòng lệnh sẽ được kết thúc bởi ký tự CR+LF

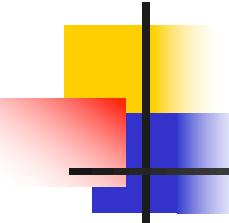


Giao thức truyền tệp FTP

■ Kết nối dữ liệu:

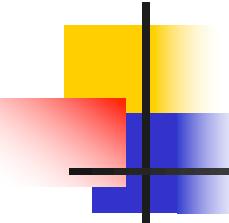
- Các file được truyền qua kết nối dữ liệu, dưới sự điều khiển của kết nối điều khiển
- Truyền file trong FTP: sử dụng lệnh RETR, STOR, LIST
- Client định nghĩa kiểu file được truyền, cấu trúc dữ liệu, và chế độ truyền





Giao thức truyền tệp FTP

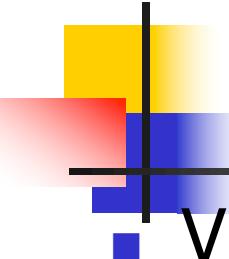
- Kiểu file:
 - ASCII file: text file với mỗi ký tự được mã hóa bởi mã ASCII 7 bit
 - EBCDIC file: định dạng file được sử dụng bởi IBM, file truyền đi được mã hóa bởi mã EBCDIC
 - Image file: truyền các file nhị phân, truyền liên tiếp luồng bit mà không cần mã hóa
- Cấu trúc dữ liệu:
 - file structure: continuous stream of bytes
 - record structure: file is divided into records
 - page structure: the file is divided into pages



Giao thức truyền tệp FTP

- Chế độ truyền:

- **Stream mode**: dữ liệu được chuyển xuống lớp TCP là luồng byte liên tục. TCP chịu trách nhiệm cắt luồng dữ liệu thành các segment có kích thước phù hợp
- **Block mode**: dữ liệu được chuyển xuống lớp TCP dưới dạng khối
- **Compressed mode**: nếu dữ liệu có kích thước lớn, nó sẽ được nén lại.



Giao thức truyền tệp FTP

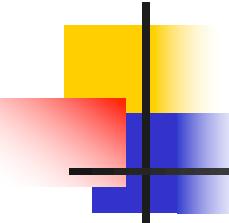
Ví dụ hoạt động của FTP:

1. After the control connection is created, the FTP server sends the 220 (service ready) response on the control connection.
2. The client sends its name.
3. The server responds with 331 (user name is OK, password is required).
4. The client sends the password (not shown).
5. The server responds with 230 (user log-in is OK).
6. The client sends the list command (ls reports) to find the list of files on the directory named report.
7. Now the server responds with 150 and opens the data connection.
8. The server then sends the list of the files or directories (as a file) on the data connection. When the whole list (file) is sent, the server responds with 226 (closing data connection) over the control connection.
9. The client now has two choices. It can use the QUIT command to request the closing of the control connection, or it can send another command to start another activity (and eventually open another data connection). In our example, the client sends a QUIT command.
10. After receiving the QUIT command, the server responds with 221 (service closing) and then closes the control connection.

```
$ ftp voyager.deanza.fhda.edu
Connected to voyager.deanza.fhda.edu.
220 (vsFTPd 1.2.1)
530 Please login with USER and PASS.
Name (voyager.deanza.fhda.edu:forouzan): forouzan
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls reports
227 Entering Passive Mode (153,18,17,11,238,169)
150 Here comes the directory listing.
```

| | | | | | |
|------------|---|------|-----|------------------|----------|
| drwxr-xr-x | 2 | 3027 | 411 | 4096 Sep 24 2002 | business |
| drwxr-xr-x | 2 | 3027 | 411 | 4096 Sep 24 2002 | personal |
| drwxr-xr-x | 2 | 3027 | 411 | 4096 Sep 24 2002 | school |

```
226 Directory send OK.
ftp> quit
221 Goodbye.
```



Giao thức truyền tệp FTP

FTP ẩn danh (Anonymous FTP):

- ❖ Để sử dụng FTP, người dùng cần có tài khoản và mật khẩu trên máy chủ từ xa.
- ❖ Một số trang web có sẵn một tập hợp các tệp để truy cập công khai, nhằm kích hoạt FTP ẩn danh.
- ❖ Để truy cập các tệp này, người dùng không cần phải có tài khoản hoặc mật khẩu. Thay vào đó, người dùng có thể sử dụng **anonymous** làm tên người dùng và **guest** làm mật khẩu.
- ❖ Quyền truy cập của người dùng vào hệ thống rất hạn chế. Một số trang web chỉ cho phép người dùng ẩn danh một tập hợp con các lệnh.
- ❖ Ví dụ: hầu hết các trang web đều cho phép người dùng sao chép một số tệp nhưng không cho phép điều hướng qua các thư mục.⁸²

Giao thức truyền tệp FTP

Ví dụ hoạt động của Anonymous FTP:

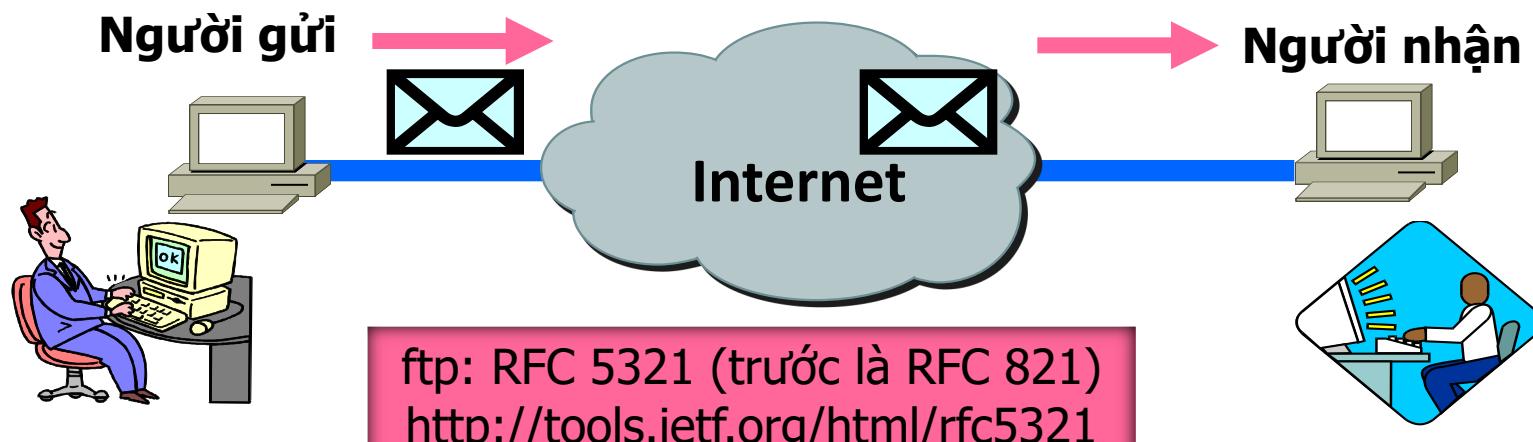
```
$ ftp internic.net
Connected to internic.net
220 Server ready
Name: anonymous
331 Guest login OK, send "guest" as password
Password: guest
ftp > pwd
257 '/' is current directory
ftp > ls
200 OK
150 Opening ASCII mode
```

```
bin
...
...
...
```

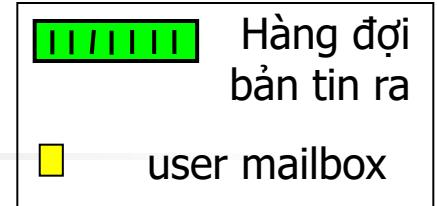
```
ftp > close
221 Goodbye
ftp > quit
```

Thư điện tử (Email)

- Dịch vụ phổ biến nhất
- Nguyên tắc “lưu và chuyển tiếp” (store and forward)
- Người dùng cần có tài khoản (account) thư
- Gửi, nhận thư: Outlook Express; Eudora
- Thư điện tử trên Web: yahoo.com, hotmail.com, gmail.com, mail.ptit.edu.vn

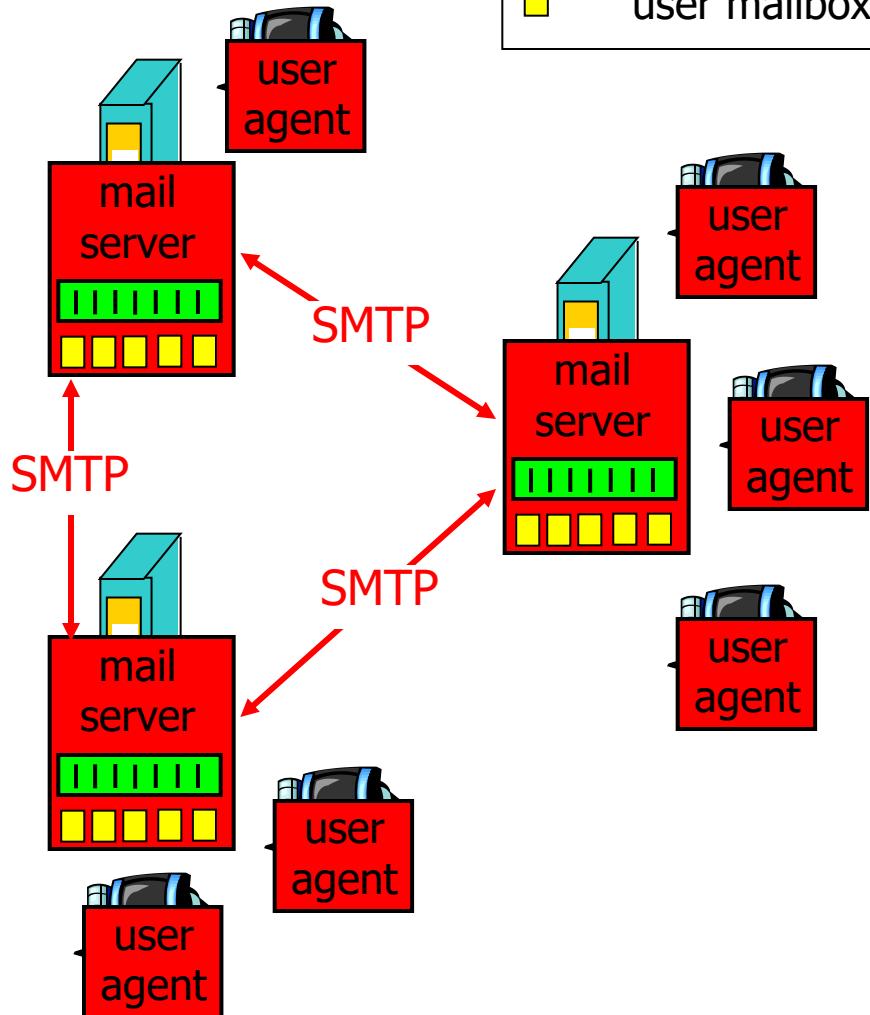


Thư điện tử (Email)



Ba thành phần cơ bản:

- user agents (đại lý người sử dụng)
- mail servers (máy chủ thư)
- Simple Mail Transfer Protocol: **SMTP** (giao thức truyền thư đơn giản)



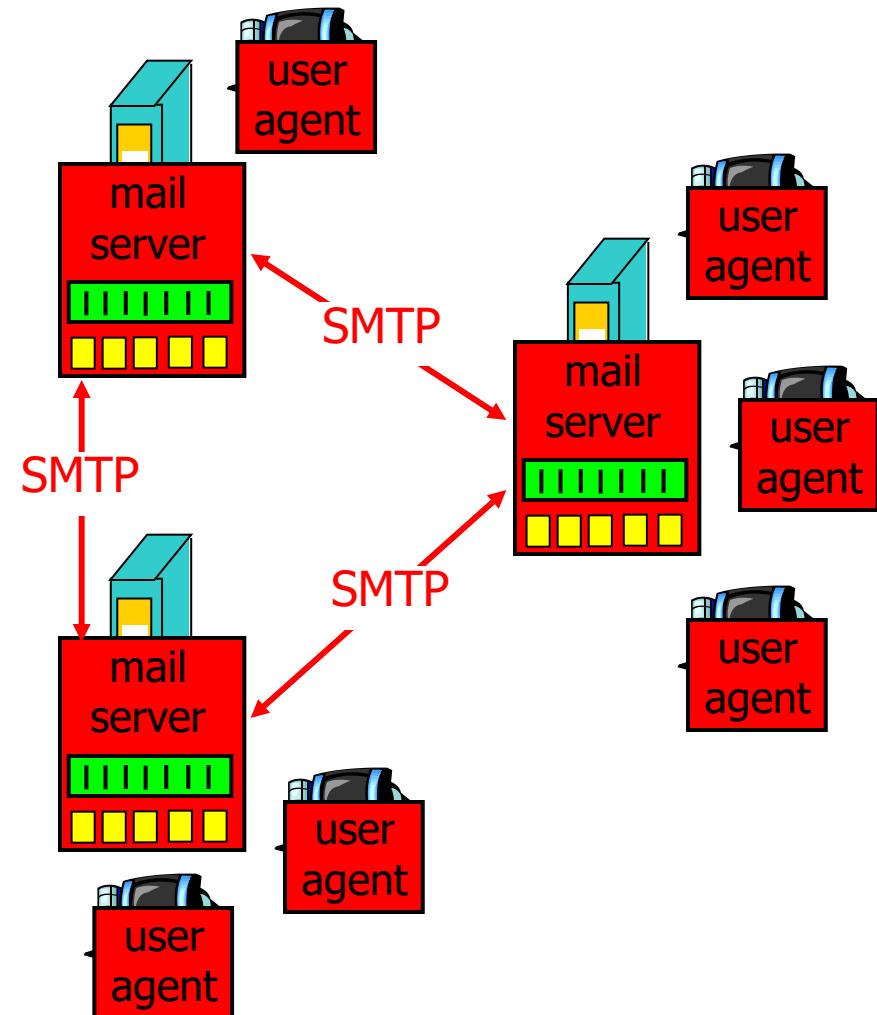
User Agent

- Soạn thư
- Đọc thư
- Trả lời và chuyển tiếp thư
- Ví dụ: Eudora, Outlook, Mozilla Thunderbird (GUI based UA), mail, pine, elm (Command driven UA)

Thư điện tử (Email)

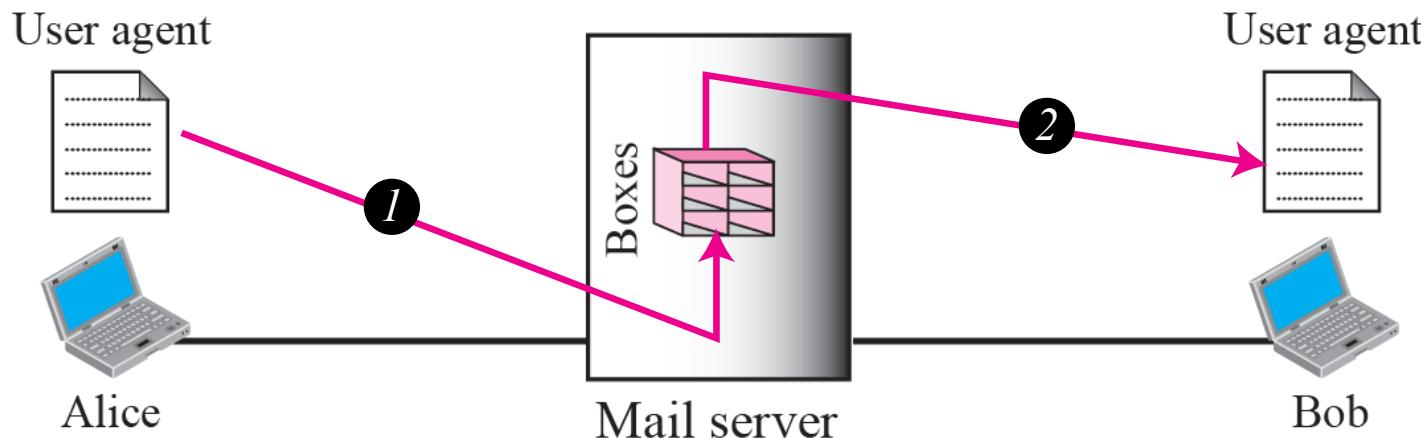
Mail Servers

- **mailbox** chứa bản tin gửi tới người sử dụng
- **message queue** hàng đợi thư đầu ra (sẽ được gửi đi)
- **SMTP protocol** giữa server thư để gửi thư điện tử
 - client: máy chủ gửi thư
 - “server”: máy chủ nhận thư



Thư điện tử (Email)

Kiến trúc: kịch bản 1

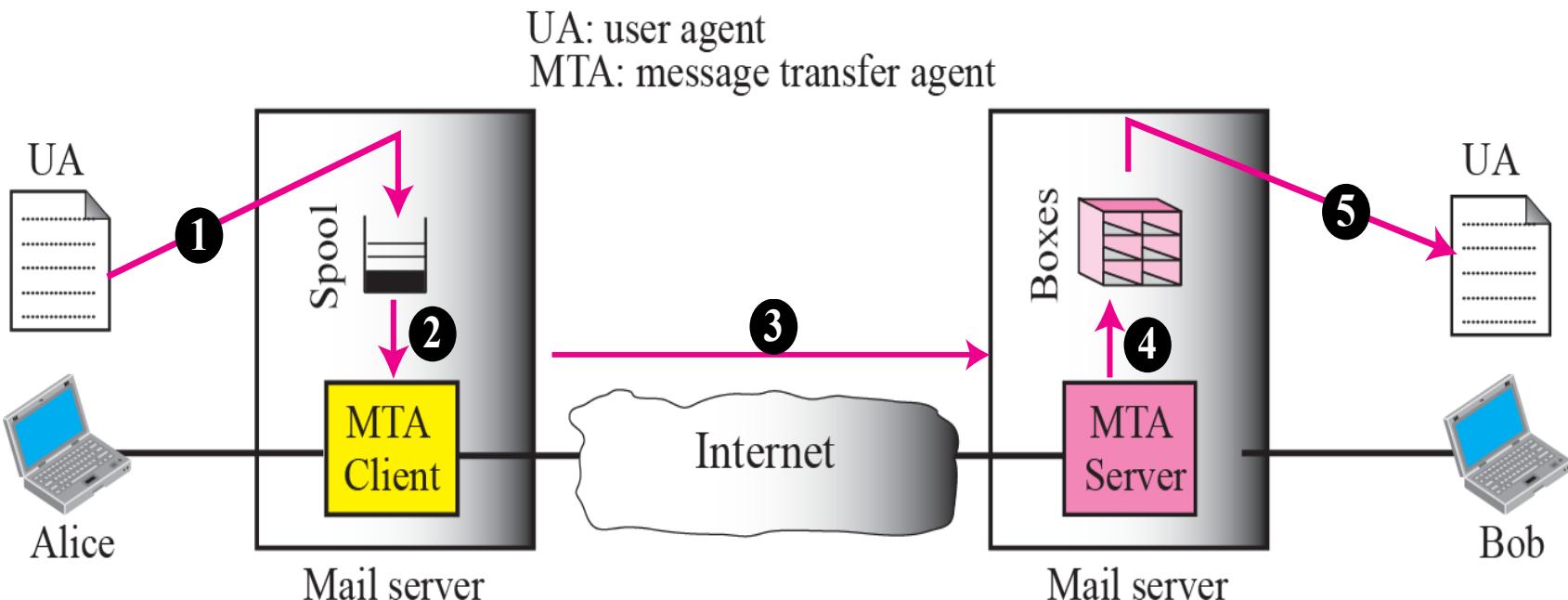


When the sender and the receiver of an e-mail are on the same mail server, we need only two user agents.

Thư điện tử (Email)

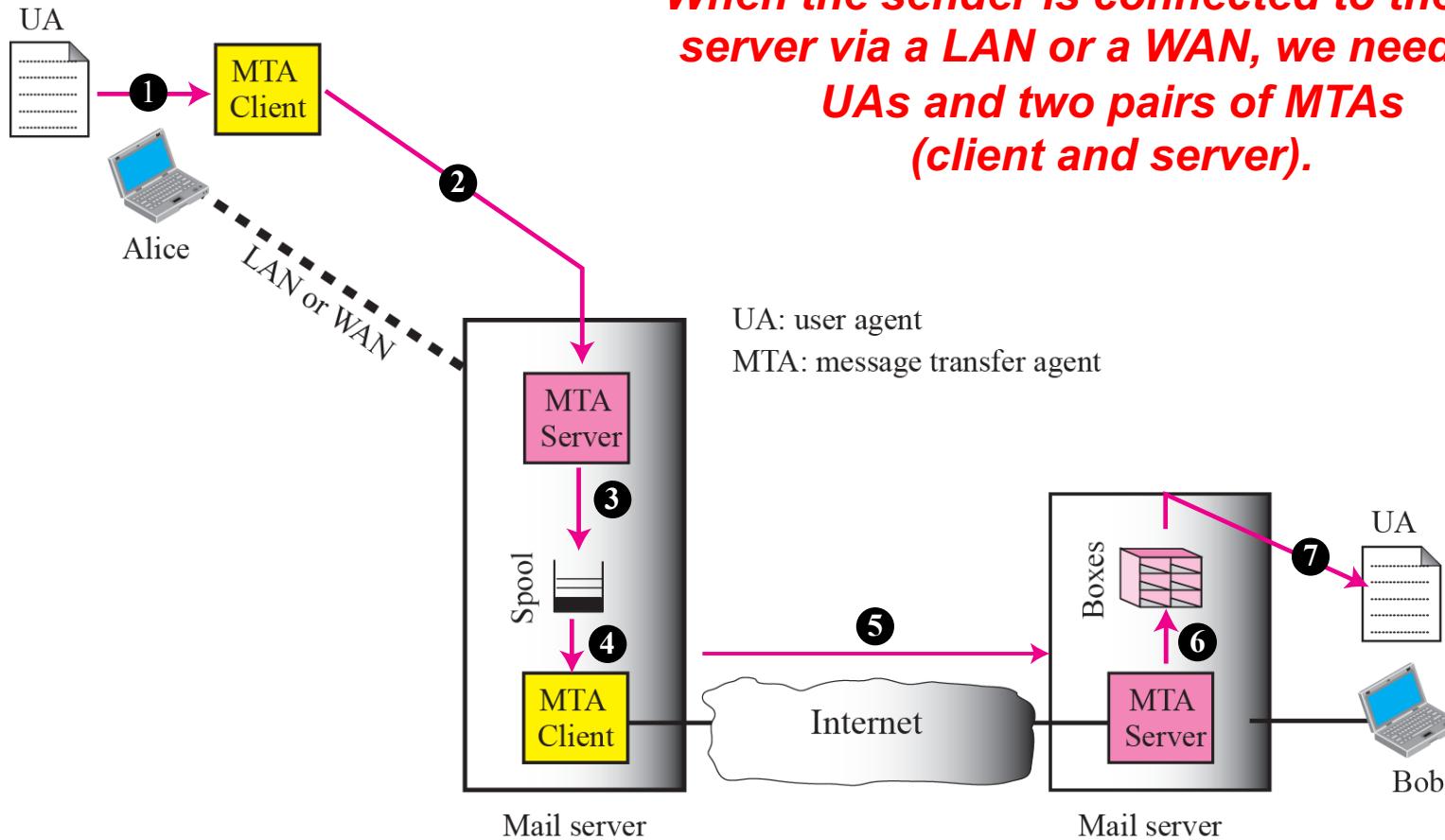
- Kiến trúc: kịch bản 2

When the sender and the receiver of an email are on different systems, we need two UAs and a pair of MTAs (client and server).



Thư điện tử (Email)

- Kiến trúc: kịch bản 3

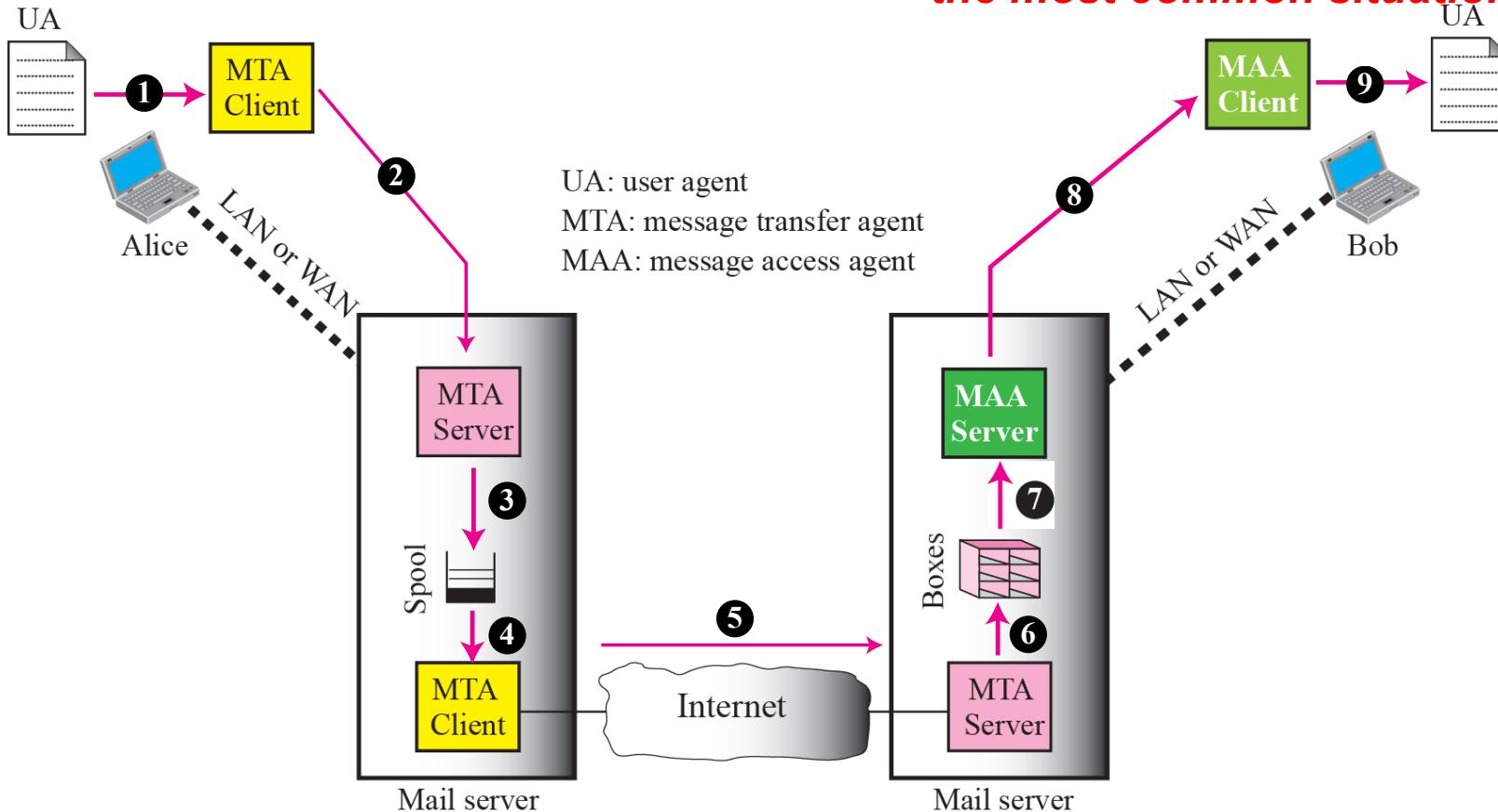


When the sender is connected to the mail server via a LAN or a WAN, we need two UAs and two pairs of MTAs (client and server).

Thư điện tử (Email)

- Kiến trúc: kịch bản 4

When both sender and receiver are connected to the mail server via a LAN or a WAN, we need two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server). This is the most common situation today.

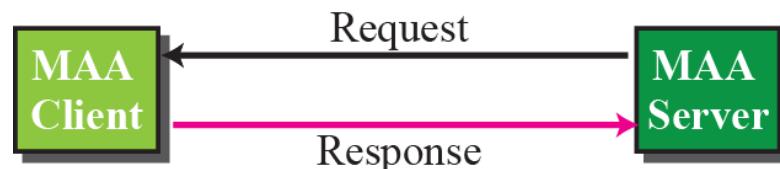


Thư điện tử (Email)

- Push và Pull



a. Client pushes messages



b. Client pulls messages

Thư điện tử (Email) Định dạng thư

Behrouz Forouzan
De Anza College
Cupertino, CA 96014

Sophia Fegan
Com-Net
Cupertino, CA 95014

Sophia Fegan
Com-Net
Cupertino, CA 95014
Jan. 5, 2005

Subject: Network

Dear Ms. Fegan:
We want to inform you that
our network is working pro-
perly after the last repair.

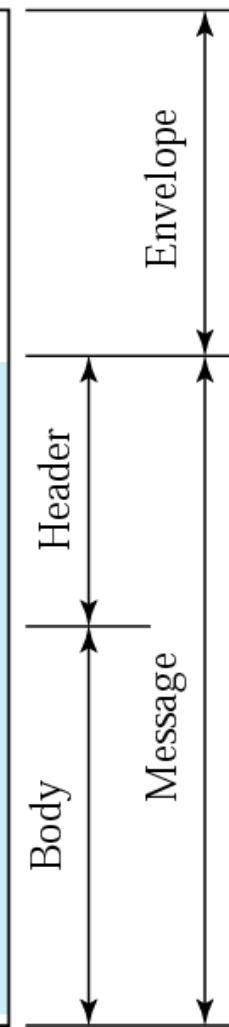
Yours truly,
Behrouz Forouzan

Mail From: forouzan@deanza.edu
RCPT To: fegan@comnet.com

From: Behrouz Forouzan
To: Sophia Fegan
Date: 1/5/05
Subject: Network

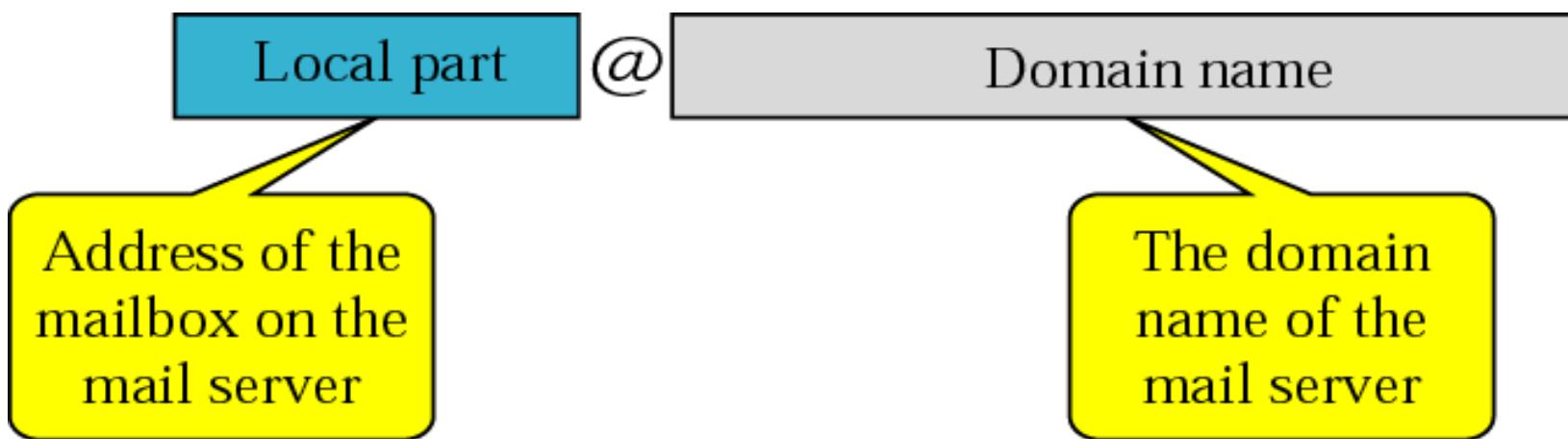
Dear Ms. Fegan:
We want to inform you that
our network is working pro-
perly after the last repair.

Yours truly,
Behrouz Forouzan



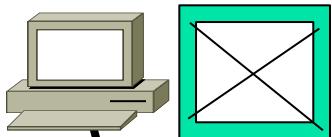
Thư điện tử (Email)

Email address



Thư điện tử (Email)

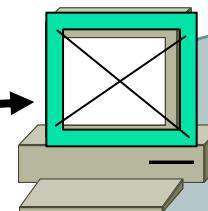
Người gửi



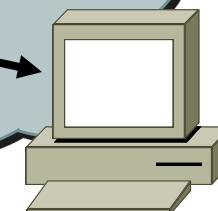
Hoạt động của giao thức truyền thư điện tử

Giao thức gửi
(**SMTP**)

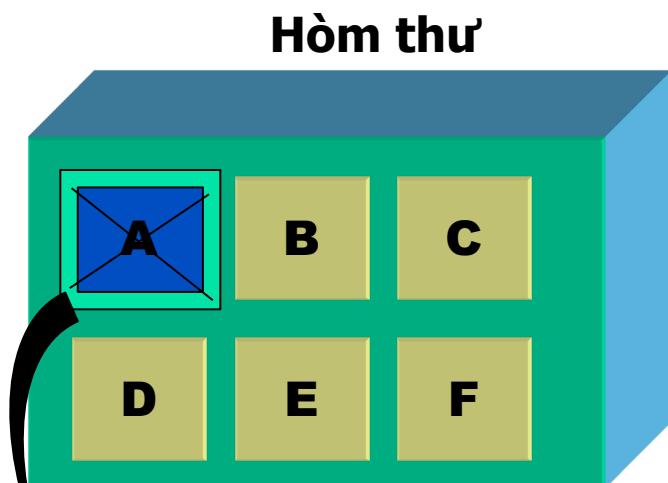
Máy chủ thư A



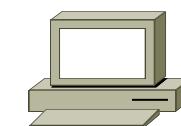
Giao thức gửi
(**SMTP**)



Máy chủ thư B



Giao thức nhận
(**POP3**)

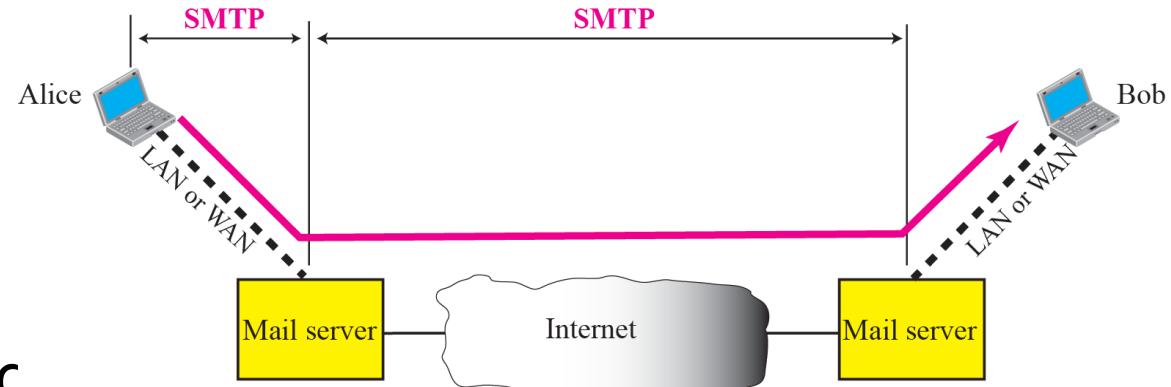


Người nhận

Thư điện tử (Email)

Giao thức SMTP

- Sử dụng TCP để thực hiện truyền thư tin cậy từ client tới server, dùng cổng 25.
- Truyền trực tiếp: từ server gửi tới server nhận.
- Ba pha truyền thư
 - Bắt tay (greeting)
 - Truyền thư
 - Đóng/ kết thúc
- Tương tác lệnh/đáp ứng
 - Lệnh (commands): văn bản ASCII
 - Đáp ứng (response): mã trạng thái và mệnh đề
- Bản tin dùng mã 7-bit ASCII
- Không có kiểm tra lỗi, không có mật mã hóa



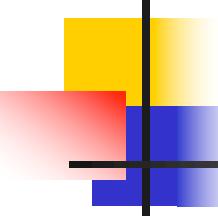
Thư điện tử (Email)

Giao thức SMTP



Table 23.1 *Commands*

| Keyword | Argument(s) | Keyword | Argument(s) |
|-----------|-----------------------|-----------|--------------------|
| HELO | Sender's host name | NOOP | |
| MAIL FROM | Sender of the message | TURN | |
| RCPT TO | Intended recipient | EXPN | Mailing list |
| DATA | Body of the mail | HELP | Command name |
| QUIT | | SEND FROM | Intended recipient |
| RSET | | SMOL FROM | Intended recipient |
| VRFY | Name of recipient | SMAL FROM | Intended recipient |

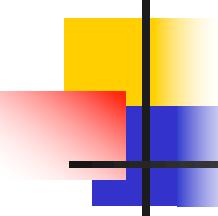


Thư điện tử (Email)

Giao thức SMTP



- **MAIL** : câu lệnh này để thiết lập một địa chỉ trả lại. Thông thường địa chỉ này là địa chỉ của người gửi và thường được sử dụng trong trường hợp server “bounce” mail-gửi lại mail cho người gửi nếu ko thể chuyen mail.
- **HELO**: câu lệnh này nói cho server biết rằng client muốn bắt đầu một phiên gửi mail.
- **RCPT** :câu lệnh này dùng để thiết lập một địa chỉ nhận của thông điệp. Câu lệnh này có thể được sử dụng nhiều lần khi muốn gửi thông điệp đến nhiều đích.
- **DATA** :câu lệnh này dùng để gửi nội dung của thông điệp.
- **QUIT** : câu lệnh dùng để kết thúc việc gửi mail.



Thư điện tử (Email)

Giao thức SMTP



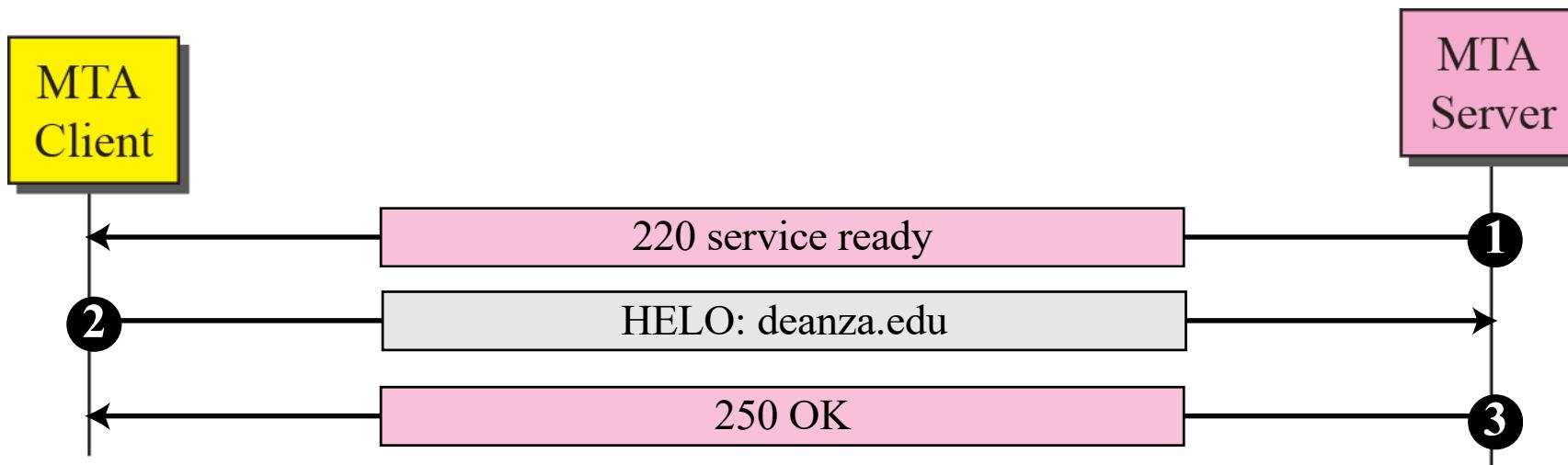
- 2xx: yêu cầu thành công. Client có thể tiếp tục
- 3xx:câu lệnh được chấp nhận,nhưng server muốn có thêm một vài thông tin. Client có thể sẽ phải gửi thêm một số thông tin
- 4xx:câu lệnh không thành công, nhưng vẫn được lưu tạm lại, client có thể thử lại sau
- 5xx: câu lệnh không thành công, client ko thể thử lại

Table 23.2 Responses

| Code | Description |
|--------------------------------------------|------------------------------------------------------|
| Positive Completion Reply | |
| 211 | System status or help reply |
| 214 | Help message |
| 220 | Service ready |
| 221 | Service closing transmission channel |
| 250 | Request command completed |
| 251 | User not local; the message will be forwarded |
| Positive Intermediate Reply | |
| 354 | Start mail input |
| Transient Negative Completion Reply | |
| 421 | Service not available |
| 450 | Mailbox not available |
| 451 | Command aborted: local error |
| 452 | Command aborted; insufficient storage |
| Permanent Negative Completion Reply | |
| 500 | Syntax error; unrecognized command |
| 501 | Syntax error in parameters or arguments |
| 502 | Command not implemented |
| 503 | Bad sequence of commands |
| 504 | Command temporarily not implemented |
| 550 | Command is not executed; mailbox unavailable |
| 551 | User not local |
| 552 | Requested action aborted; exceeded storage location |
| 553 | Requested action not taken; mailbox name not allowed |
| 554 | Transaction failed |

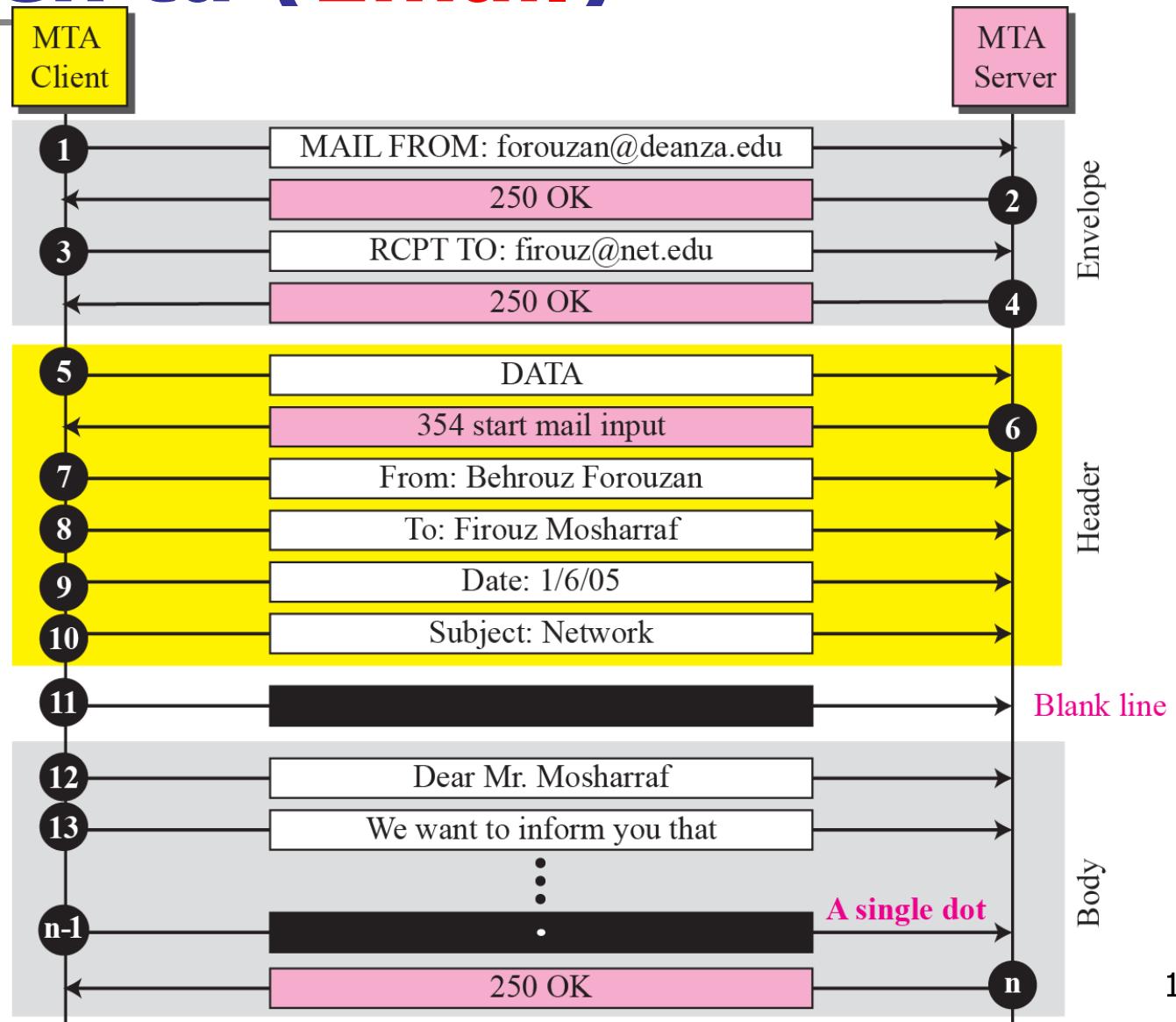
Thư điện tử (Email)

Giao thức SMTP: Thiết lập kết nối



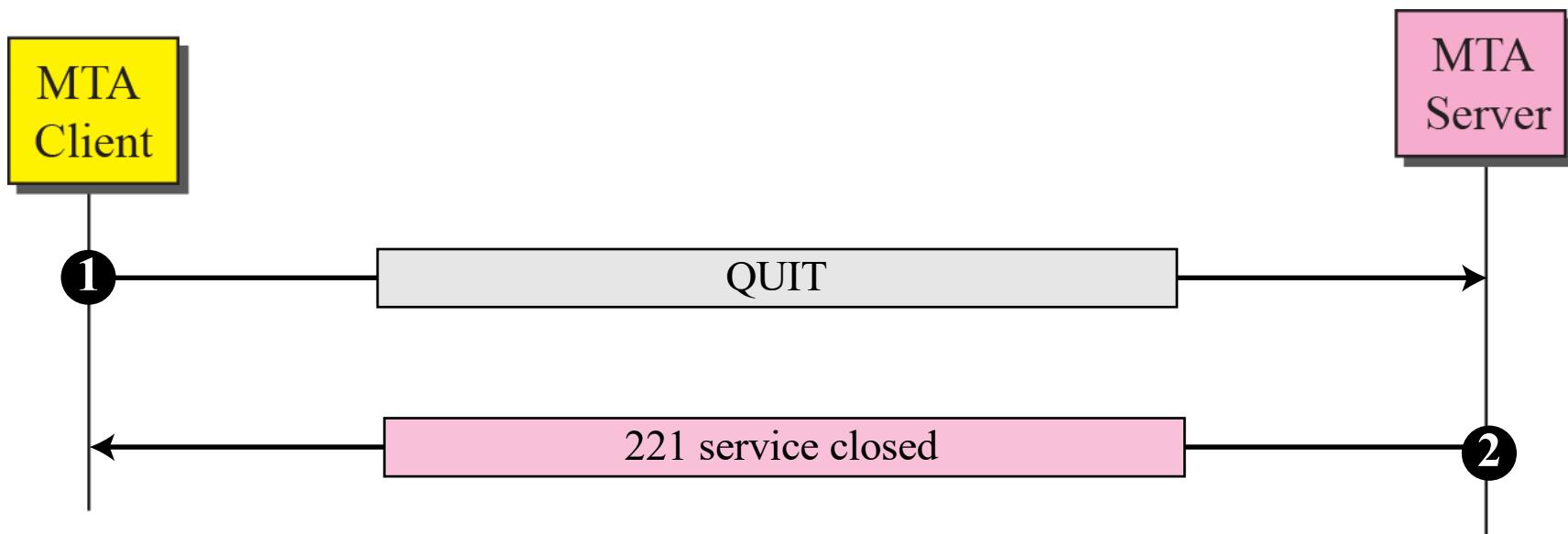
Thư điện tử (Email)

Giao thức SMTP: Truyền bản tin



Thư điện tử (Email)

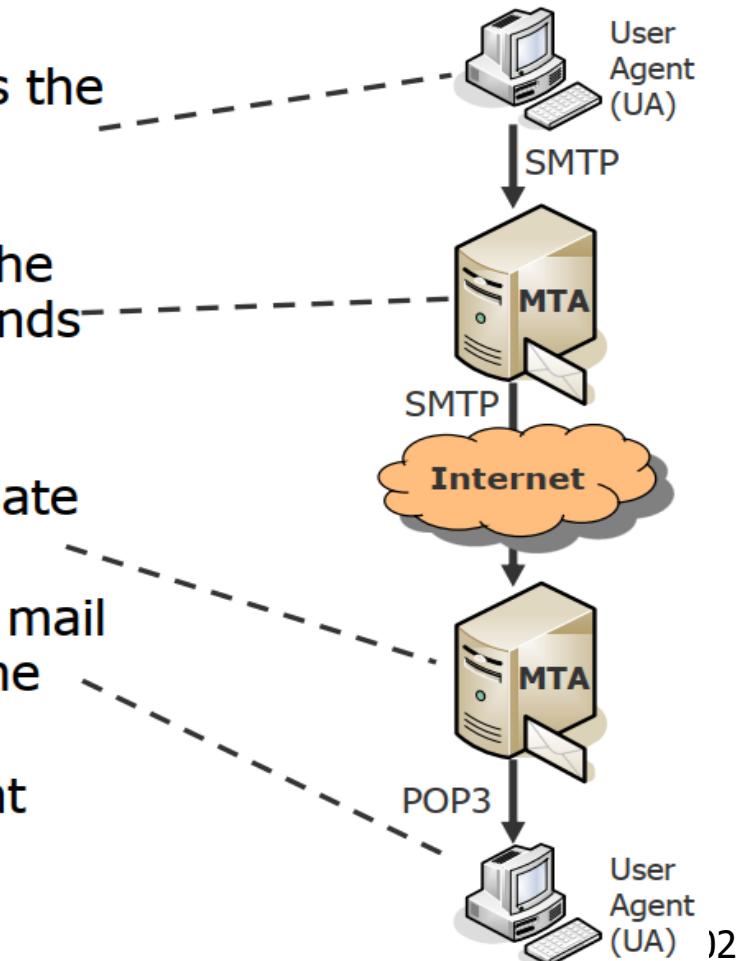
Giao thức SMTP: Giải phóng kết nối



Thư điện tử (Email)

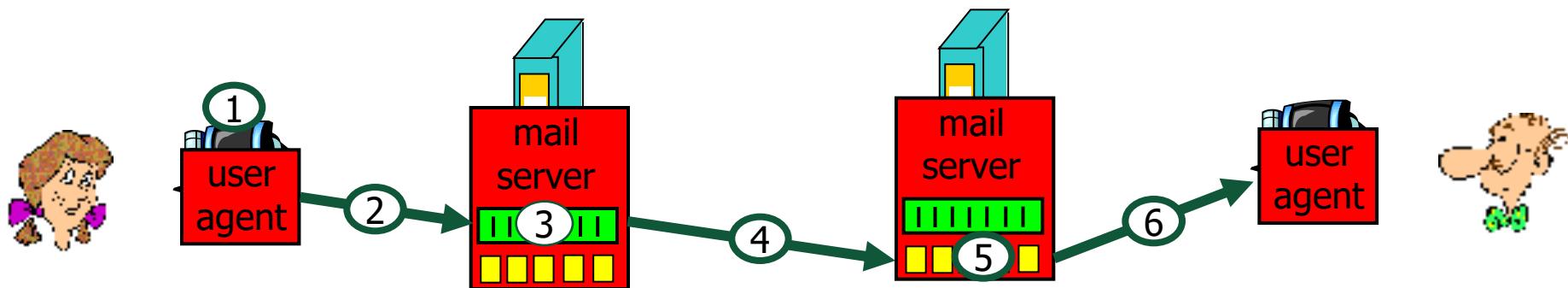
Email over POP3 and SMTP

- User 1: writes an email
- Client 1 (UA 1): formats the email, produces the receiver list, and sends the email to its mail server (MTA 1)
- Server 1 (MTA 1): Sets up a connection to the SMTP server (MTA 2) of the receiver and sends a copy of the email
- Server (MTA 2): Produces the header of the email and places the email into the appropriate mailbox
- Client 2 (UA 2): sets up a connection to the mail server and authenticates itself with username and password (unencrypted!)
- Server (MTA 2): sends the email to the client
- Client 2 (UA 2): formats the email
- User 2: reads the email



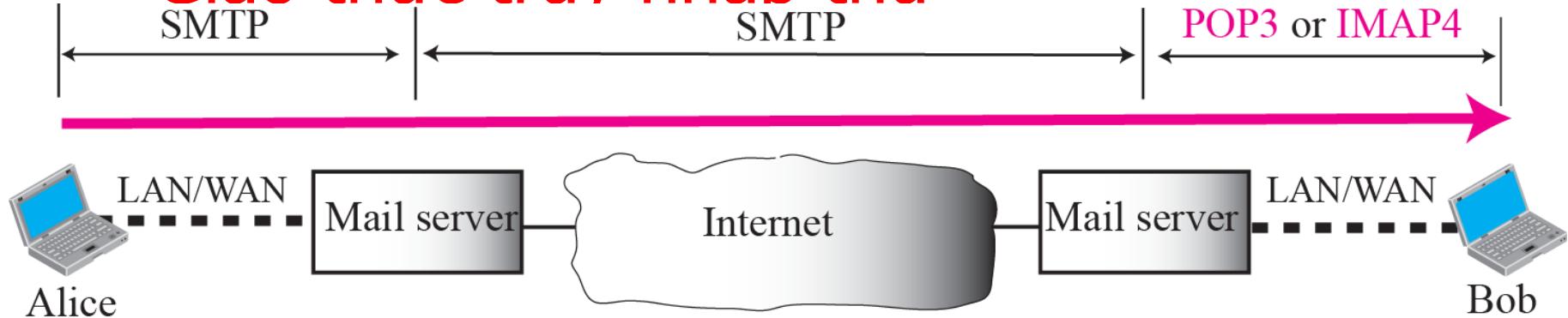
Thư điện tử (Email)

- 1) Lan dùng UA để viết thư gửi tới nam@ptit.edu.vn
- 2) UA của Lan gửi thư tới server thư của cô ấy, thư được lưu trong hàng đợi thư
- 3) Phía client của giao thức SMTP mở kết nối TCP với server thư của Nam
- 4) SMTP phía khách gửi thư của Lan trên kết nối TCP đó
- 5) Server thư của Nam chuyển thư vào hòm thư của Nam
- 6) Nam sử dụng UA của mình để đọc thư



Thư điện tử (Email)

Giao thức truy nhập thư

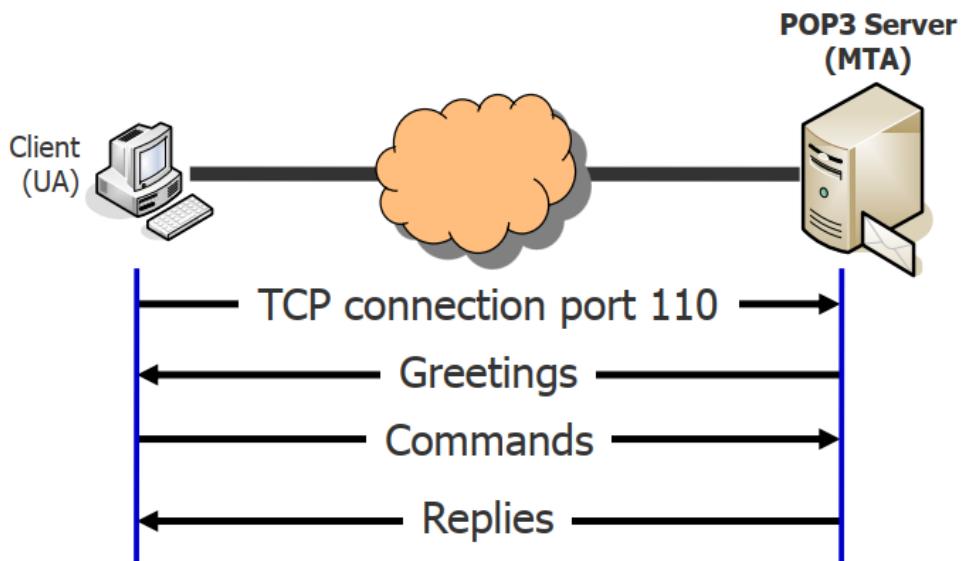


- SMTP: chuyển phát/lưu trữ tới server bên nhận
- Giao thức truy nhập thư: nhận từ server
 - **POP**: Post Office Protocol [RFC 1939]
 - Nhận thực (agent <-->server) và tải về (download)
 - **IMAP**: Giao thức truy nhập thư Internet [RFC 1730]
 - Nhiều đặc tính hơn (Phức tạp hơn)
 - Thao tác với các thư lưu trữ trên server
 - **HTTP**: gmail, Hotmail, Yahoo! Mail, etc.

Thư điện tử (Email)

POP3 protocol

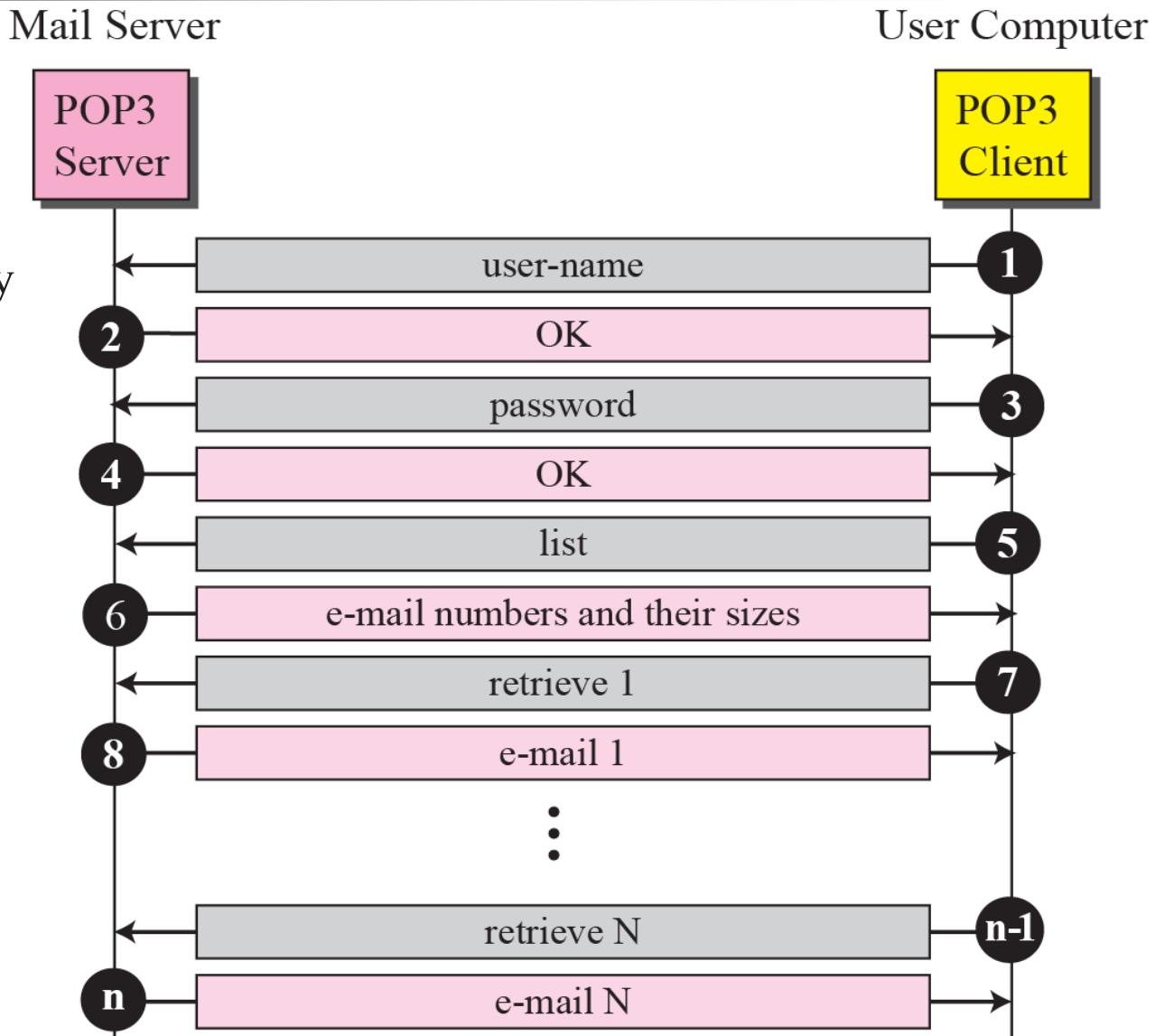
- POP3 là giao thức truy nhập thư rất đơn giản (RFC 1939)
 - chức năng của nó khá hạn chế
- POP3 bắt đầu khi người sử dụng (máy khách) mở một kết nối TCP tới máy chủ thư (máy chủ) trên cổng 110
- Sử dụng các lệnh để
 - Đăng nhập và đăng xuất
 - Tải thư
 - Xóa thư trên server

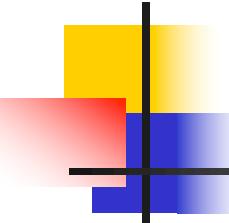


POP3 protocol

POP3 có 2 chế độ hoạt động:

- **Delete mode**: mail bị xóa khỏi mailbox sau khi tải về => phù hợp với user chỉ sử dụng 1 máy tính cố định để đọc thư.
- **Keep mode**: mail vẫn giữ ở mailbox sau khi tải về => phù hợp khi user truy cập mail ở nhiều thiết bị.





Thư điện tử (Email)

IMAP

- IMAP là giao thức truy nhập thư (RFC 3501)
- Một máy chủ IMAP sẽ liên kết mỗi bản tin với một thư mục, khi bản tin lần đầu đến máy chủ, nó được gắn với thư mục INBOX của người nhận.
- Người nhận có thể chuyển bản tin này vào một thư mục mới do người sử dụng tạo ra, đọc bản tin, xoá bản tin
- Giao thức IMAP cung cấp các lệnh để cho người sử dụng tạo thư mục và chuyển bản tin từ thư mục này tới thư mục khác.
- IMAP cung cấp các lệnh cho phép người sử dụng tìm kiếm thư trong thư mục từ xa theo những tiêu chí riêng
- Máy chủ IMAP duy trì thông tin trạng thái người sử dụng xuyên suốt các phiên IMAP (khác với POP3)

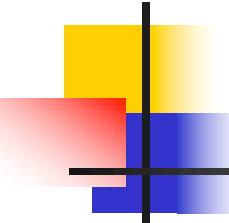
Thư điện tử (Email)

Thông tin thêm về POP3

- Chế độ (mode) “tải và xoá” (ví dụ trước): Nam không thể đọc lại email nếu anh ấy thay đổi máy khách.
- Chế độ “Tải và giữ”: nhiều bản sao trên các máy khách.
- POP3 không giữ trạng thái suốt phiên

IMAP

- Giữ toàn bộ thư tại địa điểm duy nhất là server
- Cho phép người dùng tổ chức thư vào các thư mục
- IMAP giữ trạng thái người dùng suốt phiên:
 - Tên các thư mục và ánh xạ giữa ID của bản tin và tên thư mục



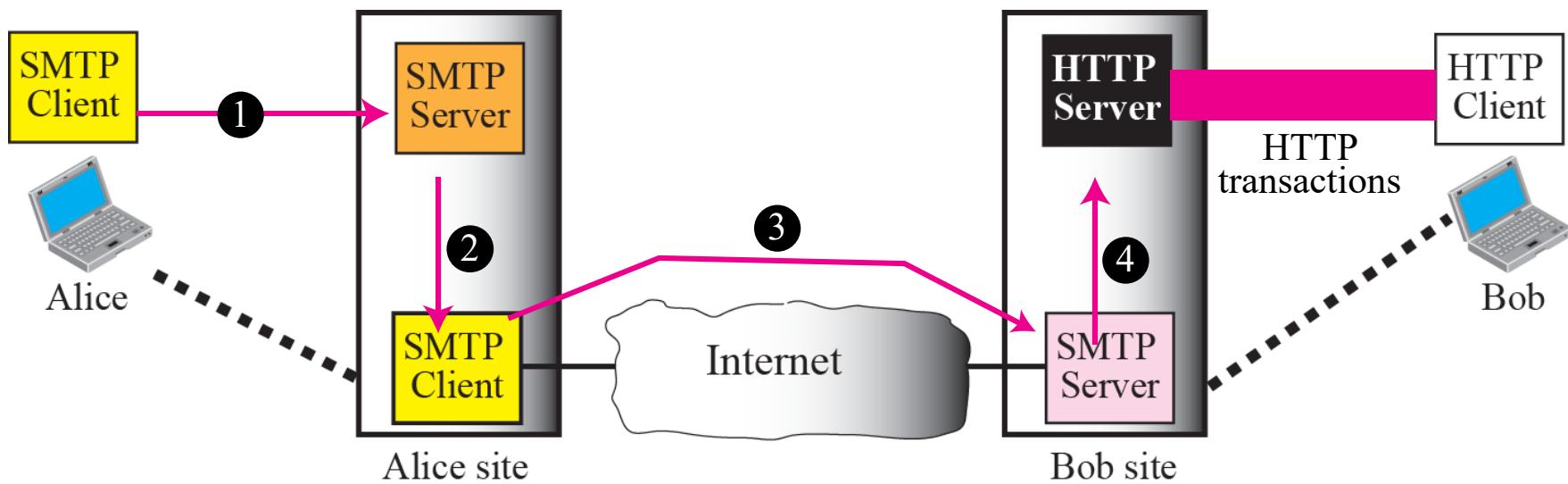
Thư điện tử (Email)

Web e-mail

- Ví dụ: Google, Hotmail, Yahoo.
- Đại lý người sử dụng là một trình duyệt Web.
- Người sử dụng truyền thông với hòm thư từ xa của họ thông qua HTTP.
 - Khi người nhận (Bob) muốn truy nhập thư trong hòm thư của mình, thì thư điện tử được gửi từ máy chủ thư của Bob tới trình duyệt của Bob sử dụng giao thức HTTP chứ không phải là POP3 hay IMAP
 - Khi người gửi (Alice) muốn gửi thư điện tử, thư điện tử của cô ấy sẽ được gửi từ trình duyệt của cô ta tới máy chủ thư của cô ấy trên HTTP chứ không phải SMTP.
 - Tuy nhiên, máy chủ thư của Alice vẫn gửi bản tin tới và nhận bản tin từ những máy chủ thư khác qua SMTP.

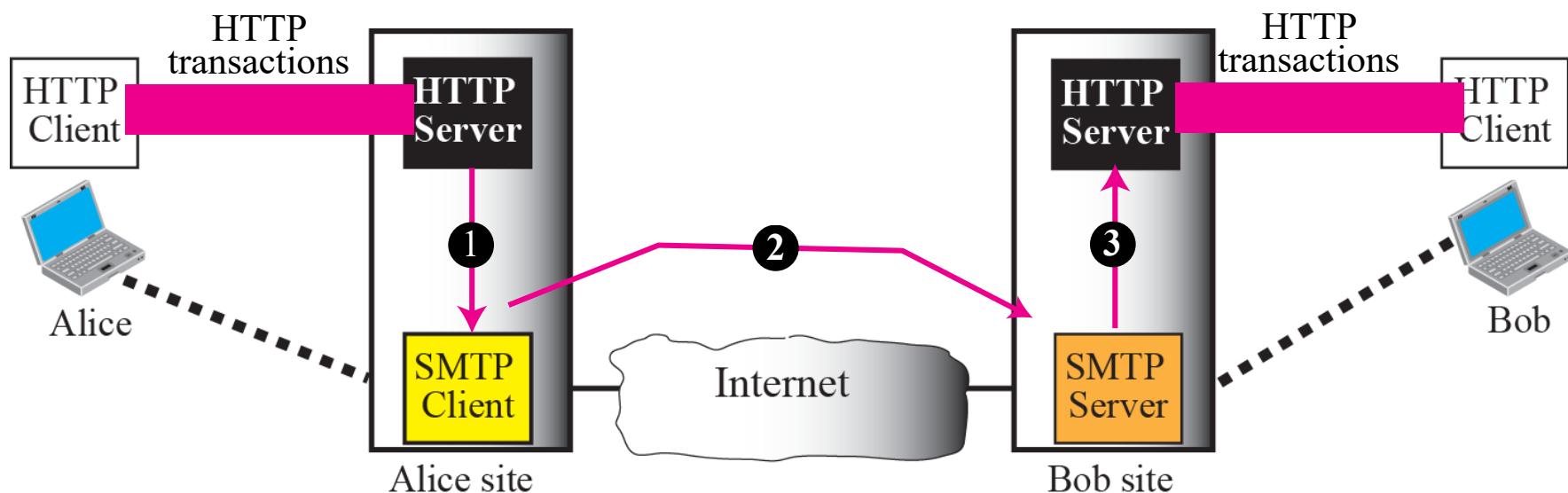
Thư điện tử (Email)

Web e-mail: kịch bản 1



Thư điện tử (Email)

Web e-mail: kịch bản 2



Thư điện tử (Email)

So sánh SMTP với HTTP

- SMTP sử dụng kết nối ổn định (persistent)
- SMTP yêu cầu bản tin (thư) (header & body) là mã 7-bit ASCII
- SMTP server dùng CRLF.CRLF để xác định kết thúc bản tin (thư)

So sánh

- HTTP: chủ yếu là kéo (pull)
- SMTP: đẩy (push)
- Cả hai đều dùng lệnh/phản hồi ASCII, các mã trạng thái, mệnh đề
- HTTP: mỗi đối tượng đóng trong bản tin phản hồi của nó
- SMTP: **nhiều đối tượng** gửi trong một bản tin gồm nhiều phần

Chương 2: Các ứng dụng và giao thức

- 2.1 Tổng quan về các ứng dụng và dịch vụ hạ tầng
- 2.2 Ứng dụng WEB và các giao thức
- 2.3 Ứng dụng truyền tệp và các giao thức
- 2.4 Ứng dụng thư điện tử và các giao thức
- 2.5 Ứng dụng hệ thống tên miền**
- 2.6 Các ứng dụng mạng ngang hàng

DNS - Hệ thống tên miền

- DNS (Domain Name System) – Hệ thống tên miền là ứng dụng client-server, nhận dạng duy nhất mỗi host ứng với một tên duy nhất.

Application layer

Ping

SMTP

FTP

Telnet

NNTP

etc...

NFS
RPC

DNS

TFTP

BOOTP

etc...

Transport layer

OSPF

ICMP

IGMP

BGP

TCP

UDP

RIP

Internet layer

IP

Network Access layer

ARP

RARP

Data link

Media
(physical)

Q1: Cho biết ý nghĩa của địa chỉ IP và cấu trúc địa chỉ.
 Q2: Em hiểu thế nào là host address và net address trong địa chỉ IPv4?

- Dùng cổng 53
- Có thể sử dụng UDP (chủ yếu) hoặc TCP (hiếm khi, chỉ khi bản tin phản hồi có kích thước lớn hơn 512 byte)
- DNS được định nghĩa trong các RFC 1034 và 1035

Tổng quan về DNS

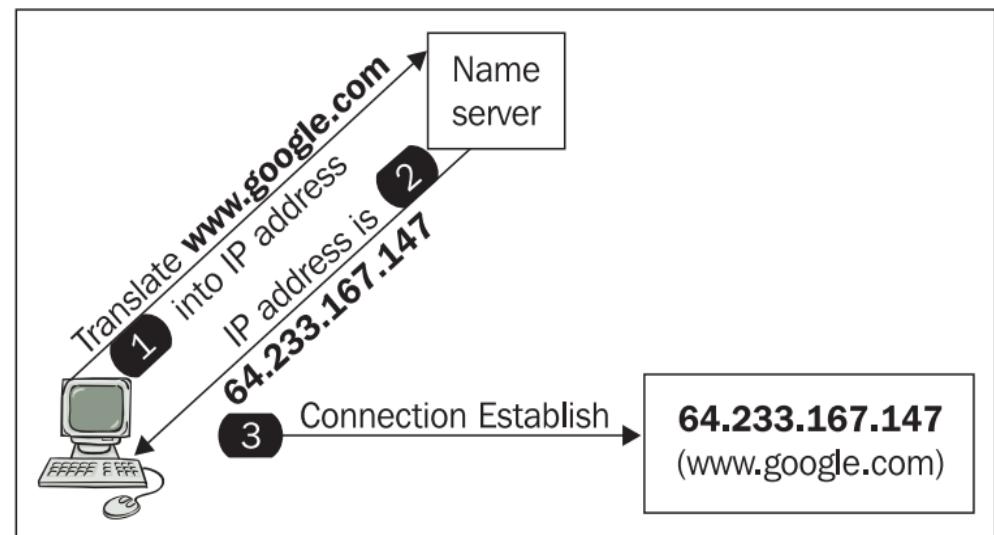
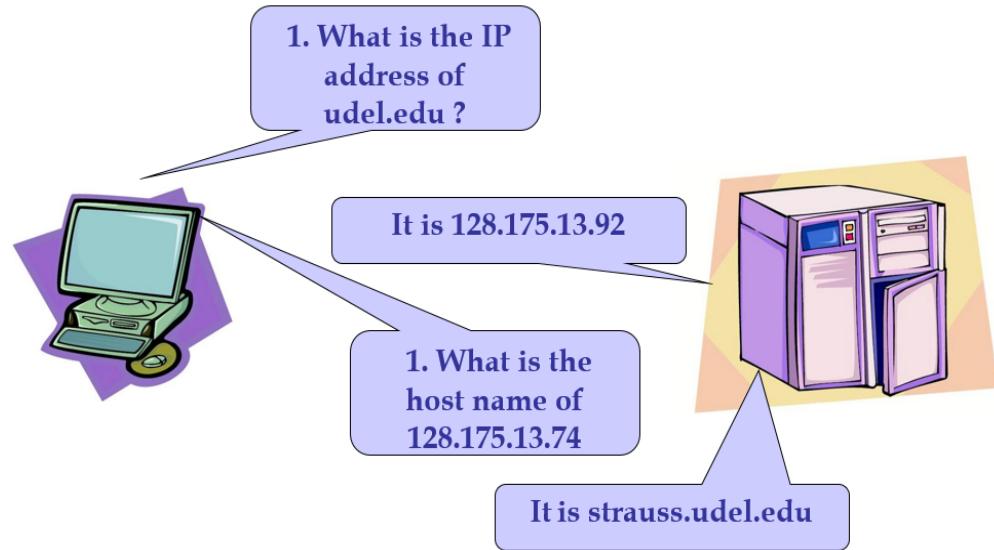
Con người: có nhiều kiểu nhận dạng như

- tên (name)
- số CMT, hộ chiếu (passport) ...

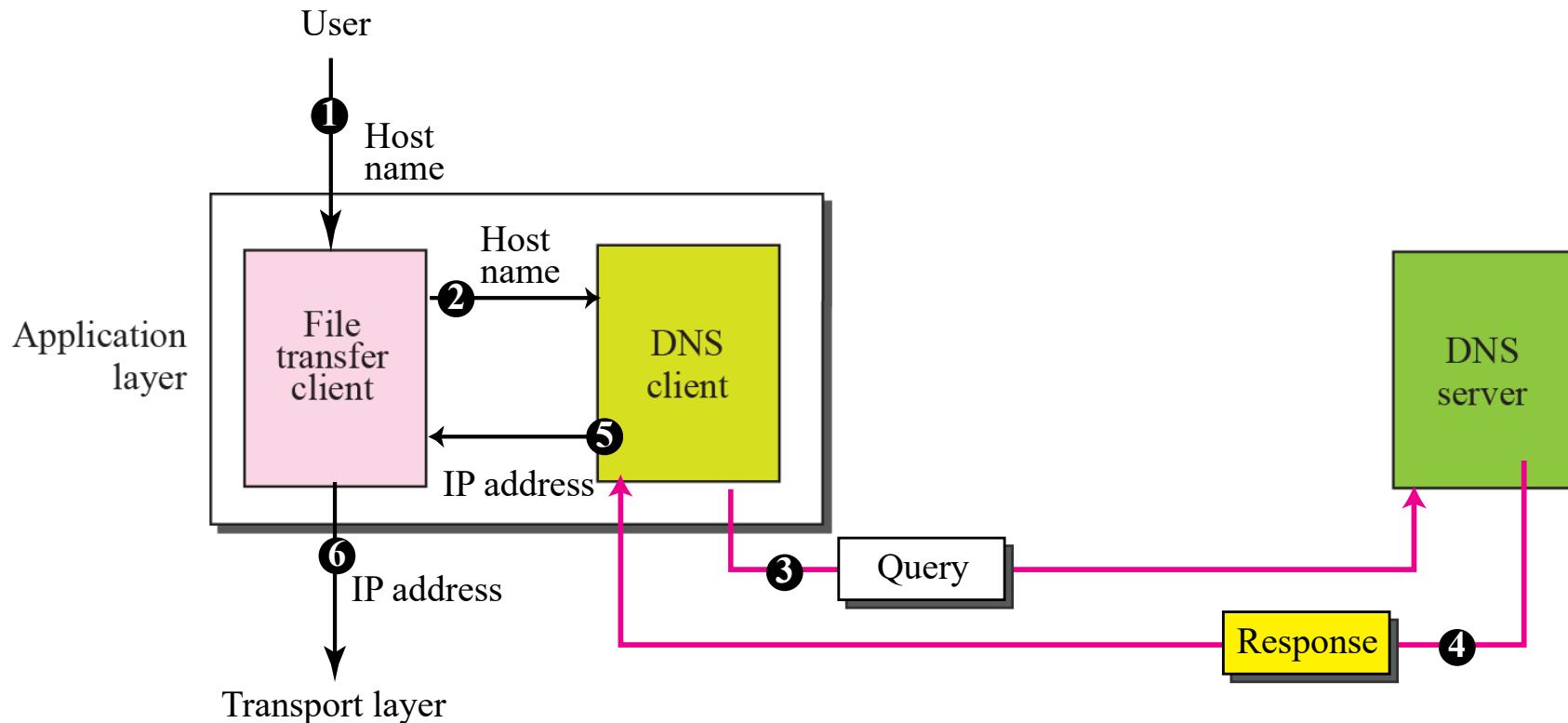
Trạm, router trên Internet :

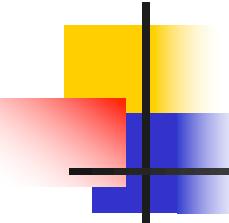
- Địa chỉ IPv4 (32 bit) sử dụng cho gói dữ liệu
- Tên host, (mail.yahoo.com) ... sử dụng cho con người

Q: ánh xạ giữa địa chỉ IP và tên như thế nào



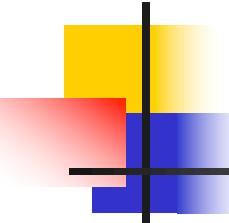
Tổng quan về DNS





Tổng quan về DNS

- ❑ Vào những năm 1970 mạng ARPAnet của bộ quốc phòng Mỹ rất nhỏ và dễ dàng quản lý các liên kết vài trăm máy tính với nhau.
- ❑ Do đó mạng chỉ cần một file HOSTS.TXT chứa tất cả thông tin cần thiết về máy tính trong mạng và giúp các máy tính chuyển đổi được thông tin địa chỉ và tên mạng cho tất cả máy tính trong mạng ARPAnet một cách dễ dàng.
 - */etc/hosts* on Linux
 - *C:\Windows\System32\drivers\etc\hosts* on Windows
- ❑ Đây là hệ thống nhân công và tập trung
 - *Các thay đổi được cập nhật thường xuyên một cách nhân công*
 - *Các nhà quản trị có thể đặt tên một cách tùy tiện, bất kỳ tên nào cũng hợp lệ*



Tổng quan về DNS

- ❑ Khi mạng máy tính ARPAnet ngày càng phát triển thì việc quản lý thông tin chỉ dựa vào một file HOSTS.TXT là rất khó khăn và không khả thi.
- ❑ Vì thông tin bổ sung và sửa đổi vào file HOSTS.TXT ngày càng nhiều và nhất là khi ARPAnet phát triển hệ thống máy tính dựa trên giao thức TCP/IP dẫn đến sự phát triển tăng vọt của mạng máy tính.
- ❑ Nhiều máy có thể có file host.txt không chính xác
- ❑ Do đó, DNS ra đời

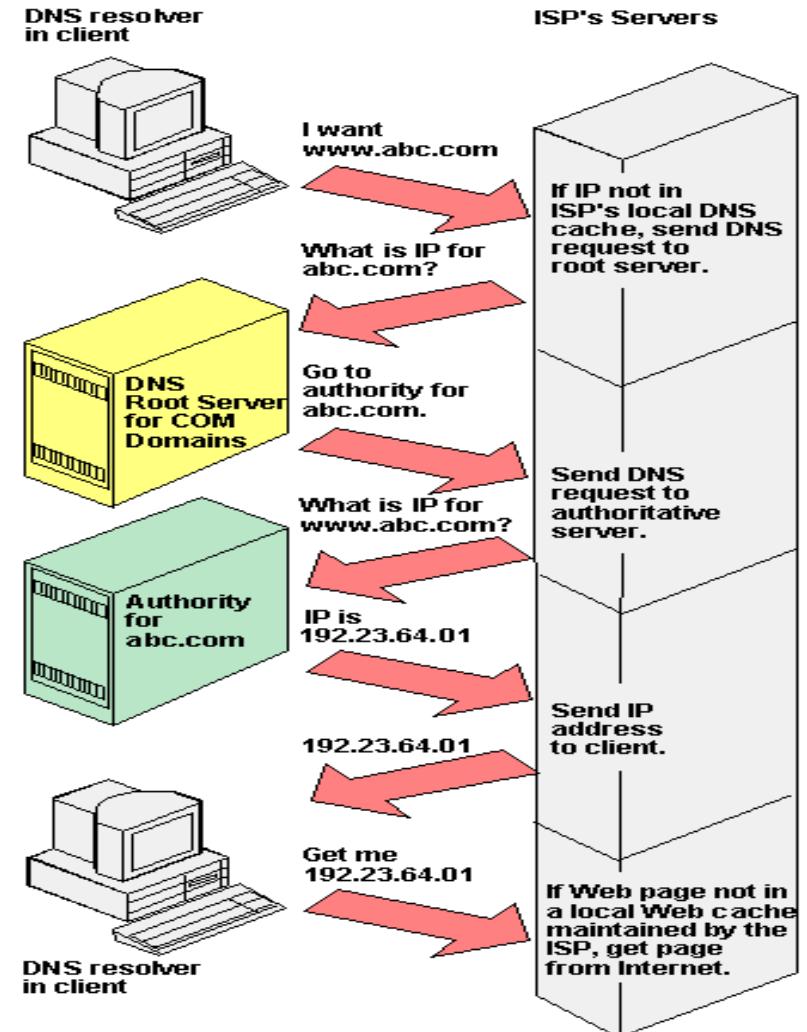
Tổng quan về DNS

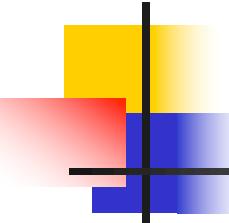
Hệ thống tên miền (DNS):

- Cơ sở dữ liệu phân tán* được thực hiện trong cấu trúc phân cấp máy chủ tên miền (*name servers*)
- Giao thức lớp ứng dụng* cho phép các trạm chủ truy vấn cơ sở dữ liệu phân tán
- Kiến trúc client/server sử dụng **UDP** cổng **53**, một số thực thi sử dụng **TCP**
- Được các giao thức ứng dụng khác sử dụng - bao gồm **HTTP**, **SMTP** và **FTP** - để phiên dịch tên trạm chủ sang địa chỉ IP

Q: Tại sao không tập trung DNS

From Computer Desktop Encyclopedia
© 2010 The Computer Language Co., Inc.





Tổng quan về DNS

Dịch vụ DNS

- Phiên dịch tên trạm sang địa chỉ IP
- Bí danh trạm chủ: Một trạm chủ có tên trạm phức tạp có thể có một hay nhiều bí danh.
- Bí danh server thư: để người dùng dễ nhớ hơn
- Phân tải: sử dụng để thực hiện phân bõ tải giữa các máy chủ nhân rộng
 - Các server Web được nhân rộng ra: tập các địa chỉ IP cho một tên chính tắc được lưu ở DNS

Tại sao không tập trung DNS?

- Khi có lỗi ở 1 điểm thì cả hệ thống lỗi
- Lưu lượng tập trung vào một DNS server sẽ quá lớn
- Khoảng cách tới cơ sở dữ liệu ở xa
- Duy trì và cập nhật: khó khăn

Không đáp ứng quy mô mạng lớn

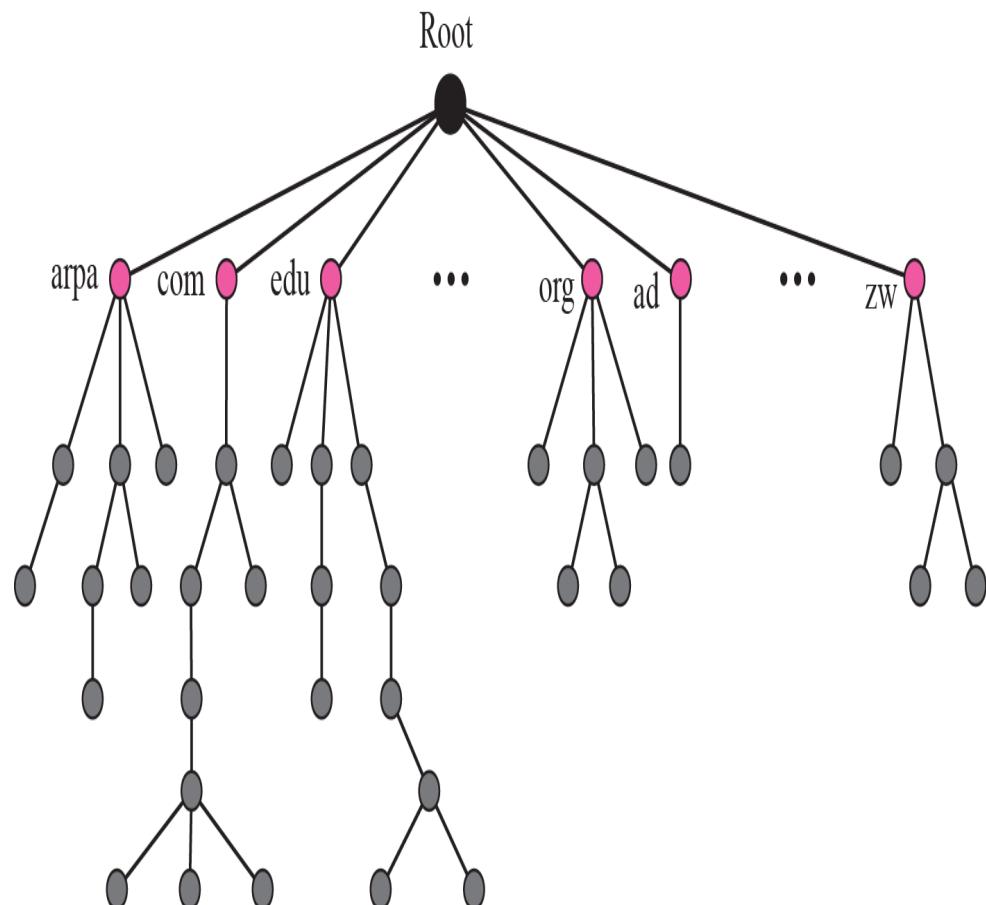
Các thành phần của DNS

- **The name space:** là cấu trúc cơ sở dữ liệu của DNS
- **The servers:** Lưu trữ thông tin về name space theo đơn vị zone
- **The name resolvers:** thực hiện phân giải tên miền

Các thành phần của DNS

The name space:

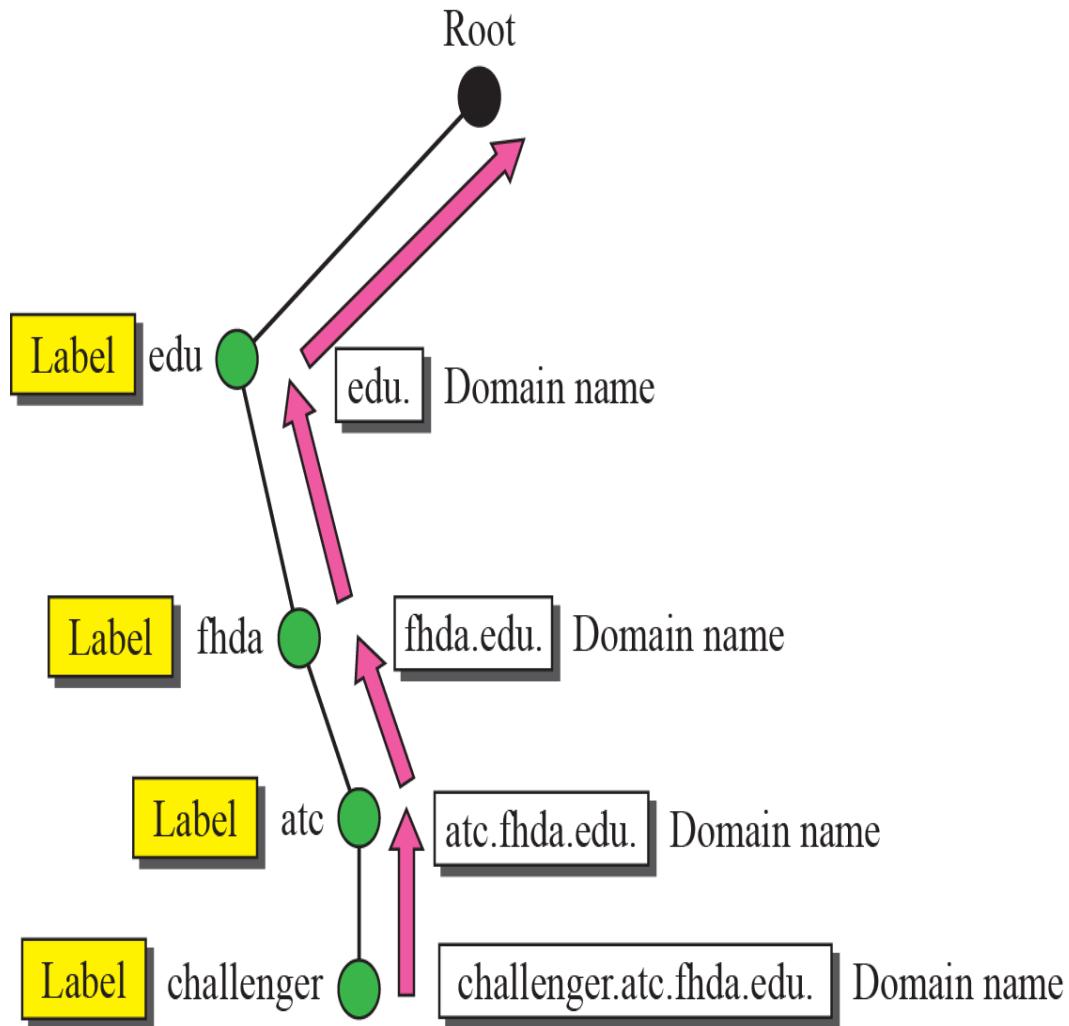
- ❖ Có cấu trúc hình cây với nút Root là gốc, tối đa 127 mức
- ❖ Mỗi nút có một nhãn, trừ nút Root



Các thành phần của DNS

The name space:

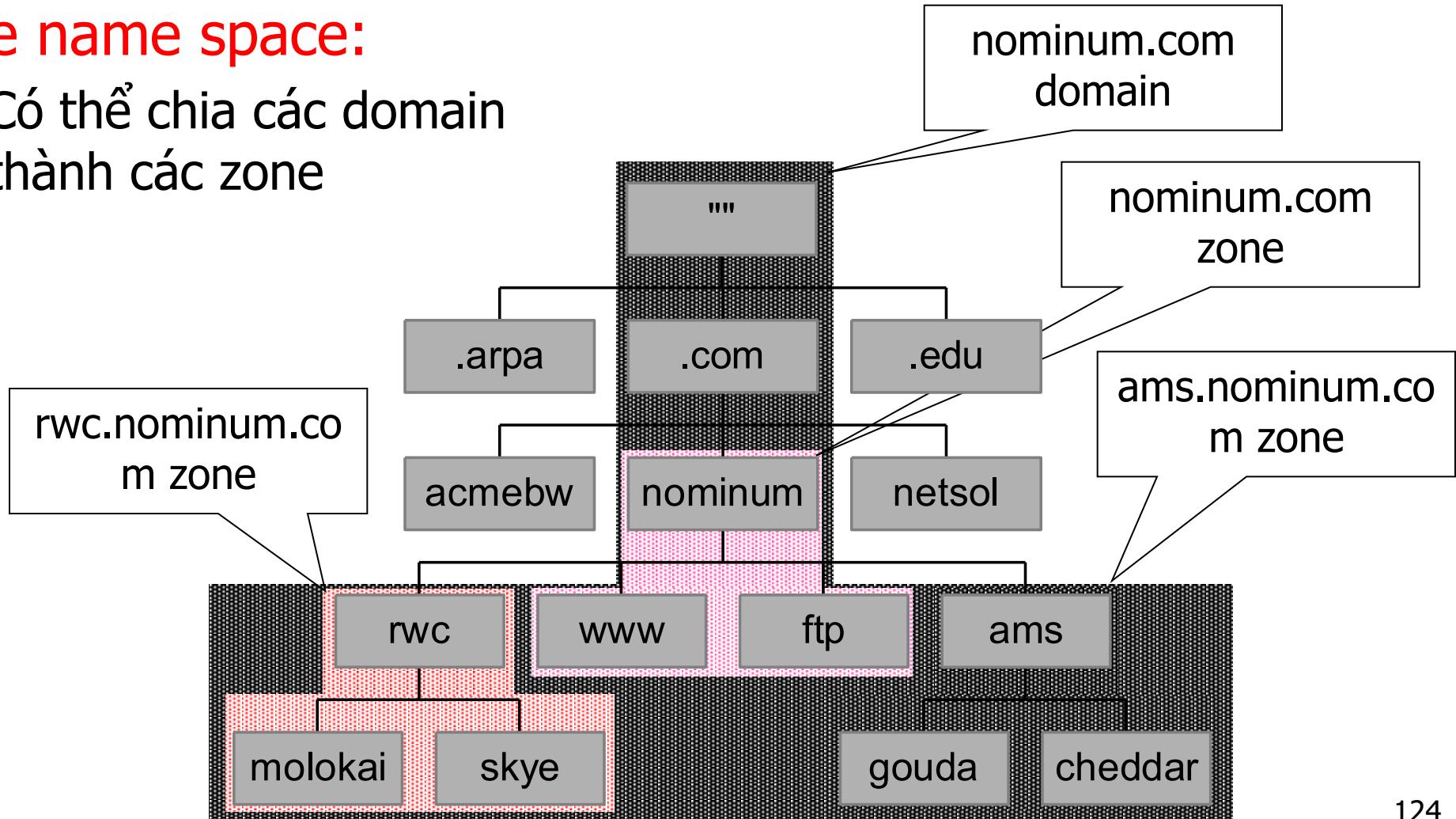
- Domain name là chuỗi các nhãn từ một nút đến root, được tách biệt bởi dấu . và có độ dài tối đa 255 kí tự
- Tên miền của một nút xác định vị trí của nút đó trong name space
- Có thể chia các domain thành các zone



Các thành phần của DNS

The name space:

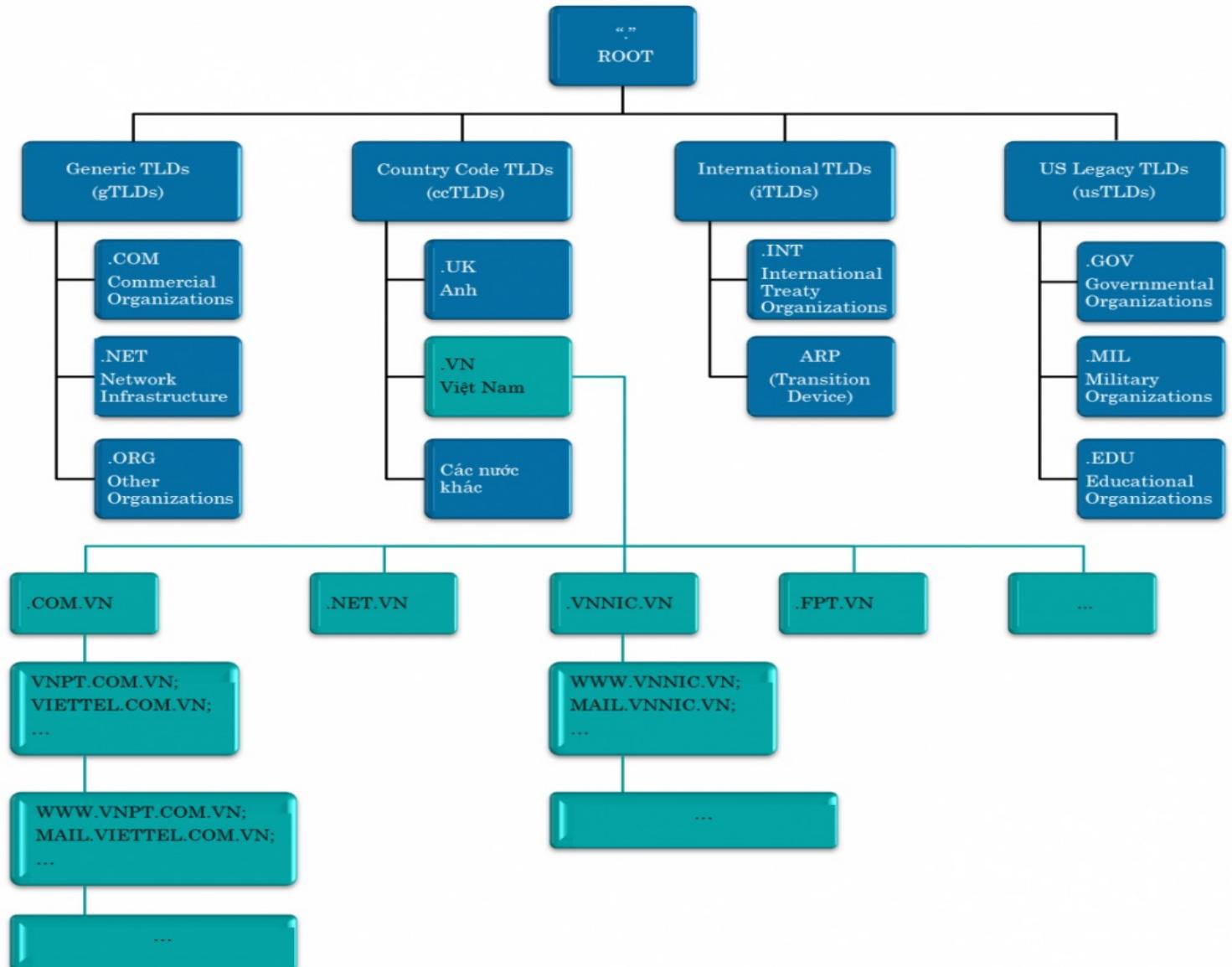
- ❖ Có thể chia các domain thành các zone



Các thành phần của DNS

Cấu trúc hệ thống tên miền:

- Top Level Domain Names (TLDs)
- Subdomains



Các thành phần của DNS

Root domain được biểu diễn bằng 1 dấu chấm (.), mặc định không ghi. Nên ta chỉ thấy tên của top-level domain.

| Tên miền | Mô tả |
|----------|-----------------------------------------------------|
| .com | Các tổ chức, công ty thương mại |
| .org | Các tổ chức phi lợi nhuận |
| .net | Các trung tâm hỗ trợ về mạng |
| .edu | Các tổ chức giáo dục |
| .gov | Các tổ chức thuộc chính phủ |
| .mil | Các tổ chức quân sự |
| .int | Các tổ chức được thành lập bởi các hiệp ước quốc tế |

Các thành phần của DNS

Các domain tồn tại đã quá tải nên phát sinh ra các top-level domain mới, và các tên miền quốc gia ví dụ :

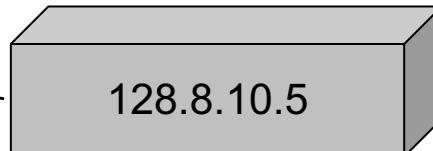
| Tên miền | Mô tả |
|-------------------|---------------------------------------------------|
| .arts | Các tổ chức liên quan đến nghệ thuật và kiến trúc |
| .rec | Các tổ chức có tính chất giải trí, thể thao |
| .firm | Các tổ chức kinh doanh thương mại |
| .info | Các tổ chức liên quan đến thông tin |
| Tên miền Quốc gia | Tên quốc gia |
| .vn | Việt Nam |
| .us | Mỹ |
| .uk | Anh |
| .jp | Nhật |
| .cn | Trung quốc |

Các thành phần của DNS

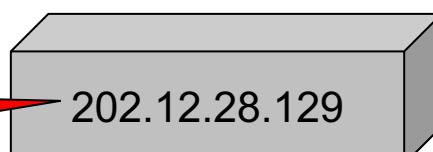
- **The name server:** Lưu trữ thông tin về name space
 - ❖ Thường có nhiều hơn một name server chịu trách nhiệm cho cùng một zone: đảm bảo tính dư thừa và trải đều tải
 - ❖ Một name server cũng có thể chịu trách nhiệm cho nhiều zone

128.8.10.5 serves data for both nominum.com and isc.org zones

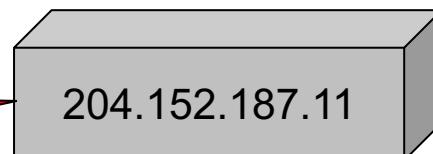
Name Servers



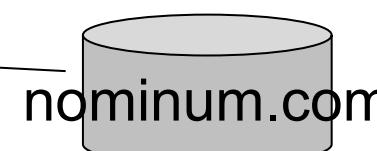
202.12.28.129 serves data for nominum.com zone only



204.152.187.11 serves data for isc.org zone only

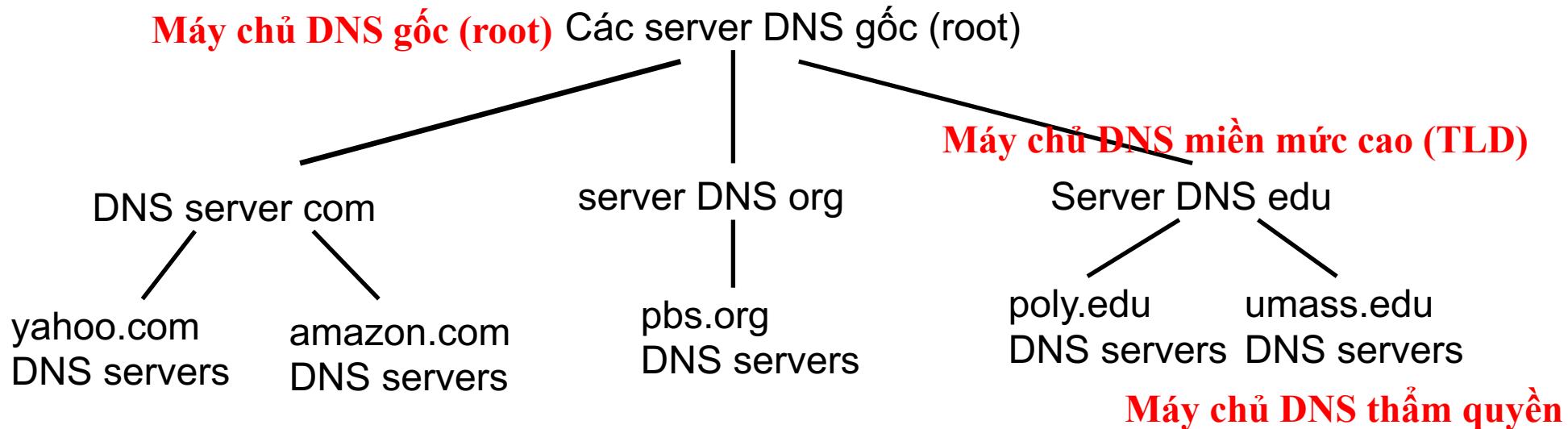


Zones



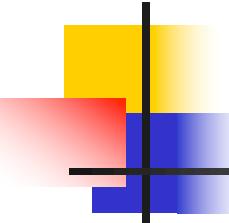
Các thành phần của DNS

Cơ sở dữ liệu phân cấp, phân tán: Phân cấp server



Client muốn tìm địa chỉ IP cho trang www.amazon.com

- client yêu cầu root server tìm server DNS com
- client yêu cầu server DNS com tìm server DNS amazon.com
- client yêu cầu server DNS amazon.com lấy địa chỉ IP của www.amazon.com



Các thành phần của DNS

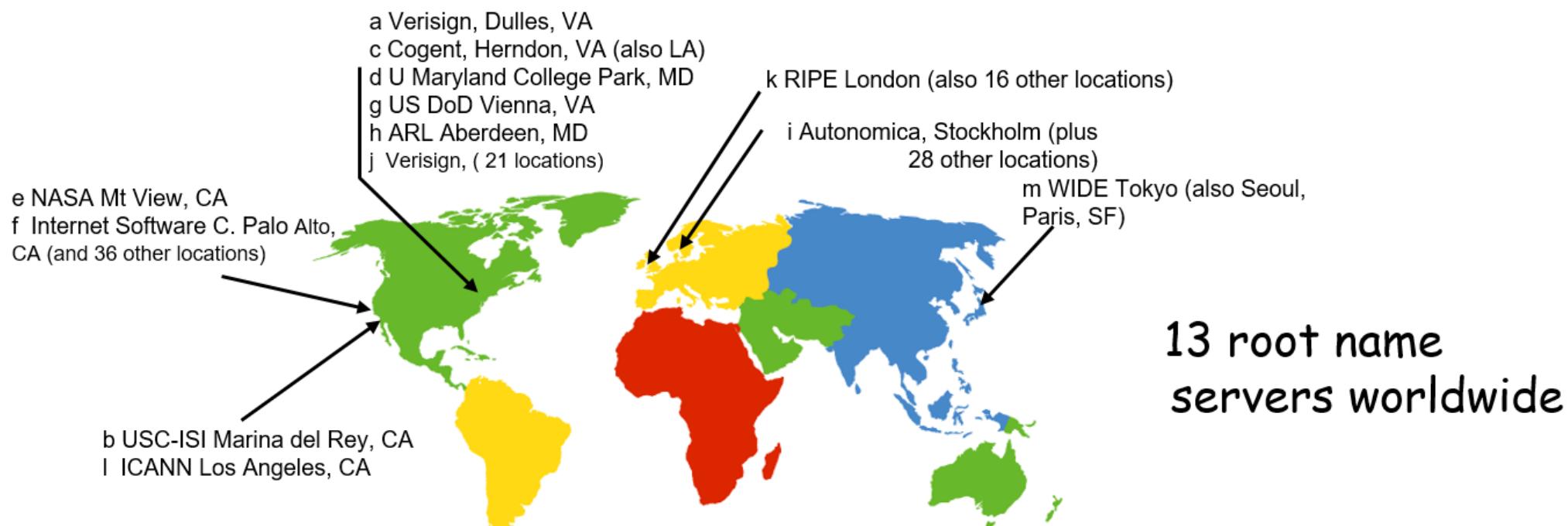
The name server:

- Chức năng của mỗi DNS server:
 - Có quyền hạn đối với một phần của hệ thống phân cấp, không cần lưu trữ toàn bộ tên DNS
 - Lưu trữ tất cả các bản ghi cho các hosts/domains trong vùng của nó
 - Biết địa chỉ của các root server
- Root server biết tất cả các server mức cao TLD

Các thành phần của DNS

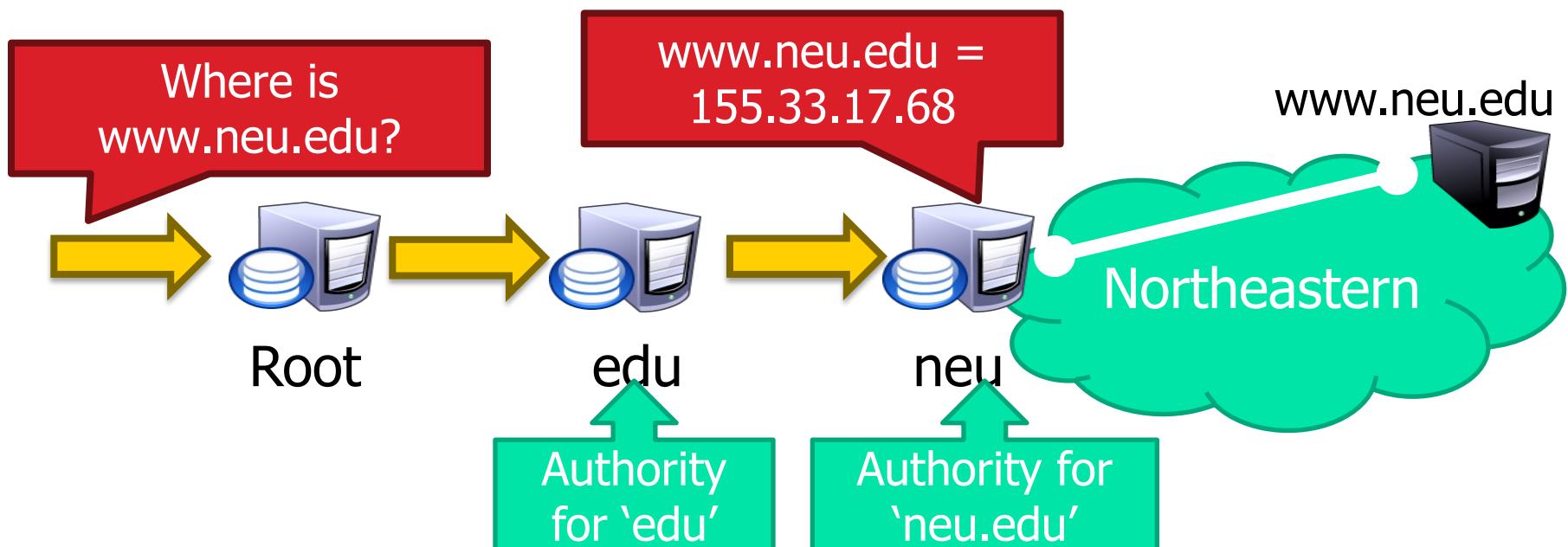
Root server:

- Được quản lý bởi ICANN, có 13 máy chủ DNS gốc (dán nhãn từ A đến M), hầu hết đặt ở Bắc Mỹ, mỗi máy chủ thực chất là một nhóm máy chủ nhánh rộng với mục đích là tăng độ an toàn và tin cậy
- Bản đồ máy chủ DNS gốc tháng 10 năm 2009:



Các thành phần của DNS

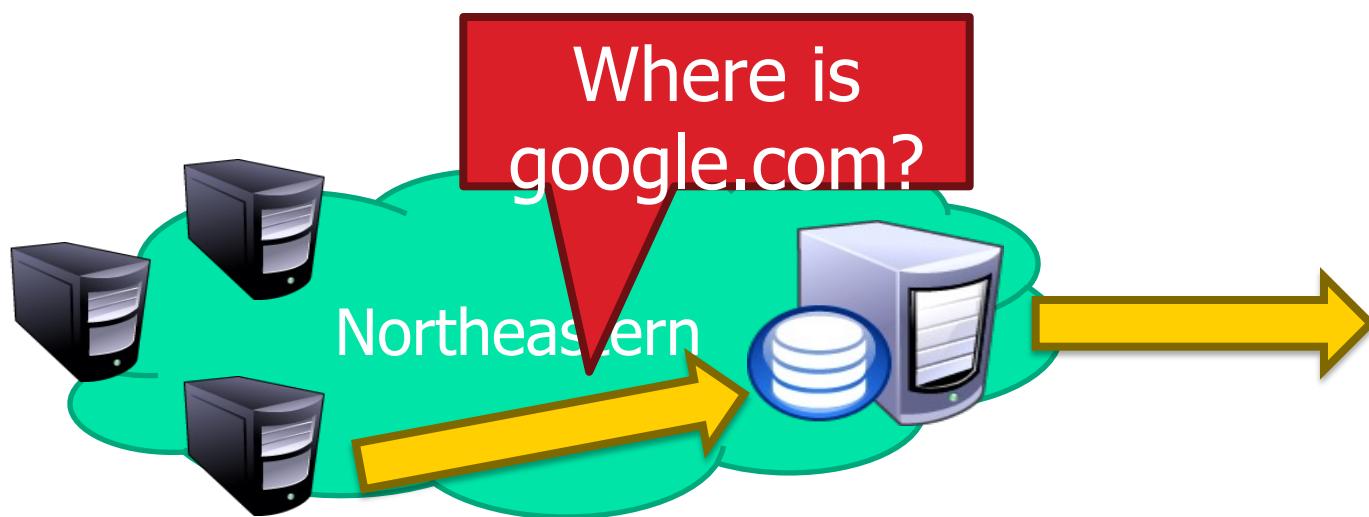
- **Server tên miền mức cao (TLD) :**
là server cho com, org, net, edu, ... và tất cả tên miền cấp quốc gia uk, fr, ca, us, jp, cn, vn ...
- **Server DNS thẩm quyền (Authoritative Name Server):**
Server DNS của các tổ chức, cung cấp ánh xạ tên trạm được cấp quyền với địa chỉ của server các tổ chức (ví dụ: trang Web, mail).
Do nhà cung cấp dịch vụ hoặc tổ chức duy trì.



Các thành phần của DNS

■ Server tên miền cục bộ

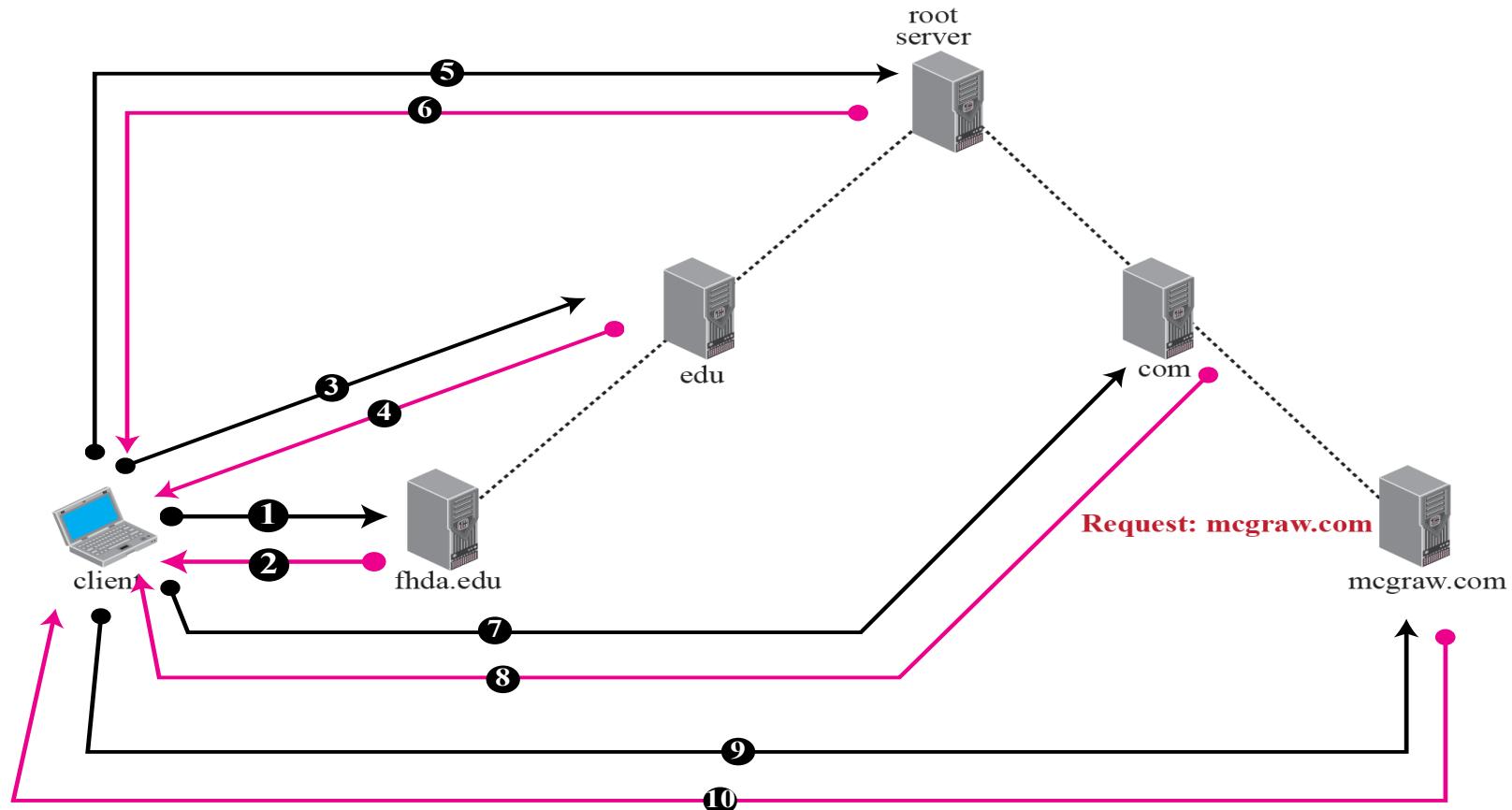
- ❖ Không bắt buộc phải có phân cấp
- ❖ Mỗi ISP/công ty (ISP dân cư, công ty, trường đại học) có một server tên miền cục bộ, được gọi là server tên miền mặc định
- ❖ Khi host yêu cầu DNS, yêu cầu được gửi tới server DNS cục bộ đầu tiên
- ❖ Thường được cấu hình qua DHCP
- ❖ Thường xuyên lưu đệm các kết quả truy vấn
- ❖ Hoạt động như proxy, chuyển tiếp yêu cầu vào phân cấp



Hoạt động của DNS

Truy vấn lặp lại (Iterative):

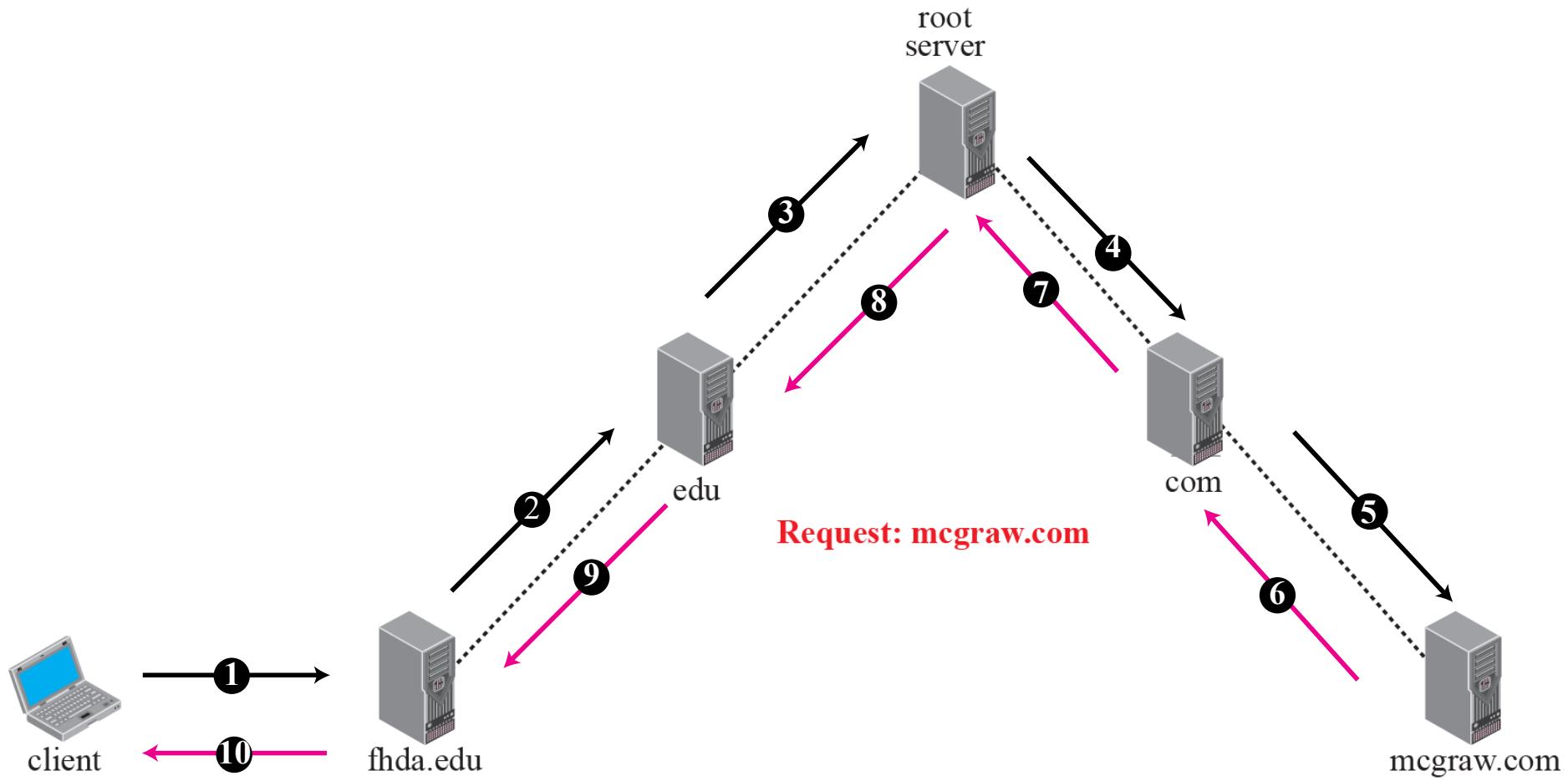
- Server được kết nối (liên hệ) trả lại tên miền của server cần để kết nối
- “Tôi không biết tên miền này, nhưng hãy hỏi server này”
- Đây là cách hoạt động phổ biến



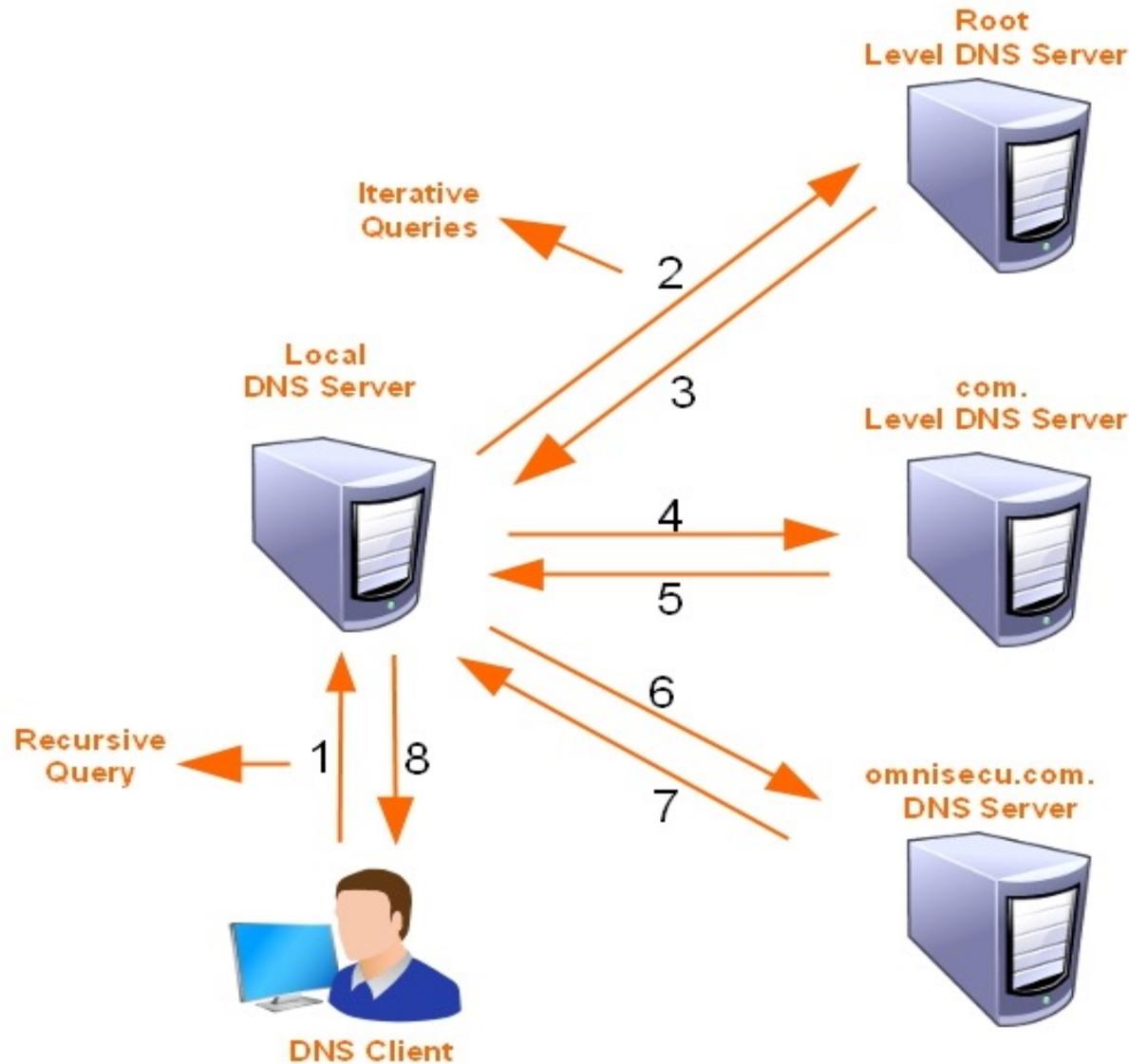
Hoạt động của DNS

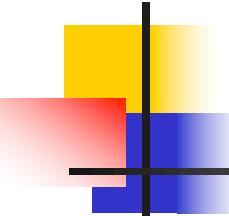
- Truy vấn đệ quy (Recursive) trong DNS

Đưa trọng trách xử lý tên miền cho server được kết nối.



Hoạt động của DNS

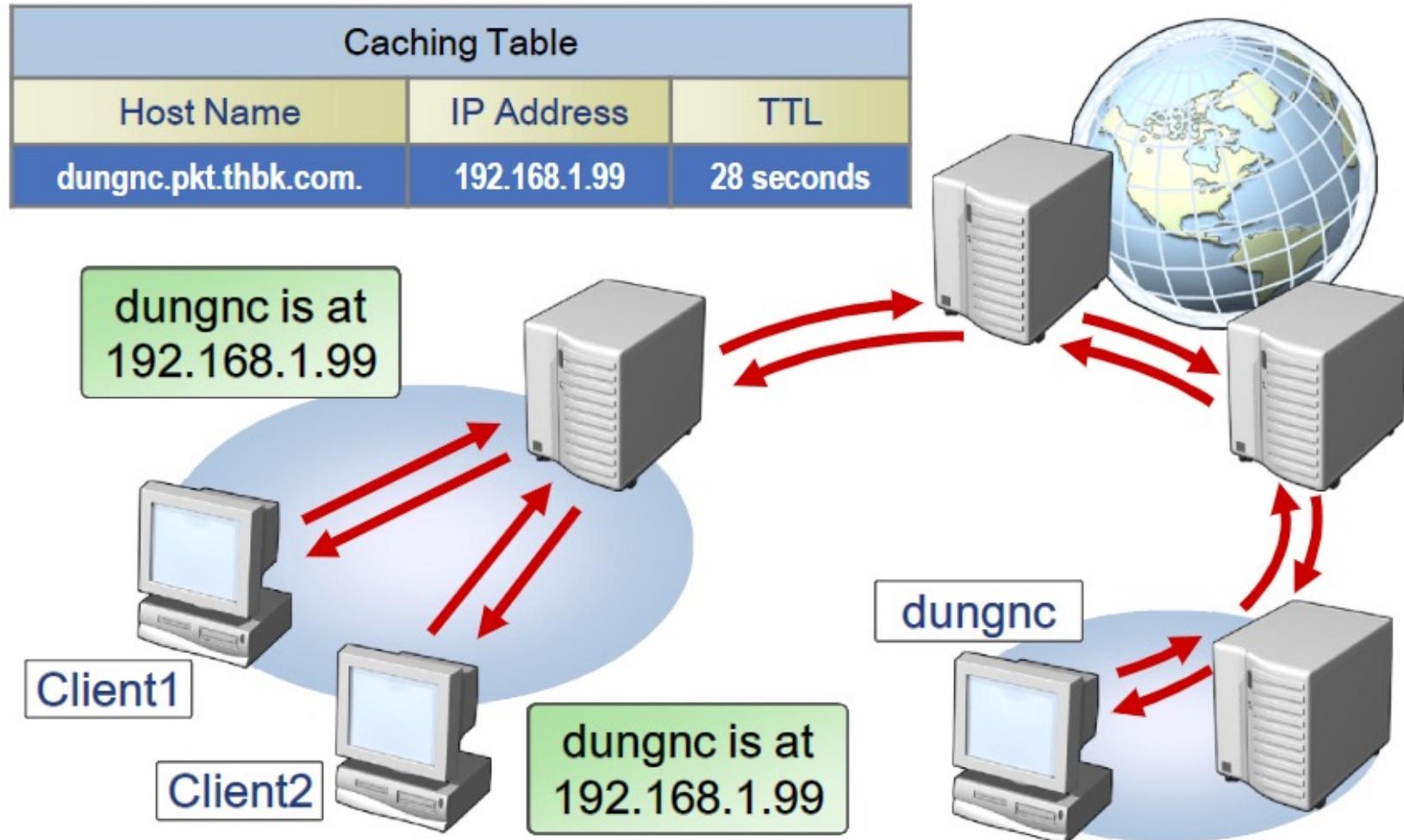




Hoạt động của DNS

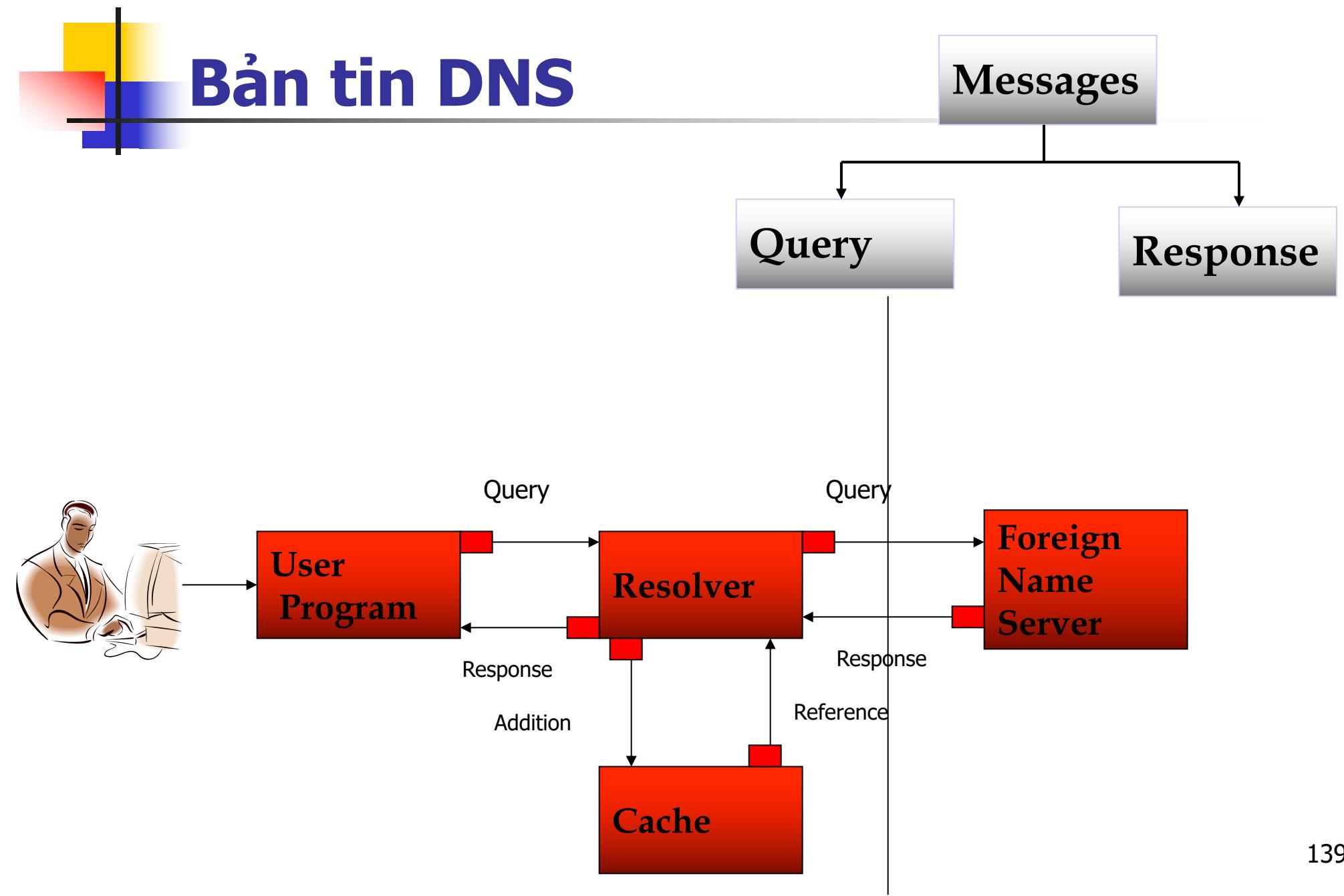
- DNS sử dụng lưu đệm DNS để cải thiện hiệu năng trễ và giảm số lượng bản tin DNS xuất hiện trên Internet
- Mỗi lần server tên miền học được ánh xạ, nó sẽ lưu đệm ánh xạ đó ở bộ nhớ nội bộ
 - Máy chủ DNS sẽ hủy thông tin đệm sau một khoảng thời gian (2 ngày) vì các trạm và ánh xạ giữa tên trạm chủ và địa chỉ IP không phải luôn cố định
 - Server tên miền mức cao (TLD) thường lưu đệm trong các server tên miền cục bộ
 - Như vậy server tên miền gốc không thường xuyên bị truy vấn
- Cơ cấu cập nhật/thông báo theo thiết kế của IETF
 - RFC 2136
 - <http://www.ietf.org/html.charters/dnsind-charter.html>

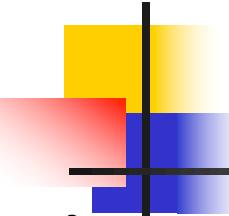
DNS Server Caching



Caching là một phương pháp lưu giữ những thông tin vừa được truy cập gần đây vào trong bộ nhớ riêng biệt của hệ thống, để lần sau nếu truy cập lại địa chỉ này sẽ nhanh hơn vì không phải tìm một lần nữa.

Bản tin DNS



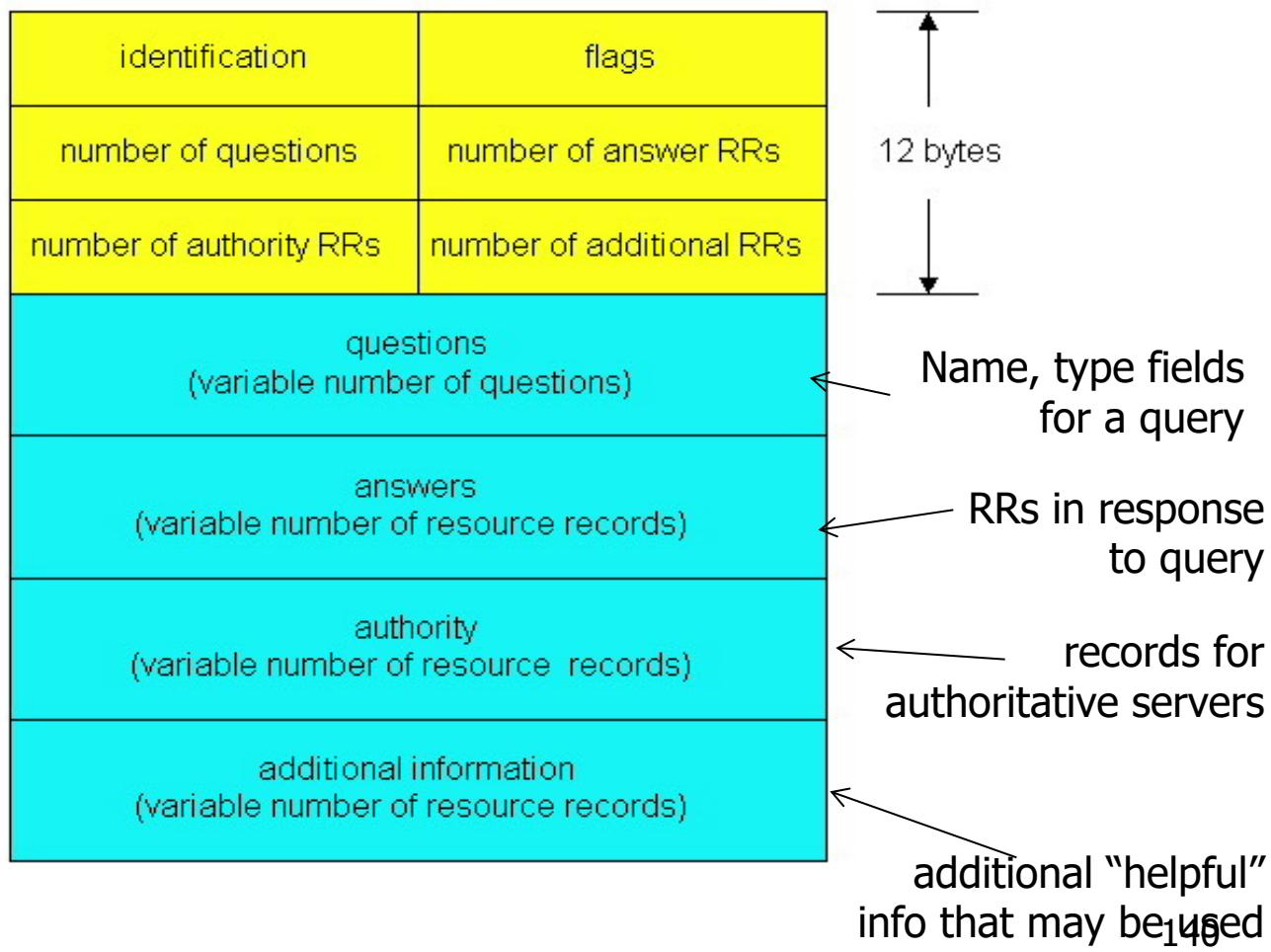


Bản tin DNS

Bản tin truy vấn - *query* và trả lời - *response* có cùng khuôn dạng

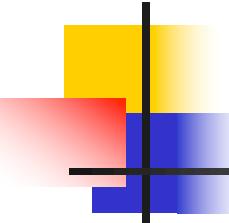
Tiêu đề: 12 byte:

- ❖ Định danh
- ❖ Cờ
- ❖ 4 trường số lượng, các trường này xác định số lượng các bản ghi trong 4 phần dữ liệu sau phần tiêu đề



Message format: Khuôn dạng bản tin

Recursion: đệ quy



Bản tin DNS

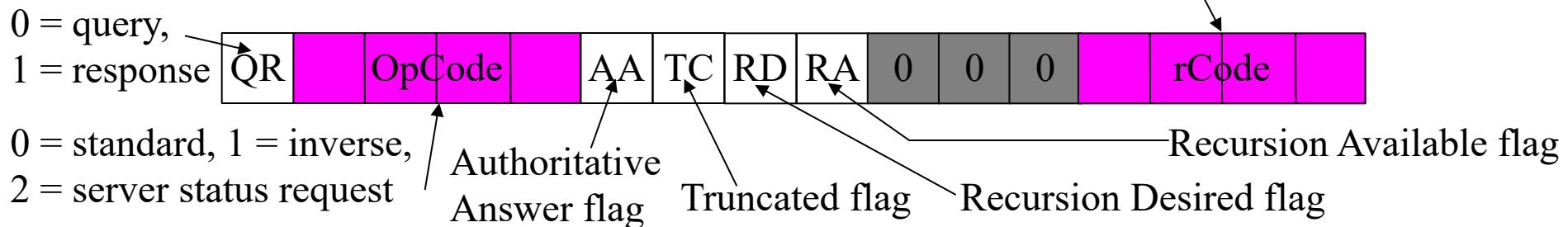
- ❖ Định danh: 16 bit cho mỗi thông điệp yêu cầu. 16 bit định danh này được ghi lại vào thông điệp trả lời, cho phép client xác định được đây là câu trả lời cho thông điệp yêu cầu nào
- ❖ Cờ: mỗi cờ ứng với một bit.
 - ❑ Cờ query/reply flag xác định thông điệp là yêu cầu (0) hay là trả lời (1).
 - ❑ Cờ authoritative được đặt trong thông điệp trả lời khi name server là authoritative name server của tên máy tính cần xác định địa chỉ IP.
 - ❑ Cờ mong muốn đệ quy (recursive-desired query) được đặt khi client (máy tính hay name server) mong muốn name server thực hiện truy vấn đệ quy khi nó không có bản ghi đó.
 - ❑ Cờ chấp nhận đệ quy (recursion-available flag) được đặt trong thông điệp trả lời nếu name server đó hỗ trợ đệ quy.

Bản tin DNS

❖ Cờ: mỗi cờ ứng với một bit.

- Cờ query/reply flag xác định thông điệp là yêu cầu (0) hay là trả lời (1).
- Cờ authoritative được đặt trong thông điệp trả lời khi name server là authoritative name server của tên máy tính cần xác định địa chỉ IP.
- Cờ mong muốn đệ quy (recursive-desired query) được đặt khi client (máy tính hay name server) mong muốn name server thực hiện truy vấn đệ quy khi nó không có bản ghi đó.
- Cờ chấp nhận đệ quy (recursion-available flag) được đặt trong thông điệp trả lời nếu name server đó hỗ trợ đệ quy.

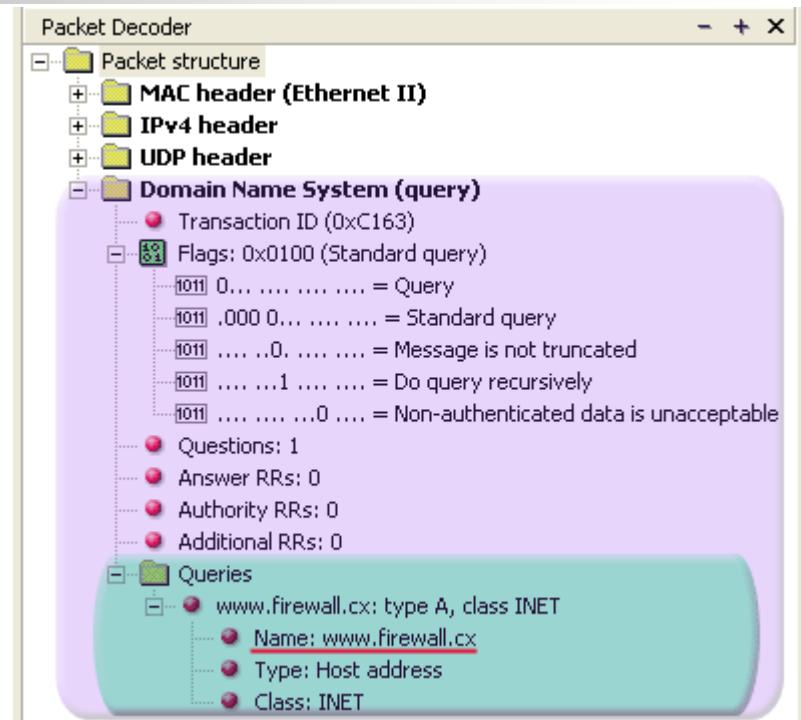
| | |
|------|-----------------------------|
| 0 | no error |
| 1 | format error |
| 2 | problem at name server |
| 3 | domain reference problem |
| 4 | query type not supported |
| 5 | administratively prohibited |
| 6-15 | reserved |

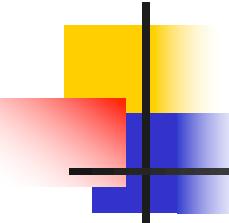


Bản tin DNS

| DNS Query | |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------|
| 2 Trans. ID: | The client uses this field to match responses to queries. |
| 2 Parameters: | Specifies the operation requested and a response code. |
| 2 Number of Questions: | Count of entries that appear in the <i>Question Section</i> . |
| 2 Number of Answers: | Count of entries that appear in the <i>Answer Section</i> . (Always set to zero in a Query) |
| 2 Number of Authority: | Count of entries that appear in the <i>Authority Section</i> . (Always set to zero in a Query) |
| 2 Number of Additional: | Count of entries that appear in the <i>Additional Section</i> . (Always set to zero in a Query) |
| <u>DNS Query Section</u> | |
| Query Domain Name: | The domain for which the query is sent. This field has a variable length. |
| 2 Query Type: | Encodes the type of question, e.g whether the question refers to a host address (A type) or mail address (MX type). |
| 2 Query Class: | Allows domain names to be used for arbitrary objects. |

All lengths shown are in bytes

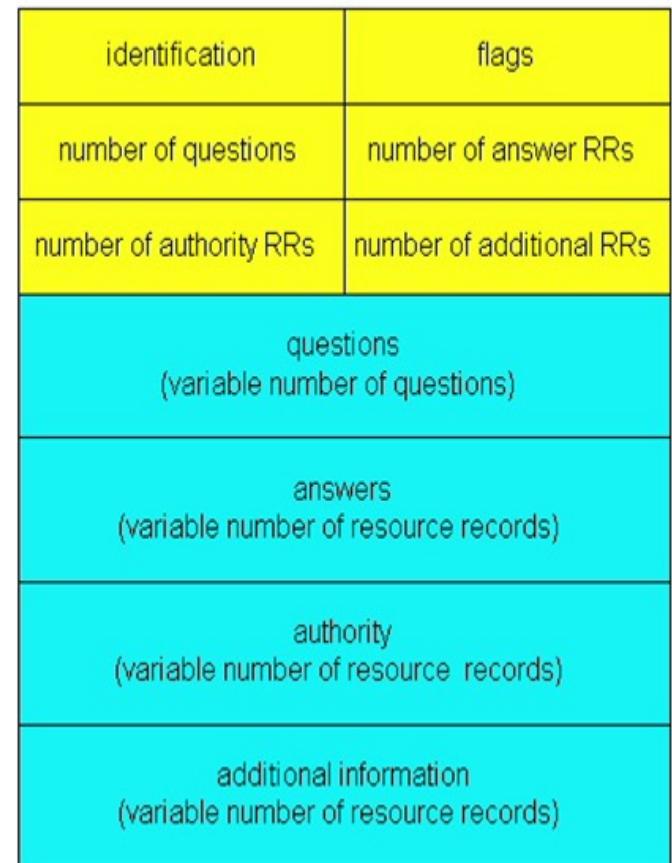




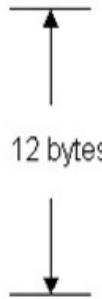
Bản tin DNS

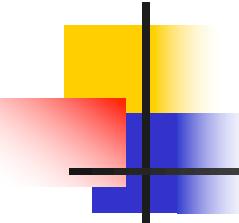
Bản tin truy vấn - *query* và trả lời - *response*

- ❖ Phần Question: chứa thông tin về câu hỏi được tạo ra, gồm (1) trường tên chứa tên đang được hỏi và (2) trường kiểu xác định kiểu câu hỏi cho tên máy tính đó (Kiểu A cho tên máy tính, kiểu MX cho mail server).
- ❖ Phần answer: Trong thông điệp trả lời từ server name, phần này chứa các bản ghi tài nguyên cho tên được yêu cầu trước đó. Chú ý rằng mỗi bản ghi tài nguyên có 4 trường: Type (A, NS, CNAME, MX,...), Name, Value, TTL. Thông điệp trả lời có thể có nhiều bản ghi tài nguyên vì tên máy tính có thể ứng với nhiều địa chỉ IP.



| | |
|-----------------------------------------------------------------|--------------------------|
| identification | flags |
| number of questions | number of answer RRs |
| number of authority RRs | number of additional RRs |
| questions (variable number of questions) | |
| answers (variable number of resource records) | |
| authority (variable number of resource records) | |
| additional information (variable number of resource records) | |

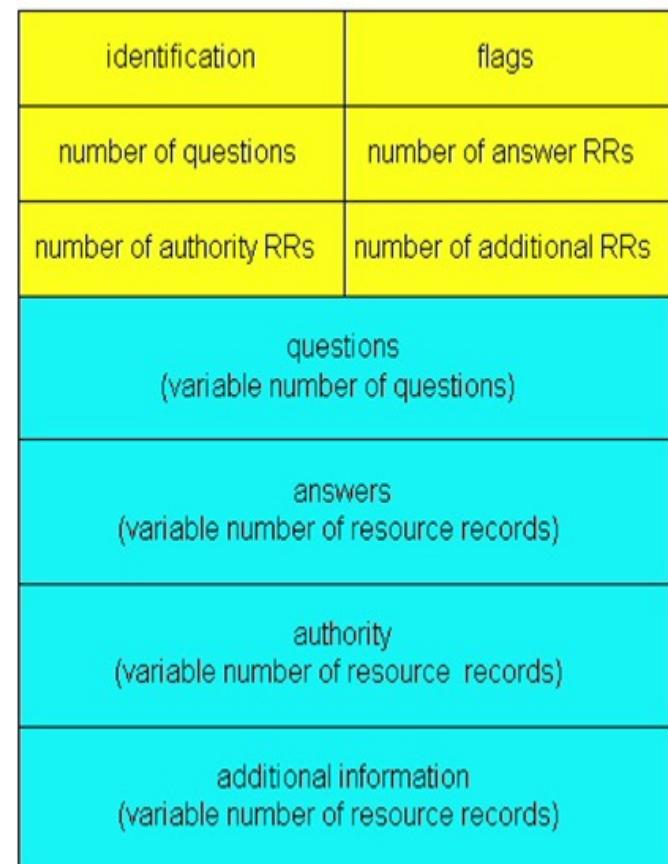




Bản tin DNS

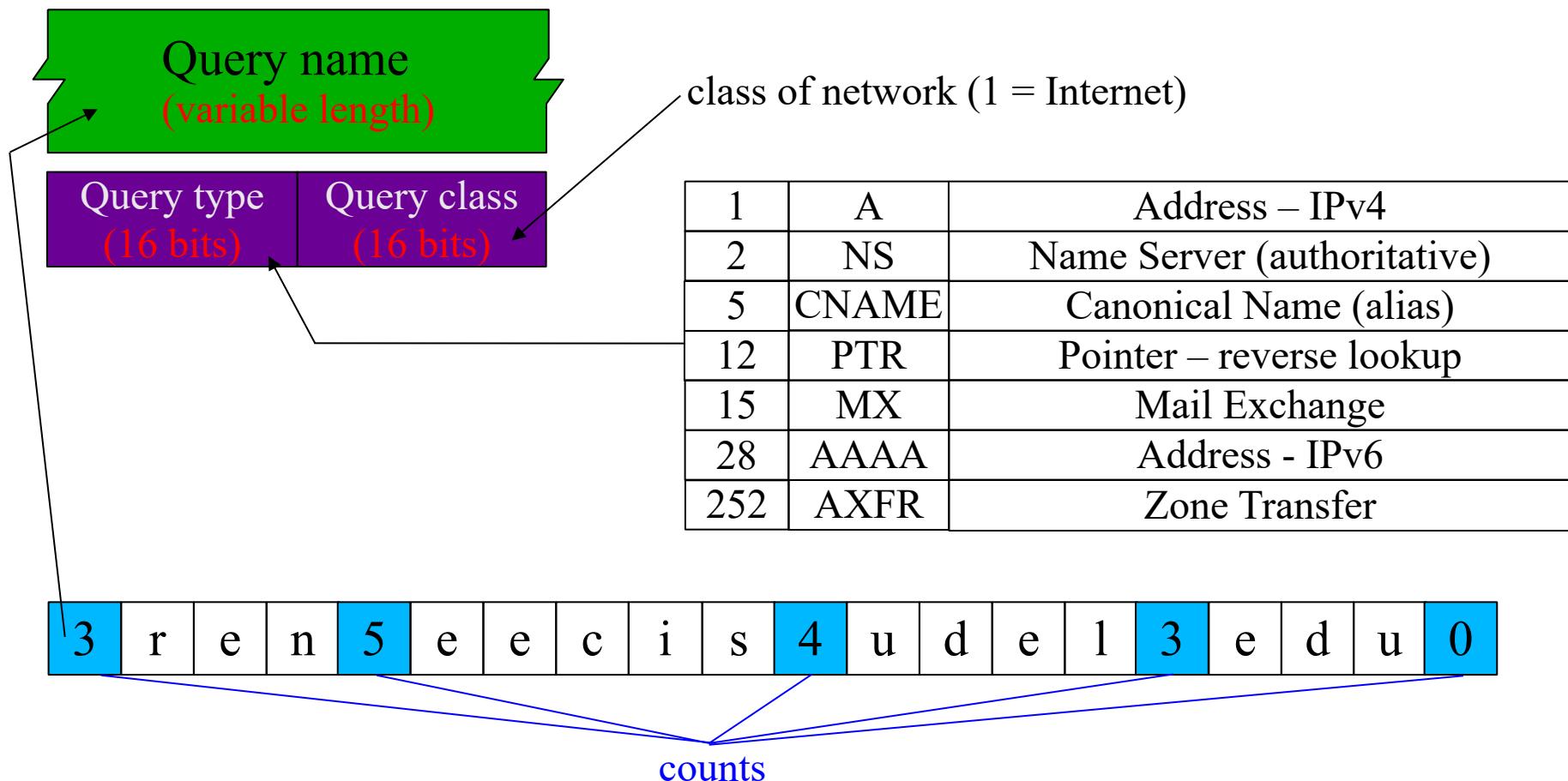
Bản tin truy vấn - *query* và trả lời - *response*

- ❖ Phần authority (thẩm quyền) chứa các bản ghi của các authoritative server.
- ❖ Phần additional (phụ trợ) chứa các bản ghi "hữu ích" khác. Ví dụ trường trả lời trong thông điệp trả lời một truy vấn MX sẽ chứa tên đầy đủ của mail server có tên bí danh đặt ở trong Name. Phần phụ trợ có thể có một bản ghi kiểu A cung cấp địa chỉ IP cho chính mail server đó.



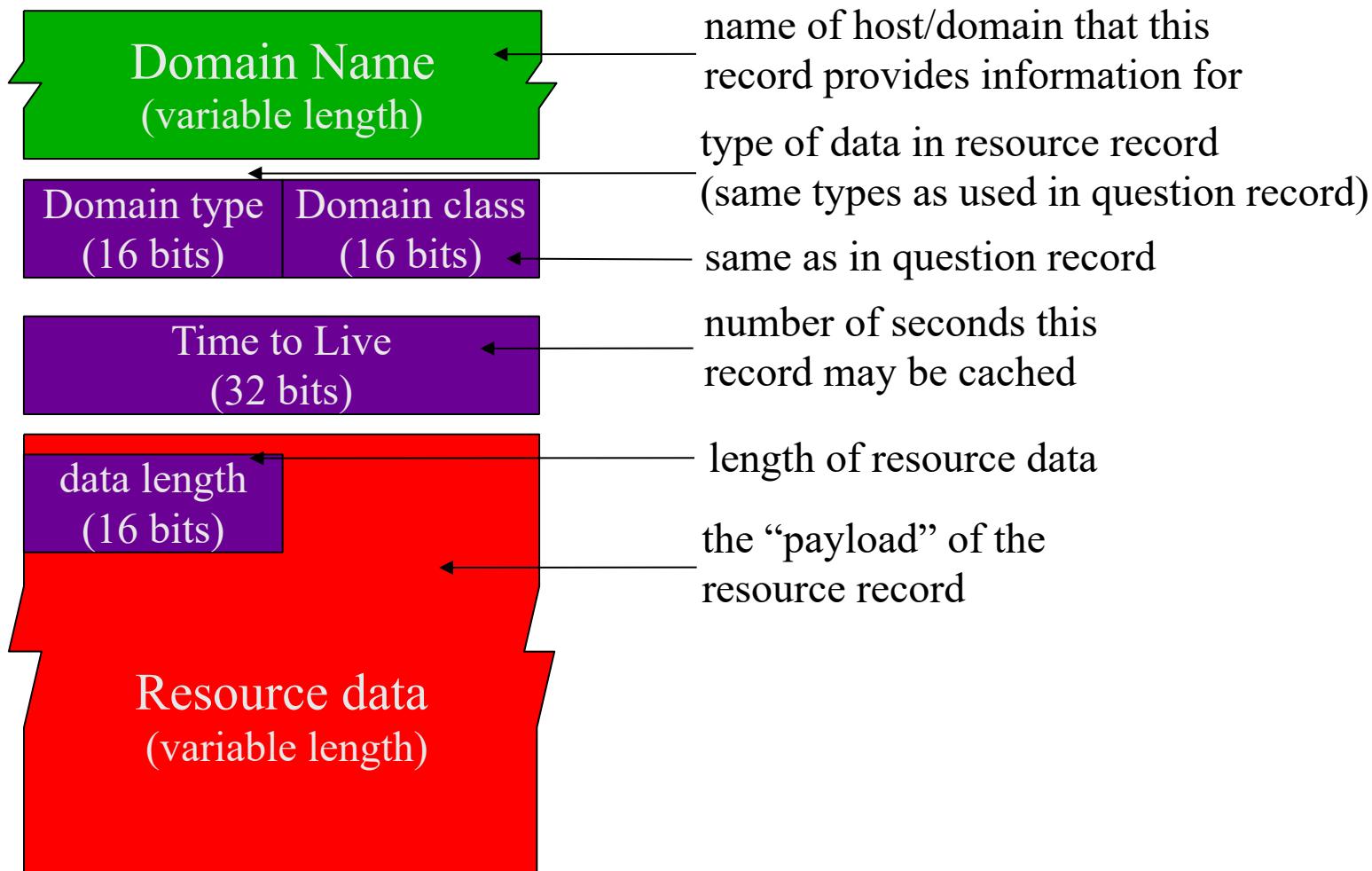
Bản tin DNS: Question Record Format

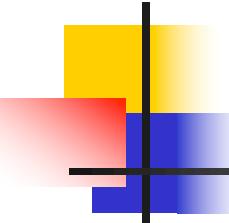
Được gửi trong bản tin query;
Được lặp lại trong bản tin response



Bản tin DNS: Resource Record Format

answer, authoritative, and additional sections in response





Bản ghi DNS

- Type = A / AAAA
 - Name = domain name
 - Value = IP address
 - A is IPv4, AAAA is IPv6

- Type = NS
 - Name = partial domain
 - Value = name of DNS server for this domain
 - “Go send your query to this other server”

Query Name: www.ccs.neu.edu
 Type: A

Resp. Name: www.ccs.neu.edu
 Value: 129.10.116.81

Query Name: ccs.neu.edu
 Type: NS

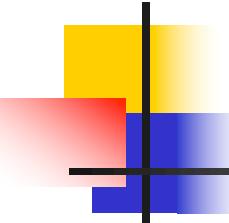
Resp. Name: ccs.neu.edu
 Value: 129.10.116.51

Bản ghi DNS

- Type = CNAME
 - Name = hostname
 - Value = canonical hostname (tên chính tắc)
 - Sử dụng cho bí danh

- Type = MX
 - Name = domain in email address
 - Value = canonical name of mail server

| | |
|-------|-------------------------------------------------------------|
| Query | Name: <u>foo.mysite.com</u> Type: CNAME |
| Resp. | Name: <u>foo.mysite.com</u> Value: <u>bar.mysite.com</u> |
| Query | Name: <u>ccs.neu.edu</u> Type: MX |
| Resp. | Name: <u>ccs.neu.edu</u> Value: <u>amber.ccs.neu.edu</u> |



Bản ghi DNS

Phân giải ngược:

- Ánh xạ từ địa chỉ IP sang tên miền
- Không gian tên miền ngược cũng được xây dựng theo cơ chế phân cấp giống như tên miền thuận.
 - Rooted at in-addr.arpa and ip6.arpa
- Bản ghi DNS ngược có **type**: PTR
 - Name = IP address
 - Value = domain name

Query

Name: 129.10.116.51
Type: PTR

Resp.

Name: 129.10.116.51
Value: ccs.neu.edu

Example

DNS Name Field Length Example 1 - www.cisco.com

Packet Decoder

- Packet structure
 - MAC header (Ethernet II)
 - IPv4 header
 - UDP header [8+31=39 bytes]**
 - Source port = 1026 UDP->516
 - Destination port = 53 UDP->13568
 - Length = 39 bytes
 - Checksum =0x7598 (Correct)
 - Domain Name System (query)**
 - Transaction ID (0x7292) 2 bytes
 - Flags: 0x0100 (Standard query) 2 bytes
 - Questions: 1 2 bytes
 - Answer RRs: 0 2 bytes
 - Authority RRs: 0 2 bytes
 - Additional RRs: 0 2 bytes
 - Queries
 - www.cisco.com: type A, class INET
 - Name: www.cisco.com
 - Type: Host address 2 bytes
 - Class: INET 2 bytes

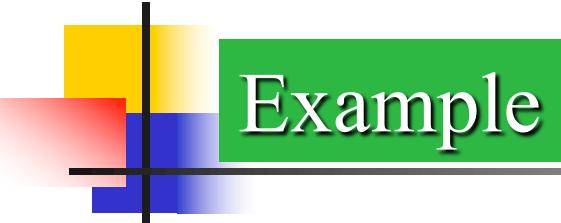
In this example, the length field in our UDP header is 39 bytes. The actual UDP Header is 8 bytes long and the DNS query section 31 bytes. By subtracting all known field lengths (typed using red) in the DNS query section, we are left with the length of the DNS Name field which is $31-16=15$ bytes !

DNS Name Field Length Example 2 - support.novell.com

Packet Decoder

- Packet structure
 - MAC header (Ethernet II)
 - IPv4 header
 - UDP header [8+36=44 bytes]**
 - Source port = 1026 UDP->516
 - Destination port = 53 UDP->13568
 - Length = 44 bytes
 - Checksum =0xA1C6 (Correct)
 - Domain Name System (query)**
 - Transaction ID (0x90D1) 2 bytes
 - Flags: 0x0100 (Standard query) 2 bytes
 - Questions: 1 2 bytes
 - Answer RRs: 0 2 bytes
 - Authority RRs: 0 2 bytes
 - Additional RRs: 0 2 bytes
 - Queries
 - support.novell.com: type A, class INET
 - Name: support.novell.com
 - Type: Host address 2 bytes
 - Class: INET 2 bytes

In this example, the length field in our UDP header is 44 bytes. The actual UDP Header is 8 bytes long and the DNS query section 36 bytes. By subtracting all known field lengths (typed using red) in the DNS query section, we are left with the length of the DNS Name field which is $36-16=20$ bytes !



Example

A resolver sends a query message to a local server to find the IP address for the host “chal.fhda.edu.”. We discuss the query and response messages separately. Figure 19.21 shows the query message sent by the resolver. The first 2 bytes show the identifier $(1333)_{16}$. It is used as a sequence number and relates a response to a query. The next bytes contain the flags with the value of 0x0100 in hexadecimal. In binary it is 0000000100000000, but it is more meaningful to divide it into the fields as shown below:

| QR | OpCode | AA | TC | RD | RA | Reserved | rCode |
|----|--------|----|----|----|----|----------|-------|
| 0 | 0000 | 0 | 0 | 1 | 0 | 000 | 0000 |

Example: *Continued*

Figure 19.22 shows the response of the server. The response is similar to the query except that the flags are different and the number of answer records is one. The flags value is 0x8180 in hexadecimal. In binary it is 1000000110000000, but again we divide it into fields as shown below:

| QR | OpCode | AA | TC | RD | RA | Reserved | rCode |
|----|--------|----|----|----|----|----------|-------|
| 1 | 0000 | 0 | 0 | 1 | 1 | 000 | 0000 |

Example forward query/response

“What is the IP address
of www.udel.edu?”

| | | | |
|-----|--------------|--------|---|
| Hdr | <i>ident</i> | 0x0100 | ← |
| | 0x0001 | 0x0000 | |
| | 0x0000 | 0x0000 | |

| | | | | |
|-----|-------------|-----|------------|-----|
| Qry | 3 | 'w' | 'w' | 'w' |
| | 4 | 'u' | 'd' | 'e' |
| | '1' | 3 | 'e' | 'd' |
| | 'u' | 0 | 0x0001 (A) | |
| | 0x0001 (IN) | | | |

flags: recursion desired (RD)

flags: query response (QR),
recursion desired (RD),
recursion available (RA)

TTL: 45301 seconds ≈ 12.6 hours

“www.udel.edu's IP address
is 128.175.13.63.”

| | | |
|------------|--------|---|
| same ident | 0x8180 | ↓ |
| 0x0001 | 0x0001 | |
| 0x0004 | 0x0004 | |

| | | | | |
|-----|-------------|-----|------------|-----|
| Qry | 3 | 'w' | 'w' | 'w' |
| | 4 | 'u' | 'd' | 'e' |
| | '1' | 3 | 'e' | 'd' |
| | 'u' | 0 | 0x0001 (A) | |
| | 0x0001 (IN) | | | |

| | | |
|-----|-------------|-----------------|
| Ans | 0xC00C | 0x0001 (A) |
| | 0x0001 (IN) | 0x0000... |
| | ...0xB2F5 | 0x0004 |
| | 0x80AF0D3F | (128.175.13.63) |

Example inverse query/response

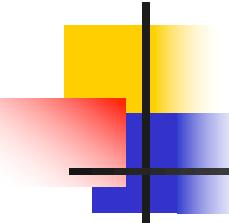
“What is the name of the host at 128.175.13.63?”

| Hdr | <i>ident</i> | 0x0100 | | |
|-----|--------------|------------|-----|-----|
| | 0x0001 | 0x0000 | | |
| | 0x0000 | 0x0000 | | |
| Qry | 2 | '6' | '3' | 2 |
| | '1' | '3' | 3 | '1' |
| | '7' | '5' | 3 | '1' |
| | '2' | '8' | 7 | 'i' |
| | 'n' | '-' | 'a' | 'd' |
| | 'd' | 'r' | 4 | 'a' |
| | 'r' | 'p' | 'a' | 0 |
| | 0x000C(PTR) | 0x0001(IN) | | |

“The host at 128.175.13.63 is named www.udel.edu.”

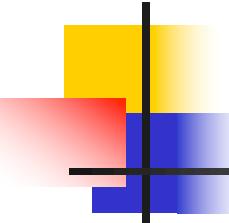
| Hdr | <i>same ident</i> | 0x8180 | | |
|-----|-------------------|-------------|-----|-----|
| | 0x0001 | 0x0001 | | |
| | 0x0004 | 0x0004 | | |
| Qry | 2 | '6' | '3' | 2 |
| | '1' | '3' | 3 | '1' |
| | '7' | '5' | 3 | '1' |
| | '2' | '8' | 7 | 'i' |
| | 'n' | '-' | 'a' | 'd' |
| | 'd' | 'r' | 4 | 'a' |
| | 'r' | 'p' | 'a' | 0 |
| | 0x000C(PTR) | 0x0001(IN) | | |
| | 0xC00C | 0x000C(PTR) | | |
| | 0x0001(IN) | 0x0000... | | |
| | ...0xB003 | 0x000E | | |
| | 3 | 'w' | 'w' | 'w' |
| | 4 | 'u' | 'd' | 'e' |
| | 'l' | 3 | 'e' | 'd' |
| | 'u' | 0 | | |
| | | | | ... |

TTL: 45056 seconds ≈ 12.5 hours



Các điểm yếu an toàn của DNS

- Tấn công tràn ngập băng thông DDoS
- Gửi lượng truy vấn DNS rất lớn tới các máy chủ TLD
- Kiểu tấn công người trung gian (man-in-the-middle)
- Khai thác cơ sở hạ tầng DNS để khởi động tấn công DDoS chống lại trạm chủ mục tiêu.



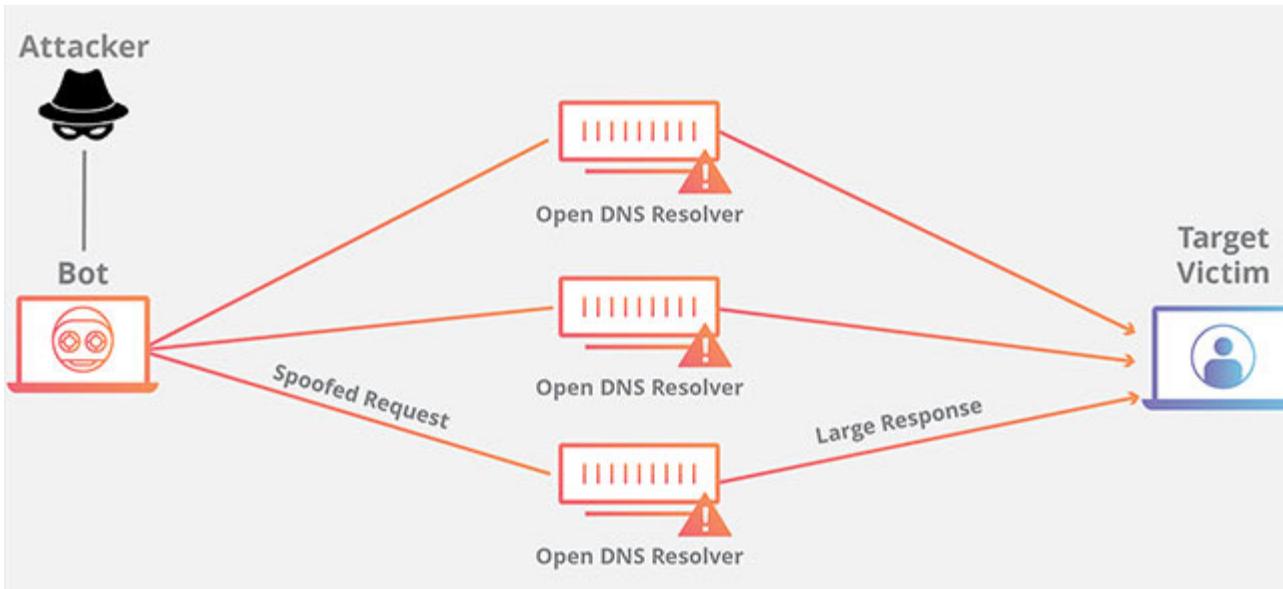
Các điểm yếu an toàn của DNS

- Tấn công tràn ngập băng thông DDoS:

- Kẻ tấn công có thể cố gửi vào từng máy chủ gốc DNS một lượng lớn gói tin làm tràn lụt máy chủ
- Năm 2002, kẻ tấn công tận dụng mạng ma (botnet) gửi những lượng bản tin ICMP khổng lồ vào toàn bộ 13 máy chủ gốc DNS
- Tuy nhiên, các máy chủ gốc được bảo vệ bởi các bộ lọc gói, được cấu hình luôn chặn các bản tin ICMP ping trực tiếp vào các máy chủ gốc. Vì vậy, những máy chủ này được bảo vệ và hoạt động bình thường.
- Hơn nữa, hầu hết máy chủ DNS cục bộ đệm địa chỉ IP của các máy chủ TLD nên nó vẫn xử lý truy vấn mà không cần phải chuyển qua máy chủ gốc DNS.

Các điểm yếu an toàn của DNS

- Gửi lượng truy vấn DNS rất lớn tới các máy chủ TLD



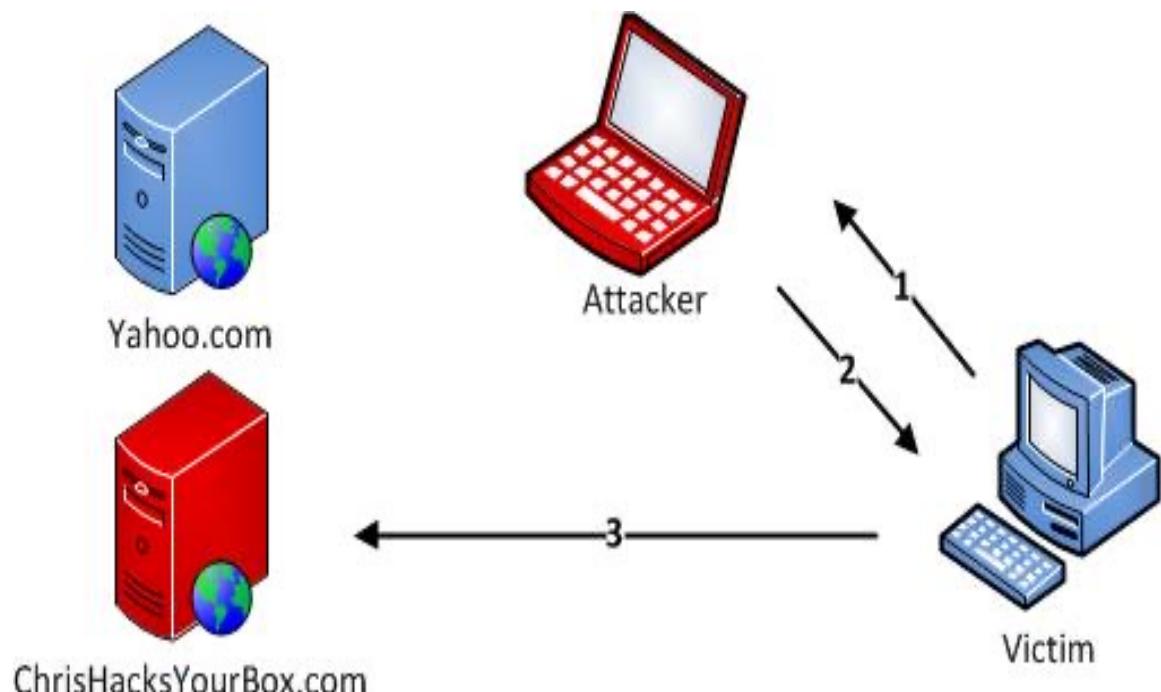
Tấn công Distributed Denial of Service (DDoS)

Các điểm yếu an toàn của DNS

- Kiểu tấn công người trung gian (man-in-the-middle)

Kẻ tấn công chặn truy vấn từ các trạm và trả về bản tin trả lời giả.

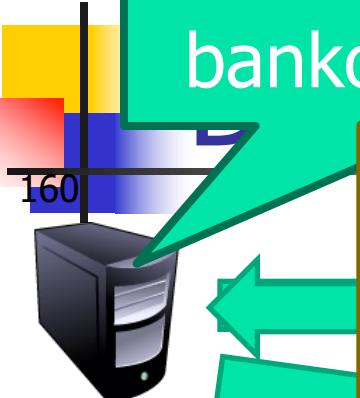
**Giải pháp?
DNSSEC?**



1. Legitimate DNS Request Destined for DNS Server
2. Fake DNS Reply from Listening Attacker
3. Victim begins communicating with malicious site as a result

Where is
bankofamerica.com?

123.45.67.89



How do you know that a given
name → IP mapping is correct?

bankofAmerica



Where is
bankofamerica.com?

66.66.66.93

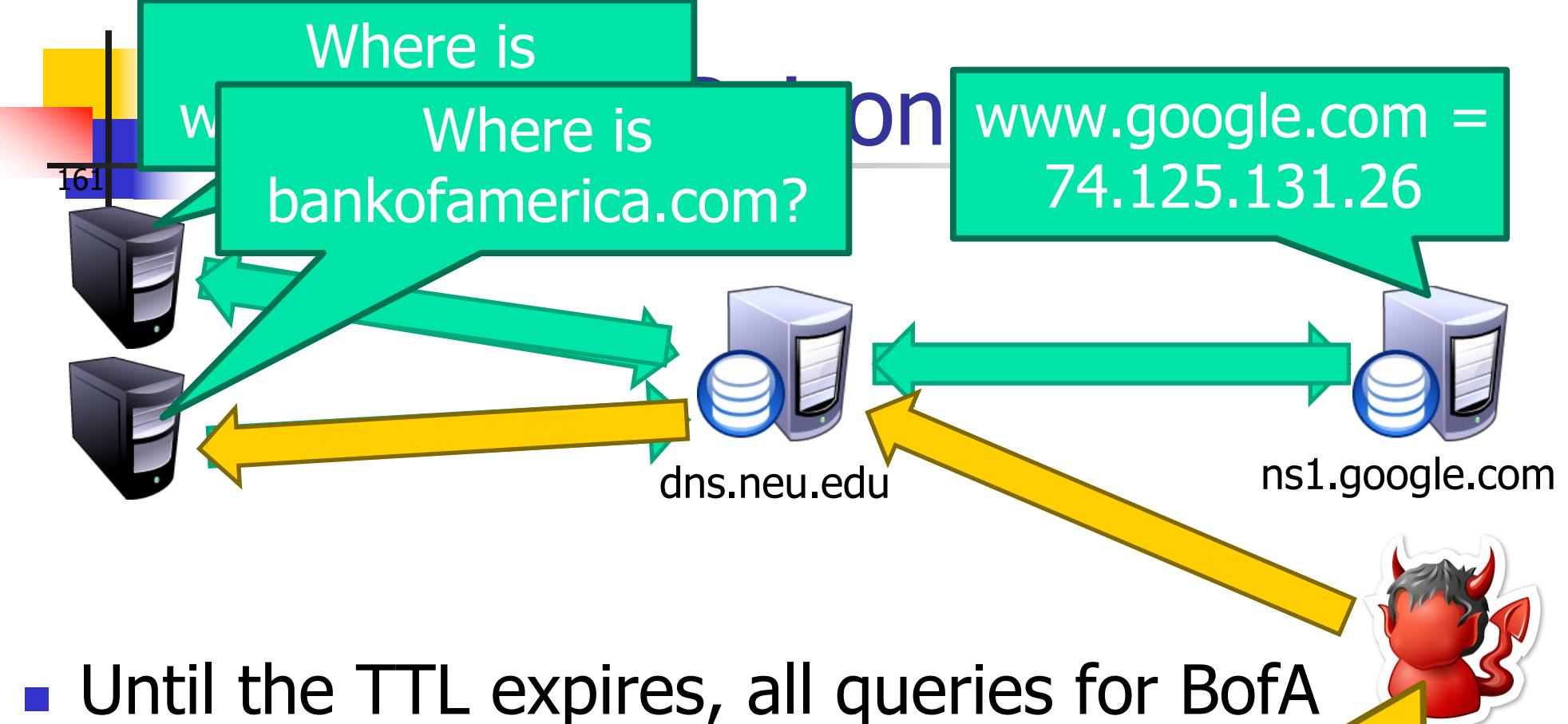
123.45.67.89



dns.evil.com



66.66.66.93



- Until the TTL expires, all queries for BofA to dns.neu.edu will return poison!
- Much worse than spoofing/middle man in the middle
 - Whole ISPs can be impacted!

bankofamerica.com
= 66.66.66.92

Chương 2: Các ứng dụng và giao thức

2.1 Tổng quan về các ứng dụng và dịch vụ hạ tầng

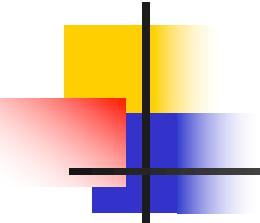
2.2 Ứng dụng WEB và các giao thức

2.3 Ứng dụng truyền tệp và các giao thức

2.4 Ứng dụng thư điện tử và các giao thức

2.5 Ứng dụng hệ thống tên miền

2.6 Các ứng dụng mạng ngang hàng



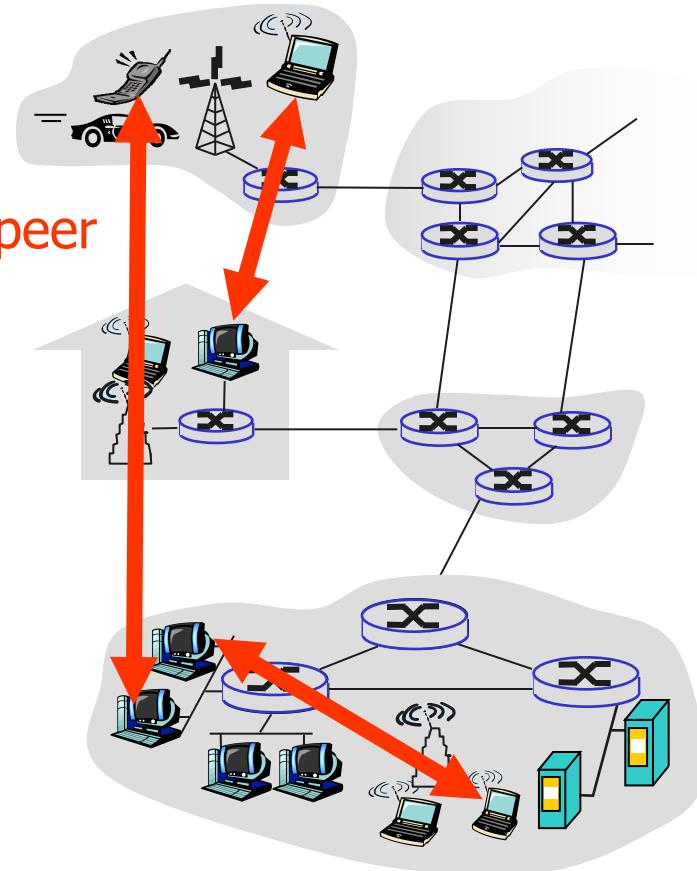
Kiến trúc mạng P2P

Hạn chế của Mô hình client-server:

- Khó mở rộng: số lượng máy khách tăng thì nhu cầu về tải, băng thông tăng lên và máy chủ có thể không có khả năng để cung cấp dịch vụ và mở rộng
- Độ tin cậy thấp: Nếu số lượng server nhỏ, lỗi server thì gây ra lỗi hệ thống (single point of failure)
- Tính hiệu quả không cao: khi người dùng trải đều trên toàn cầu, thì việc đáp trả yêu cầu của người dùng sẽ bị trễ lớn.
- Yêu cầu việc quản trị tốt

Kiến trúc mạng P2P

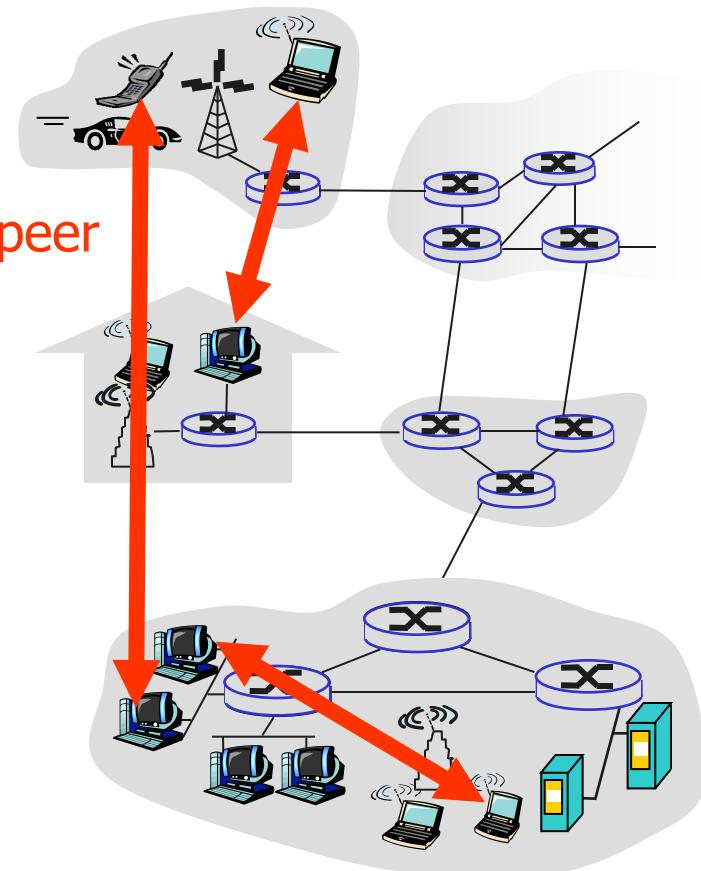
- Mô hình P2P được xem như một giải pháp khắc phục các nhược điểm của mô hình client-server
- Một mạng ngang hàng không có khái niệm máy chủ và máy khách, chỉ có peer **peer-peer**
- Kiến trúc P2P dựa vào rất ít (hoặc không dựa vào) các máy chủ
- P2P là các hệ thống phân tán, không có sự phân cấp hay điều khiển tập trung
- **Phân biệt:**
 - **Client/server: Các máy tính thực hiện các chức năng bất đối xứng**
 - **P2P: các máy tính thực hiện các chức năng đối xứng**



Kiến trúc mạng P2P

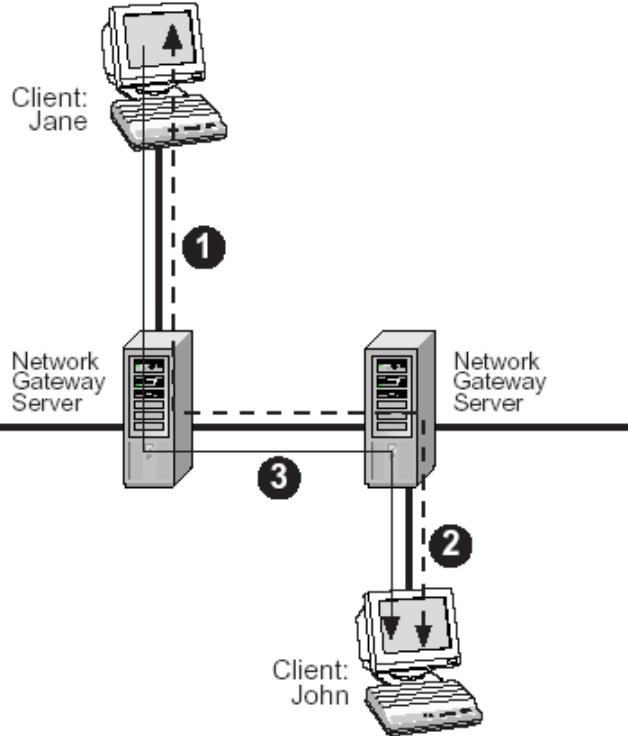
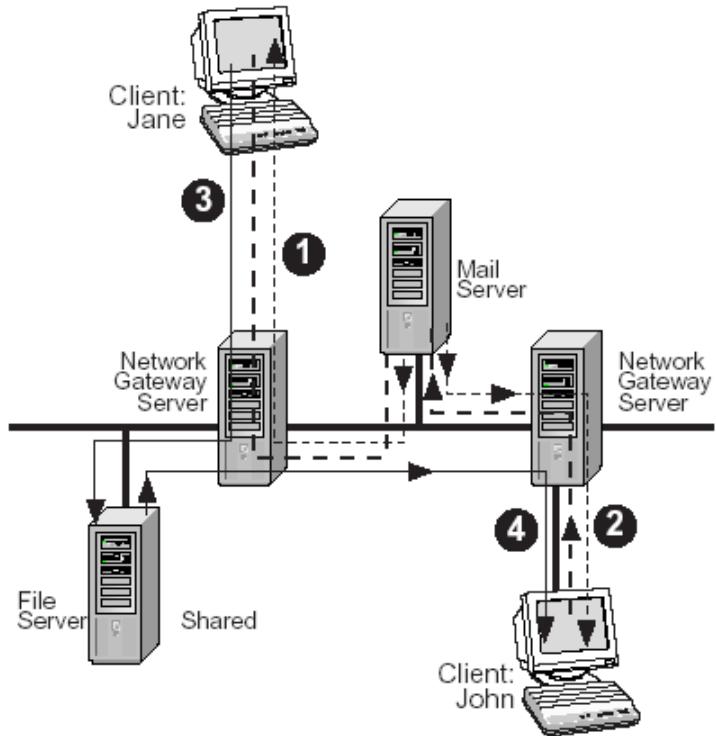
Các đặc điểm của mạng P2P

- Máy chủ không cần hoạt động liên tục.
- Các hệ thống đầu cuối tùy ý kết nối trực tiếp, vừa là client vừa là server
- Các node là các hệ tự trị, không cần được quản lý**
- Các thiết bị ngang hàng không kết nối liên tục và thay đổi địa chỉ IP.
- Có khả năng cho mạng qui mô lớn nhưng khó quản lý.
- Mạng là động** vì các node có thể tham gia hoặc rời khỏi mạng thường xuyên
- Phân thành hai loại: **mạng không cấu trúc** và **mạng có cấu trúc**.



Client-Server vs. Peer-to-Peer Example

C l i e n t - S e r v e r



- ① John mails Jane: "I need the file "Plan.doc." Jane reads when synchronizing her inbox."
- ② John "pulls" Jane's mail reply: "I will place Plan.doc on group's shared drive."
- ③ Jane uploads file to shared drive.
- ④ John downloads the file.

- ① Instant Message (IM), John to Jane: "I will need the file Plan.doc."
- ② IM, Jane to John: "It's on my shared storage now."
- ③ John downloads the file from Jane's computer.

Kiến trúc mạng P2P

Phân loại mạng ngang hàng:

- Theo mục đích sử dụng:
 - Chia sẻ file (file sharing)
 - Điện thoại VoIP
 - Đa phương tiện media streaming (audio, video)
 - Diễn đàn thảo luận (Discussion forums)
- Theo topo mạng ở tầng vật lý và mạng phủ
 - Mạng ngang hàng lai (hybrid peer to peer)
 - Mạng ngang hàng thuần túy (pure peer to peer)
 - Kiến trúc siêu ngang hàng (super peer architecture)
 - Mạng ngang hàng có cấu trúc (structured)

Kiến trúc mạng P2P

Phân loại mạng ngang hàng: Mạng ngang hàng lai

- Đặc điểm:
 - Vẫn dựa vào máy chủ tìm kiếm trung tâm
 - Cấu trúc Overlay của mạng ngang hàng lai như mạng hình sao
- Nguyên tắc hoạt động:
 - Mỗi client lưu trữ các files định chia sẻ với các nút khác trong mạng
 - Có một bảng lưu trữ thông tin kết nối của người dùng đăng ký (IP address, connection bandwidth,...)
 - Có một bảng liệt kê danh sách các file mà mỗi người dùng định chia sẻ (tên file, dung lượng, thời gian tạo file,...)
 - Các yêu cầu tìm kiếm được gửi tới máy chủ trung tâm phân tích. Nếu yêu cầu được thực hiện thì máy chủ sẽ trả về địa chỉ IP của máy chứa tài nguyên trong mạng.
 - Quá trình truyền file thực hiện theo đúng cơ chế của mạng P2P

Kiến trúc mạng P2P

Phân loại mạng ngang hàng: Mạng ngang hàng lai

- **Ưu điểm:**
 - Dễ xây dựng
 - Tìm kiếm file nhanh và hiệu quả
- **Nhược điểm:**
 - Vấn đề về luật pháp và bản quyền khi chia sẻ tài nguyên trên mạng
 - Dễ bị tấn công
 - Nghẽn cổ chai
 - Cần quản trị tập trung ở máy chủ



**Mạng ngang hàng lai
(Hybrid Peer to Peer system)**

Kiến trúc mạng P2P

Phân loại mạng ngang hàng: Mạng ngang hàng thuần túy

- Đặc điểm:
 - Không có máy chủ tìm kiếm tập trung
 - Đảm bảo tính phân tán hoàn toàn cho các nút tham gia
 - Các nút có thể tham gia hoặc rời khỏi mạng một cách tùy ý mà không ảnh hưởng đến cấu trúc mạng



**Mạng ngang hàng
thuần túy**

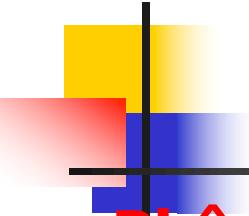
Kiến trúc mạng P2P

Phân loại mạng ngang hàng: Mạng ngang hàng thuần túy

- **Ưu điểm:**
 - Dễ xây dựng
 - Khắc phục được vấn đề nút cổ chai trong mô hình tập trung
- **Nhược điểm:**
 - Tốn băng thông
 - Phức tạp trong quá trình tìm kiếm
 - Các nút có khả năng khác nhau (CPU power, bandwidth, storage) đều phải chịu tải như nhau



**Mạng ngang hàng
thuần túy**



Kiến trúc mạng P2P

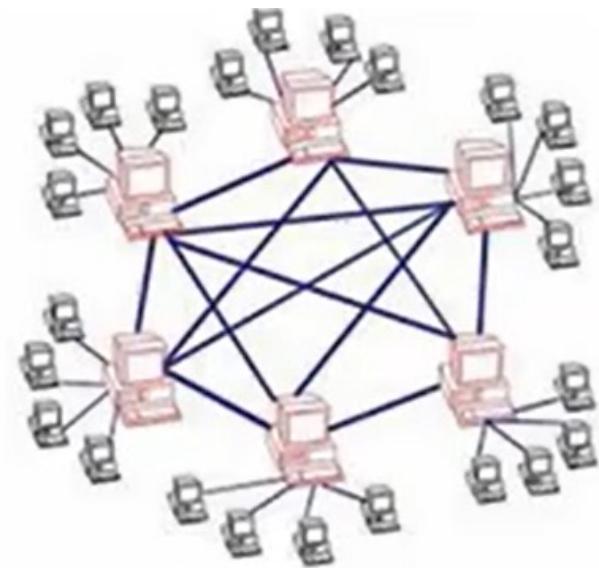
Phân loại mạng ngang hàng: Mạng siêu ngang hàng

- **Đặc điểm:**
 - Là mạng ngang hàng thế hệ 2
 - Giao thức tiêu biểu là JXTA (Juxtapose), là giao thức mã nguồn mở
- **Nguyên tắc hoạt động:**
 - Tồn tại trật tự phân cấp bằng cách định nghĩa super-peer
 - Các super-peer tạo thành một mạng không cấu trúc
 - Mỗi super-peer có kết nối đến nhiều client-peer
 - Mỗi super-peer chứa một danh sách các file được cung cấp bởi các client-peer và địa chỉ IP của chúng, do đó nó có thể trả lời ngay lập tức các yêu cầu truy vấn từ client-peer gửi tới.
 - Khi một thiết bị ngang hàng muốn tìm dữ liệu trong mạng ngang hàng không cấu trúc, yêu cầu tìm kiếm sẽ được truyền trên toàn bộ mạng để tìm càng nhiều thiết bị chia sẻ càng tốt

Kiến trúc mạng P2P

Phân loại mạng ngang hàng: Mạng siêu ngang hàng

- **Ưu điểm:** khắc phục được các nhược điểm của mạng thuần túy
 - Giảm lưu lượng trong mạng nhưng vẫn tránh được nghẽn cổ chai
 - Super-peer chịu tải chính, các node khác chịu tải nhẹ hơn
- **Nhược điểm:**
 - Mỗi super-peer trở thành điểm gây lỗi cho nhóm siêu ngang hàng tương ứng trong trường hợp số lượng client trong nhóm là rất lớn



**Kiến trúc siêu ngang hàng
(Super-peer Architecture)**

Kiến trúc mạng P2P

Phân loại mạng ngang hàng: Mạng P2P có cấu trúc

Đặc điểm:

- Các nút mạng duy trì thông tin định tuyến để tìm ra tất cả các nút trong mạng
- Topo mạng được kiểm soát chặt chẽ, khả năng mở rộng mạng được nâng cao rõ rệt.
- Mô hình có cấu trúc thường đem lại hiệu quả tìm kiếm cao, phù hợp với mạng có quy mô lớn

Nguyên tắc hoạt động:

- Các file (hoặc con trỏ tới file) được đặt ở một vị trí xác định
- Cung cấp sự liên kết (mapping) giữa nội dung (ví dụ ID của tệp) và vị trí (ví dụ địa chỉ nút)
 - Liên kết này thường dựa trên một cấu trúc dữ liệu bảng hàm băm phân tán DHT.



**Mạng ngang hàng có
cấu trúc (Structured)**

Kiến trúc mạng P2P

Phân loại mạng ngang hàng: Mạng P2P có cấu trúc

- **Ưu điểm:** khắc phục được các nhược điểm của mạng không có cấu trúc
 - Khả năng mở rộng được cải thiện do không có điểm tập trung gây ra nghẽn cổ chai tại các điểm đó
 - Các truy vấn tìm kiếm được phát đi theo thuật toán cụ thể, hạn chế tối đa lượng truy vấn, tiết kiệm băng thông
- **Nhược điểm:**
 - Việc quản lý cấu trúc của topo mạng gấp khó khăn, đặc biệt trong kịch bản tỉ lệ vào/ra mạng của các nút cao
 - Vấn đề cân bằng tải trọng mạng
 - Sự khác biệt về topo trên mạng overlay và mạng liên kết vật lý dẫn đến thời gian truy vấn trung bình cao

Ứng dụng của mạng P2P

- Giao tiếp (Communication)
- Chia sẻ file (File Sharing)
- Lưu trữ (Store)
- Các chương trình xử lý (Process)

Ứng dụng của mạng P2P

■ Ứng dụng trong giao tiếp (communication)

- Đóng vai trò quan trọng trong các ứng dụng mạng ngang hàng
- Cung cấp các thông tin và các nguồn tài nguyên sẵn có trên mạng
- Tạo khả năng các trạm kết nối trực tiếp với các trạm khác và yêu cầu tài nguyên.
- Ví dụ: Instant messaging: Napster, ICQ, Jabber



Ứng dụng của mạng P2P

■ Ứng dụng trong giao tiếp (communication)

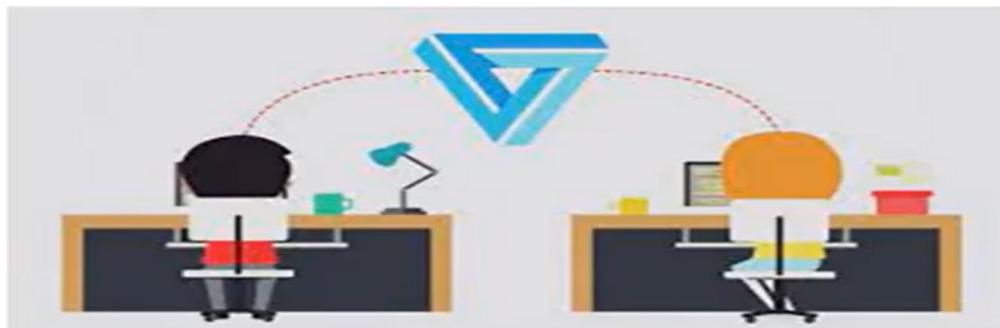
- Server trung tâm lưu trữ thông tin và danh sách người dùng đăng ký
- Khi có sự giao tiếp giữa các node, việc tìm kiếm người dùng khác được thực hiện trên server
- Khi không online, hệ thống lưu trữ các tin nhắn cho đến khi người dùng online



Ứng dụng của mạng P2P

■ Chia sẻ file:

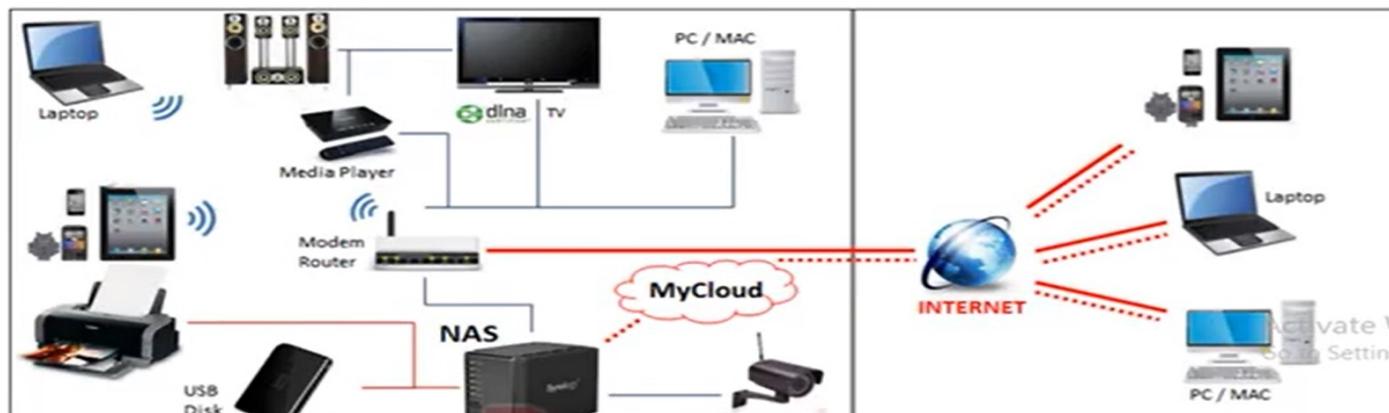
- Là ứng dụng được dùng nhiều nhất
- 70% lưu lượng mạng trên Internet dùng cho việc trao đổi file
- Đặc điểm khi chia sẻ file là các Peer có các file được download với vai trò là một Client làm cho chúng luôn sẵn sàng với các Peer khác trong vai trò của một Server.



Ứng dụng của mạng P2P

■ Lưu trữ:

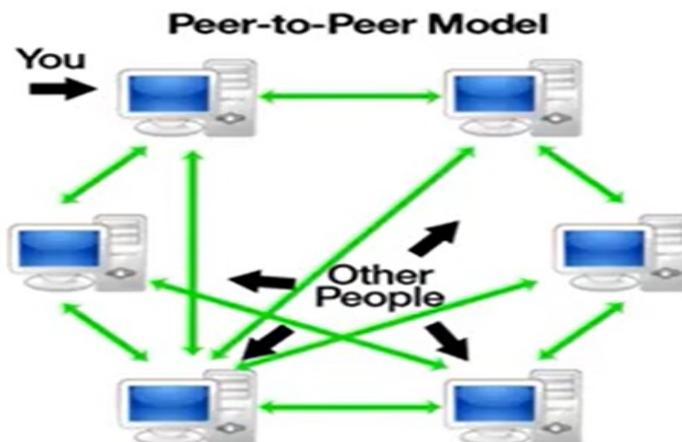
- Mạng lưu trữ ngang hàng là một cụm các máy tính được xây dựng trên một nền mạng máy tính tồn tại, chia sẻ tất cả các lưu trữ sẵn có trên mạng
- Các giải pháp lưu trữ thường dùng DAS (Direct Attached Storage) NAS (Network Attached Storage) SAN (Storage Area Networks):



Ứng dụng của mạng P2P

■ Các chương trình xử lý:

- Trong một mạng mỗi máy tính là trong suốt với các máy tính khác và tất cả các node được kết nối mạng sẽ tạo thành một máy tính logic.
- Sử dụng các ứng dụng mạng ngang hàng để bó cụm các chương trình xử lý có thể nhận được khả năng tính toán ngang bằng với một siêu máy tính đắt tiền.



So sánh P2P và Client-server

Chức năng

P2P

- Cho phép các node (PCs) đóng góp chia sẻ tài nguyên với nhau
- Tài nguyên chia sẻ của các node: ổ đĩa cứng, máy in, thư mục, file...
- Các nguồn tài nguyên này được truy cập từ bất cứ node nào trong mạng
- Các node đóng vai trò như cả Client (truy vấn thông tin) và Server (Cung cấp thông tin)

Client/Server

- Dữ liệu được lưu trữ ở một Server trung tâm, tốc độ cao (Tốc độ truy cập thường lớn hơn so với mạng P2P)
- Khi một máy Client yêu cầu lấy thông tin về thời gian nó sẽ phải gửi một yêu cầu theo một tiêu chuẩn do server định ra, nếu yêu cầu được chấp nhận thì máy Server sẽ trả về thông tin mà Client yêu cầu

So sánh P2P và Client-server

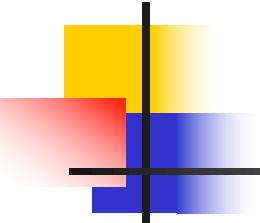
Ưu điểm

| P2P | Client/Server |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">Không cần Server riêng, các Client chia sẻ tài nguyên. Khi mạng càng mở rộng thì khả năng hoạt động của hệ thống càng tốtRẻDễ cài đặt và bảo trìThuận lợi cho việc chia sẻ file, máy in, ổ đĩa... | <ul style="list-style-type: none">Tốc độ truy cập nhanhKhả năng mở rộng caoHoạt động với bất kỳ loại ứng dụng nàoSử dụng được với các ứng dụng chia sẻ CSDLĐáng tin cậy hơn (có Server riêng)Mức độ an toàn cao nhất |

So sánh P2P và Client-server

Nhược điểm

| P2P | Client/Server |
|------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">▪ Chậm▪ Không tốt cho các ứng dụng CSDL▪ Kém tin cậy | <ul style="list-style-type: none">• Cần Server riêng• Đắt• Phức tạp trong việc bảo trì, duy trì hoạt động của mạng |



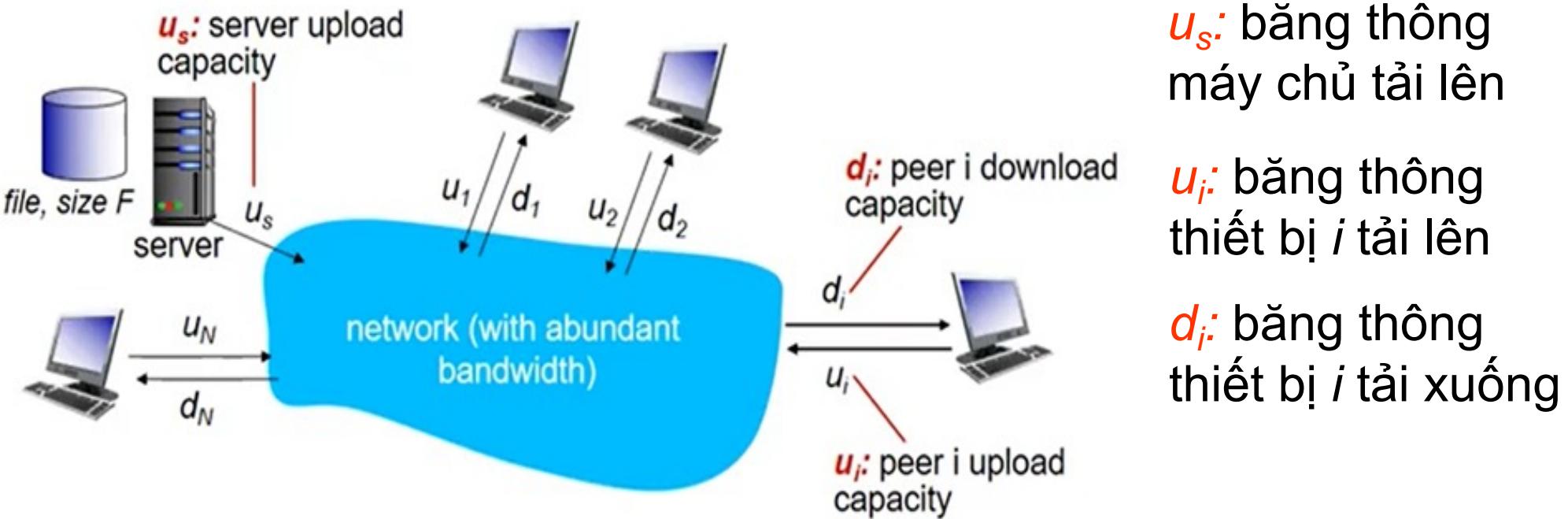
Phân bố tệp P2P

- Phân bố tệp rất lớn từ một trạm chủ đơn tới một số các trạm chủ (gọi là các thiết bị ngang hàng).
- **Kiến trúc phân bố tệp client-server:** máy chủ phải gửi bản sao của tệp tới từng thiết bị, áp đặt một tải trọng khổng lồ lên máy chủ và tiêu tốn lượng băng thông máy chủ rất lớn.
- **Kiến trúc phân bố tệp P2P:** mỗi thiết bị ngang hàng có thể phân bố lại bất cứ lượng tệp nào mà nó nhận được tới bất cứ thiết bị ngang hàng khác, do đó hỗ trợ máy chủ trong quá trình phân bố.
 - Giao thức phân bố tệp P2P thông dụng nhất là **BitTorrent**.

Phân bô tệp P2P

Bài toán: Mất bao nhiêu thời gian để phân bô tệp từ một máy chủ đến N thiết bị ?

Thời gian phân bô là thời gian thực hiện lấy được bản sao của tệp đối với tất cả N thiết bị ngang hàng.



u_s : băng thông máy chủ tải lên

u_i : băng thông thiết bị i tải lên

d_i : băng thông thiết bị i tải xuống

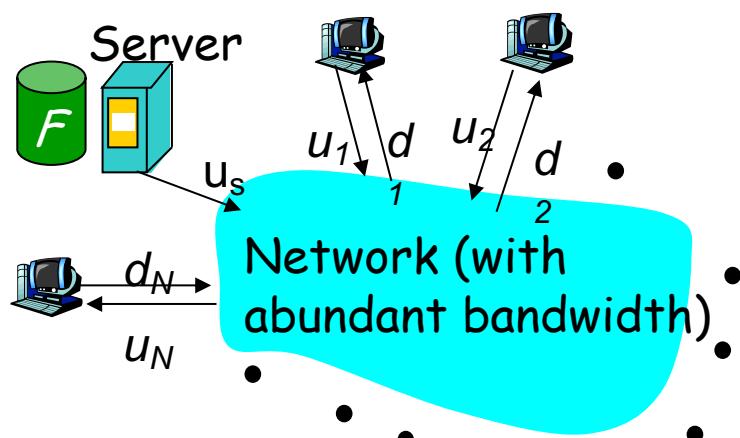
Phân bổ tệp P2P

Thời gian phân bổ tệp của kiến trúc client-server

- Thời gian máy chủ lặp lượt gửi N bản sao tới N máy khách: NF/u_s
- Thời gian máy khách i cần để tải xuống 1 file: F/d_i

Thời gian để phân bổ
tệp F đến N máy
khách sử dụng kiến
trúc **client-server**

$$= d_{cs} = \max \{ NF/u_s, F/\min(d_i) \}$$



Tăng tuyến tính với N (N lớn)

Phân bô tệp P2P

Thời gian phân bô tệp của kiến trúc P2P

- Ban đầu, chỉ server có file. Server phải gửi file tới ít nhất 1 máy khách. Thời gian máy chủ phải gửi ít nhất 1 bản sao: F/u_s
- Thời gian máy khách i cần để tải xuống 1 file: F/d_i
- Cuối cùng, quan sát tổng dung lượng upload của hệ thống: $u_s + \sum u_i$. Hệ thống phải phân phõi F bit tới N peer, do đó phải phân phõi tổng NF bits. Do đó, thời gian phân phõi: $NF / (u_s + \sum u_i)$.

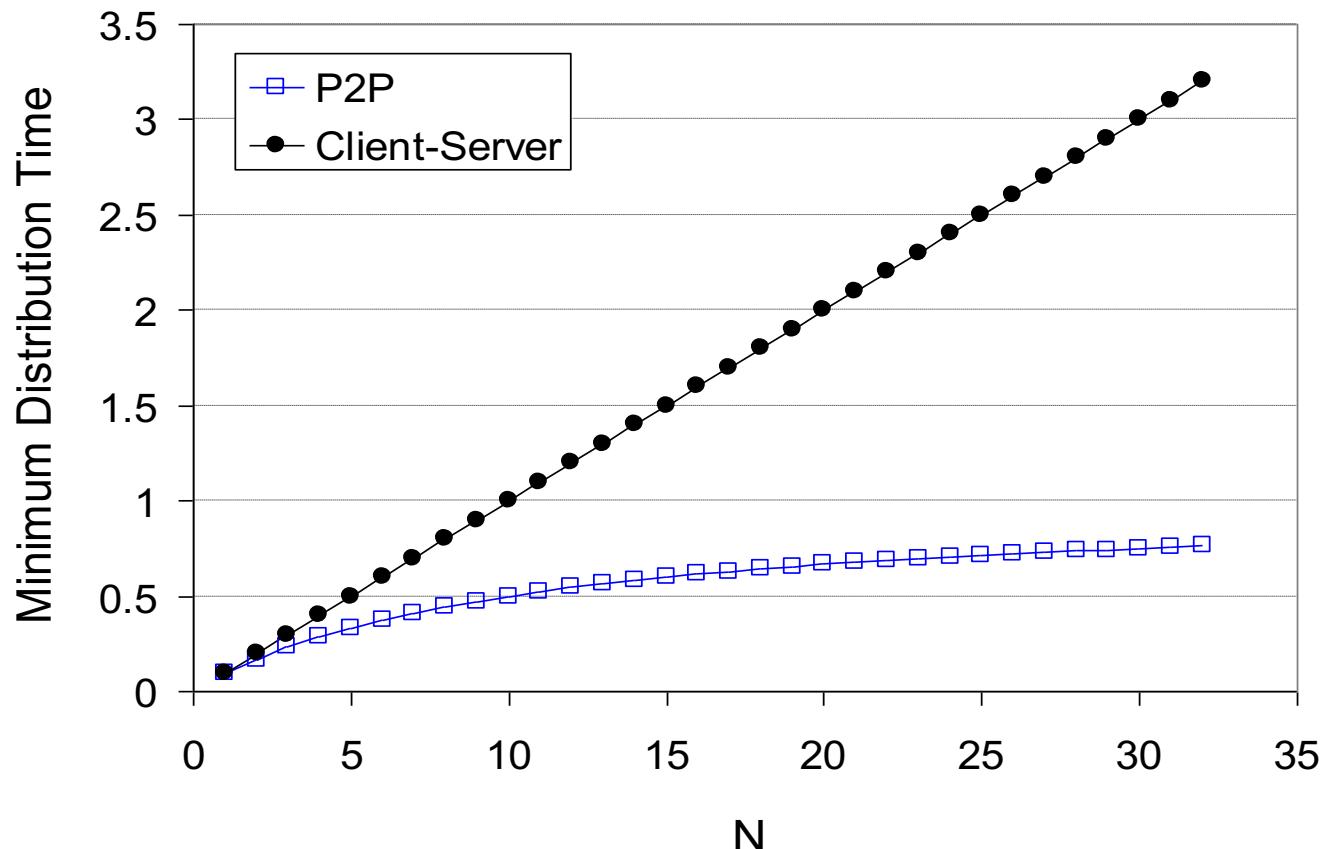
Thời gian để phân bô tệp F đến N máy khách sử dụng kiến trúc P2P:

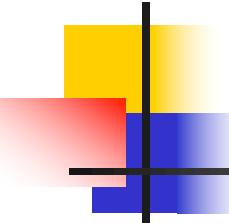
$$d_{P2P} = \max \{ F/u_s, F/\min(d_i), NF/(u_s + \sum u_i) \}$$

Phân bố tệp P2P

Ví dụ so sánh kiến trúc client-server và P2P

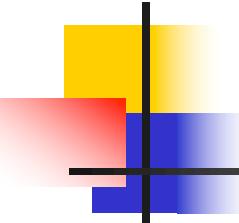
Tốc độ tải lên máy khách = u , $F/u = 1h$, $u_s = 10u$, $d_{min} \geq u_s$





Bài tập: Phân bổ tệp P2P

- Bài tập 1: Consider distributing a file of $F = 15$ Gbits to N peers. The server has an upload rate of $u_s = 30$ Mbps, and each peer has a download rate of $d_i = 2$ Mbps and an upload rate of u . For $N = 10, 100$, and $1,000$ and $u = 300$ Kbps, 700 Kbps, and 2 Mbps, prepare a chart giving the minimum distribution time for each of the combinations of N and u for both client-server distribution and P2P distribution.



Bài tập: Phân bổ tệp P2P

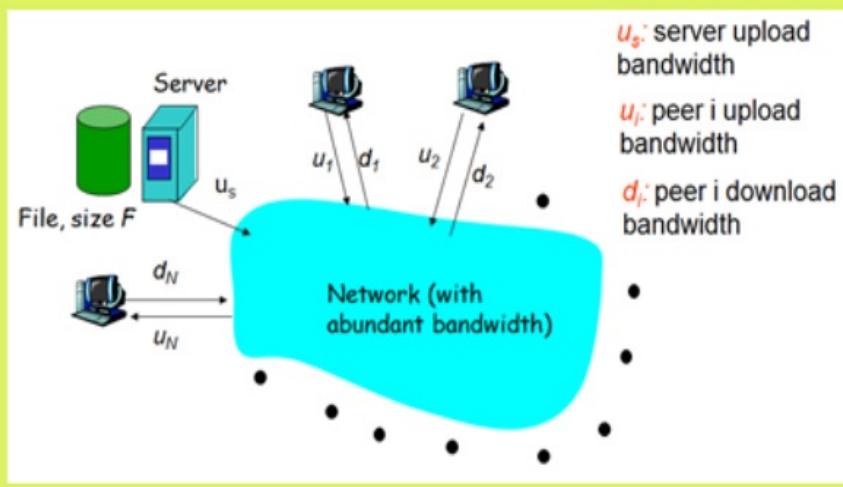
■ Bài 2:

Consider distributing a file of F bits to N peers using a client-server architecture. Assume a fluid model where the server can simultaneously transmit to multiple peers, transmitting to each peer at different rates, as long as the combined rate does not exceed u_s .

- Suppose that $u_s/N \leq d_{\min}$. Specify a distribution scheme that has a distribution time of NF/u_s .
- Suppose that $u_s/N \geq d_{\min}$. Specify a distribution scheme that has a distribution time of F/d_{\min} .
- Conclude that the minimum distribution time is in general given by $\max\{NF/u_s, F/d_{\min}\}$.

Bài tập: Phân bổ tệp P2P

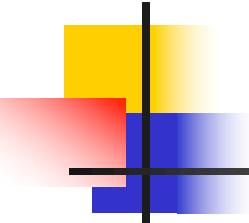
Bài 3: Phân tích và tính thời gian để toàn bộ 20 máy tính tải được bộ phim có dung lượng là 300MB khi truyền theo kiến trúc Client-Server (CS) và peer-to-peer (P2P) với $u_s=d_{min}=50$ Mb/s.



Phân bô tệp P2P

Giao thức BitTorrent

- BitTorrent là giao thức P2P thông dụng để phân bô tệp
- Tập hợp tất cả các thiết bị ngang hàng tham gia phân bô một tệp cụ thể được gọi là *torrent* (dòng chảy)
 - Các thiết bị trong torrent tải xuống các khúc dữ liệu kích cỡ bằng nhau của tệp từ một thiết bị khác
 - Tệp được chia thành các khúc (chunk) 256KB
 - Khi thiết bị ngang hàng tham gia vào torrent lần đầu, nó không có khúc dữ liệu nào
- Trong khi tải xuống, thiết bị ngang hàng tải lên các khúc dữ liệu cho thiết bị ngang hàng khác.
- Các thiết bị ngang hàng có thể vào và rời bỏ torrent
- Một khi đã có toàn bộ tệp, thiết bị ngang hàng có thể rời bỏ hoặc ở lại.



Phân bố tệp P2P

Giao thức BitTorrent

- Mỗi torrent có một nút hạ tầng gọi là bộ theo dõi (tracker).
- Khi thiết bị ngang hàng gia nhập torrent:
 - Nó tự đăng ký với bộ theo dõi
 - Định kì thông báo với bộ theo dõi nó vẫn còn trong torrent
- Bộ theo dõi duy trì giám sát các thiết bị ngang hàng đang tham gia vào torrent
- Một torrent cho trước có thể có ít hơn 10 hay nhiều hơn 1000 thiết bị ngang hàng tại bất cứ thời điểm nào.

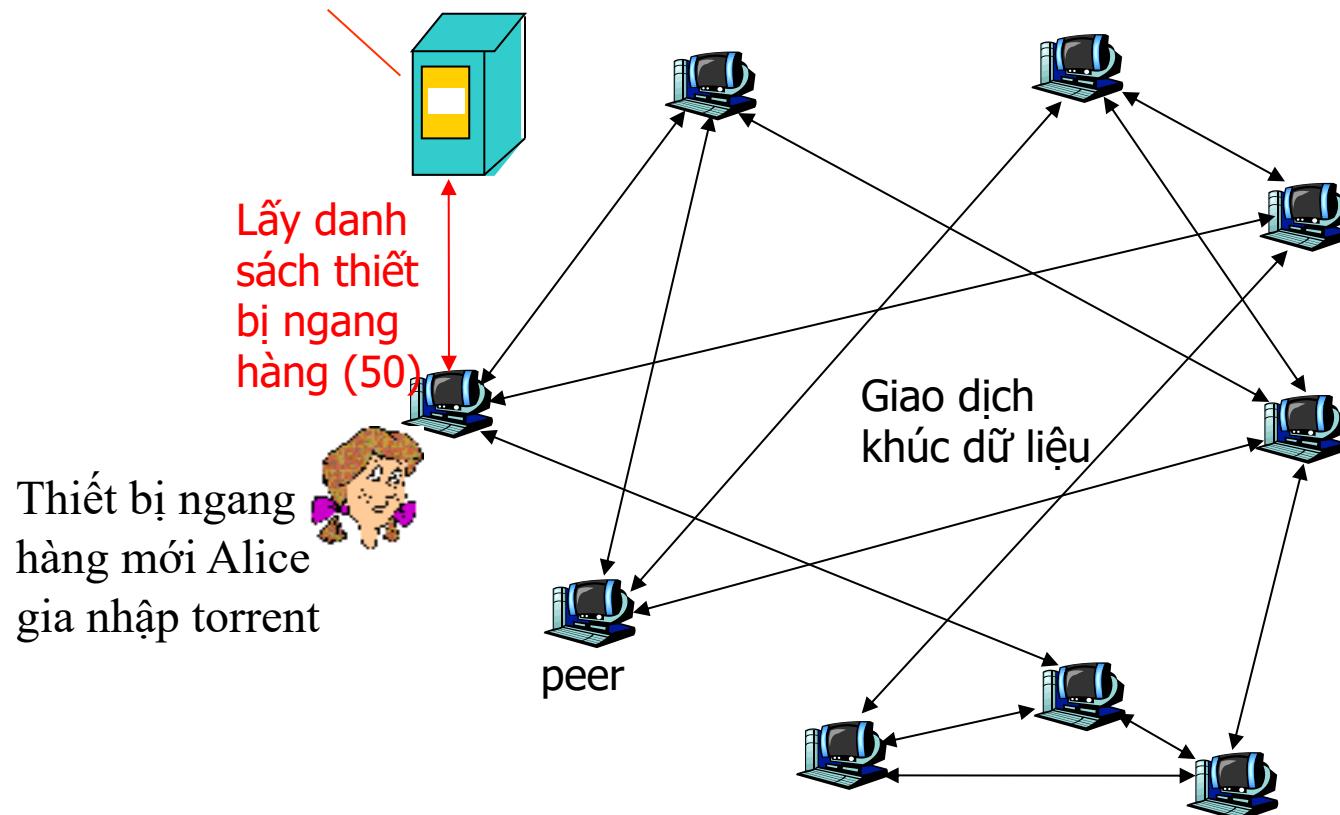
Phân bô tệp P2P

Giao thức BitTorrent

Bộ theo dõi (tracker):

Theo dõi các thiết bị ngang hàng tham gia torrent

torrent: nhóm các thiết bị ngang hàng trao đổi khúc dữ liệu tệp



Phân bố tệp P2P

Giao thức BitTorrent

Lấy khúc dữ liệu

- Tại bất cứ thời điểm nào, các thiết bị ngang hàng có tập nhỏ khác nhau các khúc dữ liệu của tệp
- Định kì theo thời gian, thiết bị ngang hàng hỏi hàng xóm của nó về danh sách khúc dữ liệu mà chúng có.
- Thiết bị ngang hàng gửi yêu cầu về các khúc dữ liệu mà nó chưa có
 - Hiếm nhất đầu tiên (rarest first).

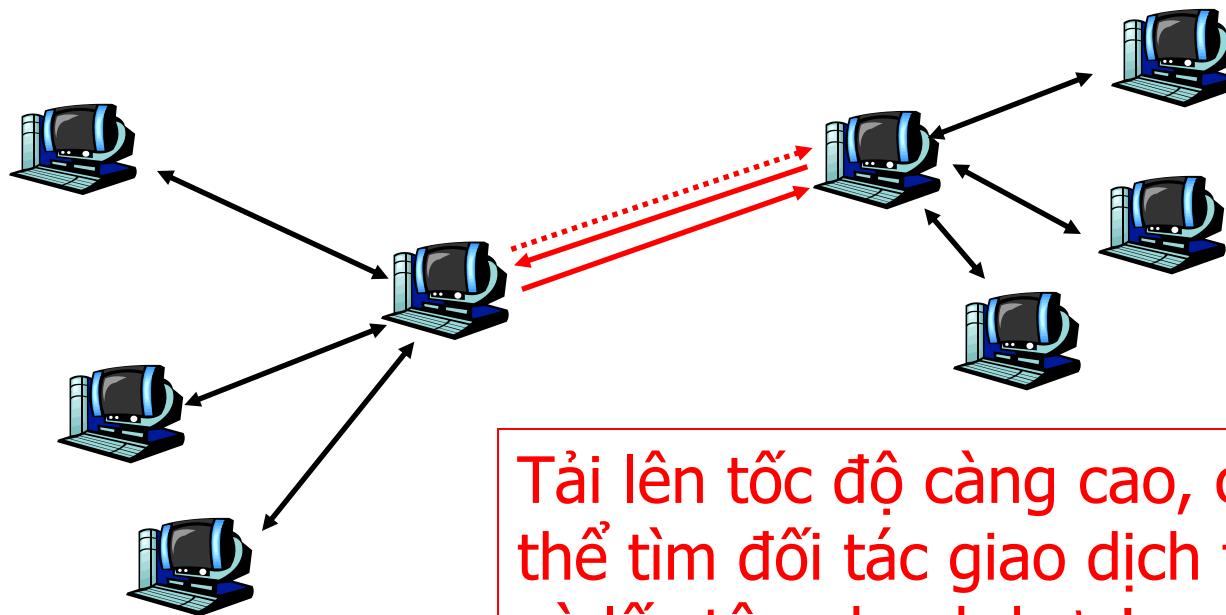
Gửi khúc dữ liệu: ăn miếng trả miếng

- Thiết bị ngang hàng gửi khúc dữ liệu cho 4 hàng xóm hiện đang gửi dữ liệu cho nó tại tốc độ lớn nhất
 - ❖ Cứ 10s lại đo và đánh giá lại 4 thiết bị đứng đầu này (mở).
- Mỗi 30s: lựa chọn ngẫu nhiên một thiết bị ngang hàng khác, bắt đầu gửi khúc dữ liệu
 - ❖ Thiết bị ngang hàng mới này có thể gia nhập nhóm 4 thiết bị đứng đầu
 - ❖ Mở tối ưu

Phân bô tệp P2P

Giao thức BitTorrent

- (1) A mở tối ưu cho B.
- (2) A trở thành một trong 4 nhà cung cấp hàng đầu của B. B đáp lại.
- (3) B trở thành một trong 4 nhà cung cấp hàng đầu của A.



Tải lên tốc độ càng cao, càng có
thể tìm đối tác giao dịch tốt hơn
và lấy tệp nhanh hơn!

Bảng hàng băm DHT

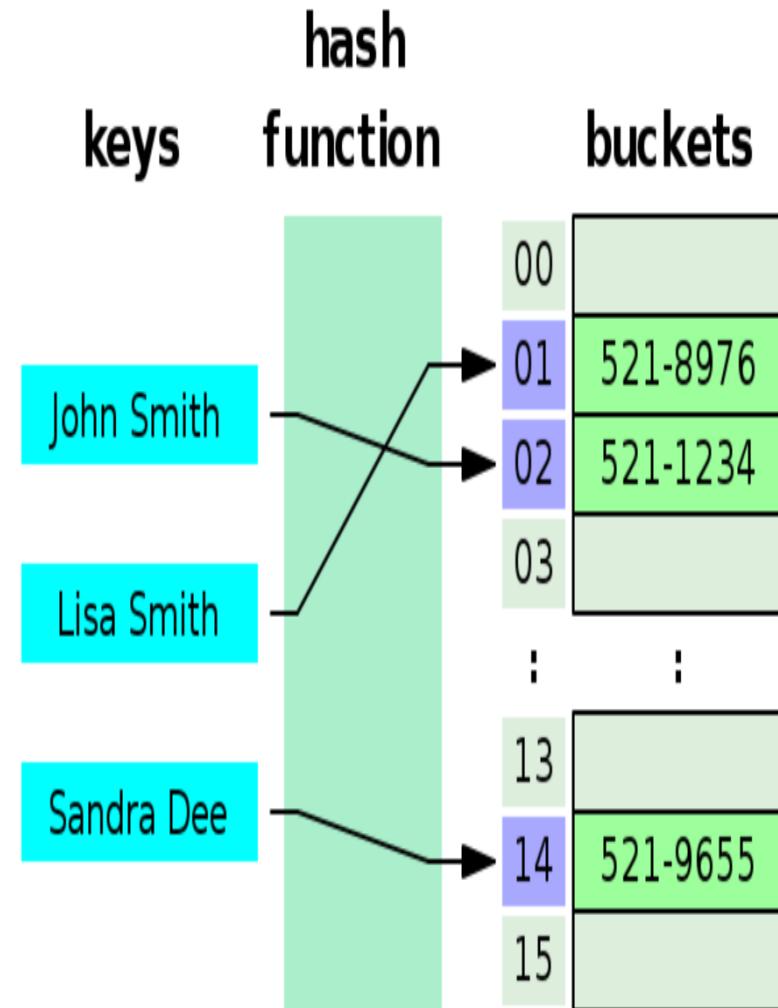
Bảng băm

- Một bảng băm (hash table) là một cấu trúc dữ liệu ánh xạ giữa khóa và giá trị, tức là tương ứng với một khóa bảng băm sẽ trả về một giá trị.
- Ví dụ về việc sử dụng bảng băm:
 - Trong trường đại học, mỗi sinh viên được chỉ định một mã sinh viên không giống nhau và qua mã sinh viên đó có thể truy xuất các thông tin của sinh viên đó.
 - Trong thư viện, mỗi một cuốn sách một mã số riêng và mã số đó có thể được dùng để xác định các thông tin của sách.
- Trong cả 2 ví dụ trên, các sinh viên và các cuốn sách được “băm” thành các mã số duy nhất (không trùng lặp)

Bảng hàng băm DHT

Bảng băm

- Giả sử có một đối tượng và muốn gán cho nó một cái khóa (key) để giúp tìm kiếm dễ dàng hơn.
- Để lưu giữ cặp <khóa, giá trị>(<key, value>), có thể sử dụng mảng bình thường để làm việc này;
 - Với chỉ số mảng là khóa
 - Giá trị tại chỉ số đó là giá trị tương ứng của khóa.
- Tuy nhiên, trong trường hợp phạm vi của khóa lớn và không thể sử dụng chỉ số mảng được, khi đó sẽ cần tới “băm”(hashing).



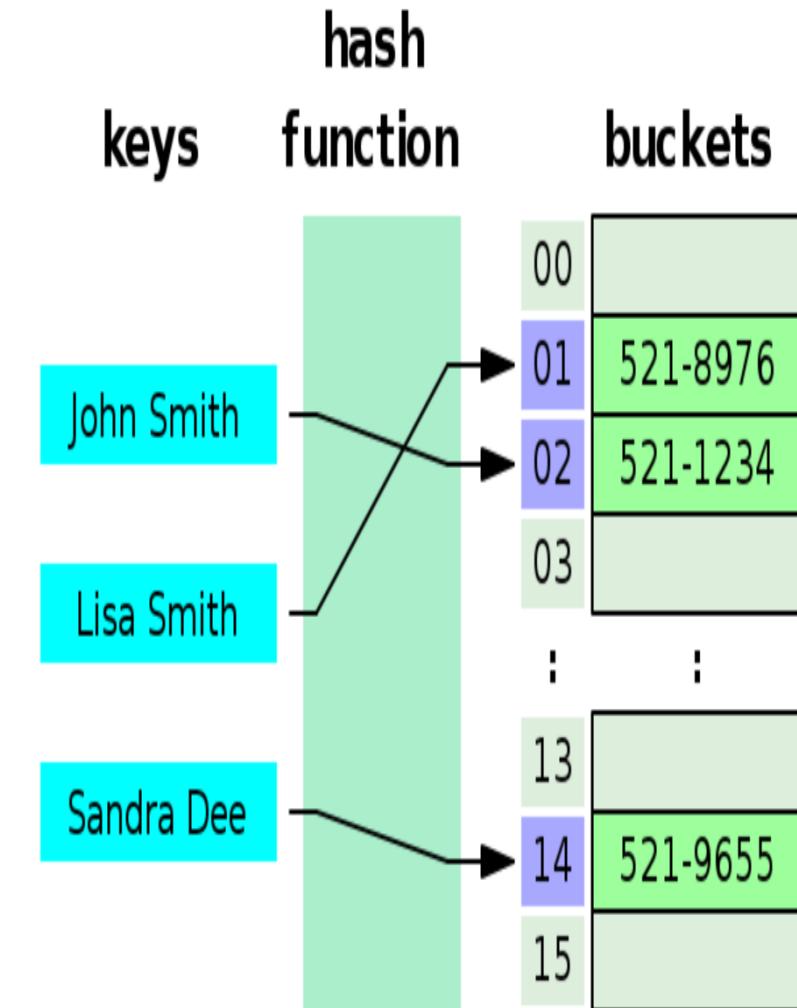
Bảng hàng băm DHT

Bảng băm

□ Trong hashing, các key có giá trị lớn sẽ được đưa về giá trị nhỏ hơn bằng cách sử dụng **hàm băm (hash functions)**.

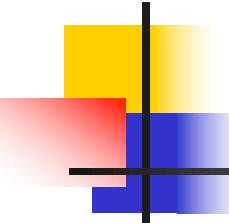
□ Các giá trị sau đó được lưu trong một cấu trúc dữ liệu gọi là **bảng băm (hash table)**

=> Có va chạm xảy ra



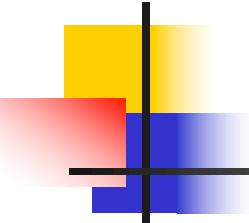
Bảng hàng băm phân tán DHT

- Là một lớp các hệ thống phân tán, cung cấp một dịch vụ tra cứu tương tự như một bảng băm: các cặp (khóa, giá trị) được lưu trữ trong DHT
- Bất kỳ nút mạng tham gia nào cũng có thể lấy được giá trị liên kết với một khóa cho trước một cách hiệu quả.
- Nhiệm vụ lưu trữ ánh xạ từ khóa tới giá trị được phân tán giữa các nút, bằng cách đó sẽ giảm bớt lỗi nếu có thay đổi trong một tập hợp các nút tham gia.
- Điều này cho phép sử dụng DHT cho một số lượng cực lớn các nút mạng và xử lý việc vào, ra, và lỗi các nút mạng một cách liên tục.



Bảng hàng băm phân tán DHT

- Thành phần quan trọng nhất trong P2P là chỉ số (cơ sở dữ liệu đơn giản) hỗ trợ tìm kiếm và cập nhật trong hệ thống phân tán
- Khi các chỉ số này được phân tán, các thiết bị ngang hàng có thể tạo lập lưu đệm nội dung và định tuyến thông minh các truy vấn giữa chúng với nhau.
- Đánh chỉ số và tìm kiếm thông tin là thành phần quan trọng trong các hệ thống P2P
- Kỹ thuật đánh chỉ số và tìm kiếm thông dụng: Bảng hàm băm phân tán DHT.



Bảng hàng băm DHT

- ❖ Chỉ số (cơ sở dữ liệu đơn giản) hỗ trợ tìm kiếm trong hệ thống phân tán
- ❖ Chỉ số: (khóa-key, giá trị-value)
(123-456-789, Nguyen)
(God Father, 203.17.123.38)
- ❖ Thiết bị ngang hàng truy vấn cơ sở dữ liệu bằng cách cung cấp khóa: nếu có cặp (khóa, giá trị) trong cơ sở dữ liệu phù hợp với khóa, cơ sở dữ liệu trả lại cặp phù hợp cho thiết bị ngang hàng truy vấn.

Bảng hàng băm DHT

P2P: Tìm kiếm thông tin - Ứng dụng

Chia sẻ tệp (ví dụ, e-mule)

- Chỉ số theo dõi động vị trí của các tệp mà các thiết bị ngang hàng chia sẻ.
- Thiết bị ngang hàng cần thông báo chỉ số về những gì nó lưu trữ.
- Thiết bị ngang hàng tìm kiếm chỉ số để xác định các tệp nó cần có thể tìm ở đâu.

Nhắn tin tức thời IM (Instant messaging)

- Chỉ số ánh xạ tên người sử dụng với vị trí
- Khi người sử dụng bắt đầu ứng dụng IM, nó cần thông báo chỉ số của vị trí của nó.
- Các thiết bị ngang hàng tìm chỉ số để xác định địa chỉ IP của người sử dụng.

Bảng hàng băm DHT

P2P: Phương án tập trung

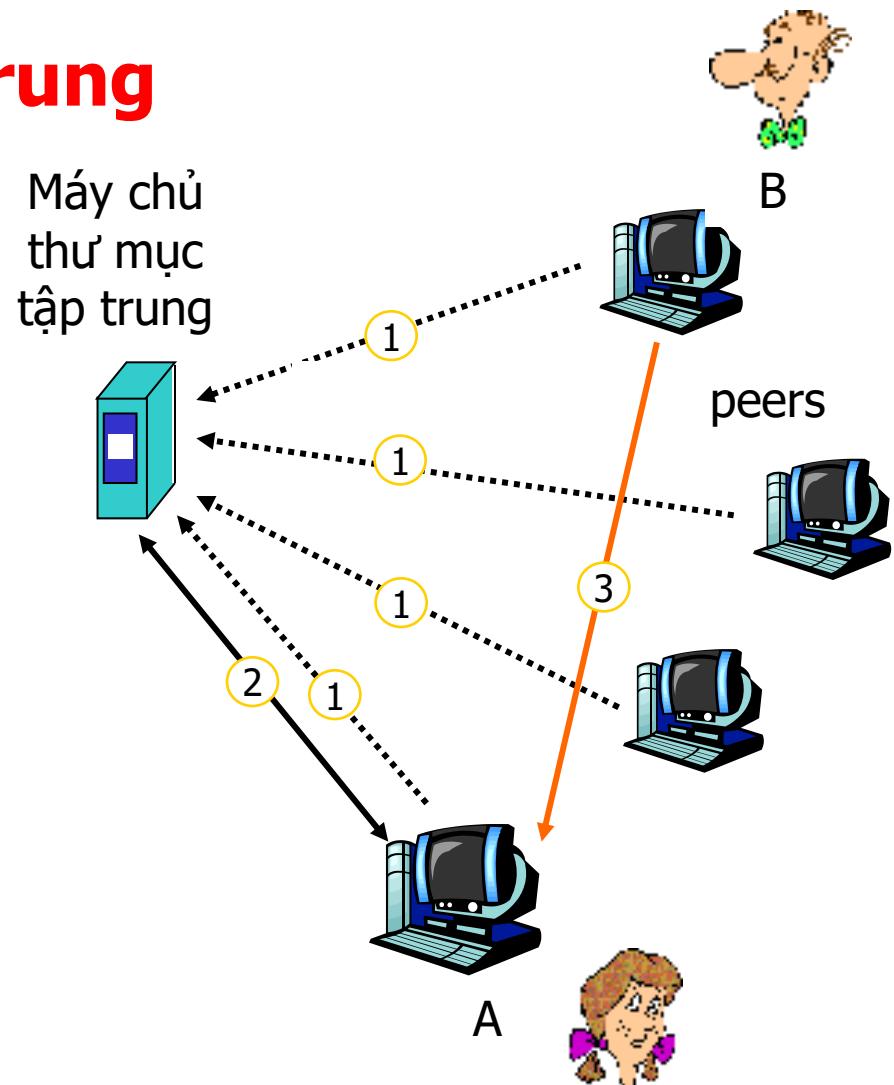
Thiết kế tập trung “Napster”

1) Khi thiết bị ngang hàng kết nối, nó thông báo cho máy chủ trung tâm:

- Địa chỉ IP
- Nội dung

2) A truy vấn “God Father”

3) A yêu cầu tệp từ B



Bảng hàng băm DHT

P2P: hệ thống phân tán

- Hệ thống phân tán bao gồm từ hàng triệu thiết bị ngang hàng kết nối với nhau không có ủy quyền trung tâm.
 - Cần phân tán cặp (khóa, giá trị) thông qua tất cả các thiết bị ngang hàng, sao cho mỗi thiết bị ngang hàng chỉ giữ một tập nhỏ của tổng số các cặp (khóa, giá trị)
 - thiết bị ngang hàng truy vấn có thể gửi truy vấn tới tất cả các thiết bị ngang hàng khác
 - thiết bị ngang hàng chứa cặp (khóa, giá trị) phù hợp với khóa có thể đáp ứng với cặp phù hợp của chúng
 - Yêu cầu mỗi thiết bị ngang hàng phải theo dõi tất cả các thiết bị ngang hàng khác
- => hoàn toàn không thể thực hiện được với mạng có quy mô lớn

Bảng hàng băm DHT

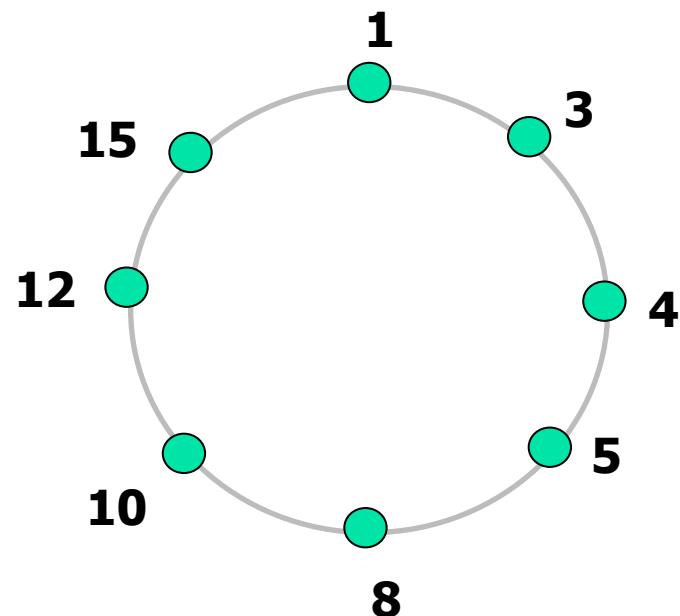
P2P: sử dụng DHT

- ❖ Gán một định danh cho mỗi thiết bị ngang hàng, mỗi định danh là một số nguyên trong dải $[0, 2^n - 1]$ với một số n cố định
- ❖ Yêu cầu mỗi khóa là một số nguyên trong dải này, để tạo các số nguyên cho các khóa này, hàm băm sẽ được sử dụng
- ❖ Vấn đề lưu trữ cặp (khóa, giá trị) trong DHT
 - mỗi thiết bị ngang hàng có một định danh số nguyên và mỗi khóa cũng là một số nguyên trên cùng một dải,
 - giải pháp tự nhiên là gán mỗi cặp (khóa, giá trị) tới thiết bị ngang hàng có định danh gần khóa nhất (ngay sau khóa)
 - Giả sử $n = 4$ sao cho tất cả thiết bị ngang hàng và định danh khóa trong dải $[0, 15]$. Tiếp tục giả sử có tám thiết bị ngang hàng trong hệ thống có định danh 1, 3, 4, 8, 10, 12, và 15. Cuối cùng, giả sử chúng ta muốn lưu trữ cặp khóa-giá trị (11, Nguyen) vào trong một từ tám thiết bị ngang hàng. Nhưng trong thiết bị ngang hàng nào? Sử dụng quy ước “gần nhất”, do thiết bị ngang hàng 12 là thiết bị ngay sau đối với khóa 11, do đó chúng ta lưu trữ cặp (11, Nguyen) trong thiết bị ngang hàng 12.
- ❖ Yêu cầu mỗi thiết bị ngang hàng phải duy trì theo dõi tất cả các thiết bị ngang hàng khác trong DHT => điều này hoàn toàn không thực tế đối với hệ thống qui mô lớn với hàng triệu thiết bị ngang hàng

Bảng hàng băm DHT

DHT vòng

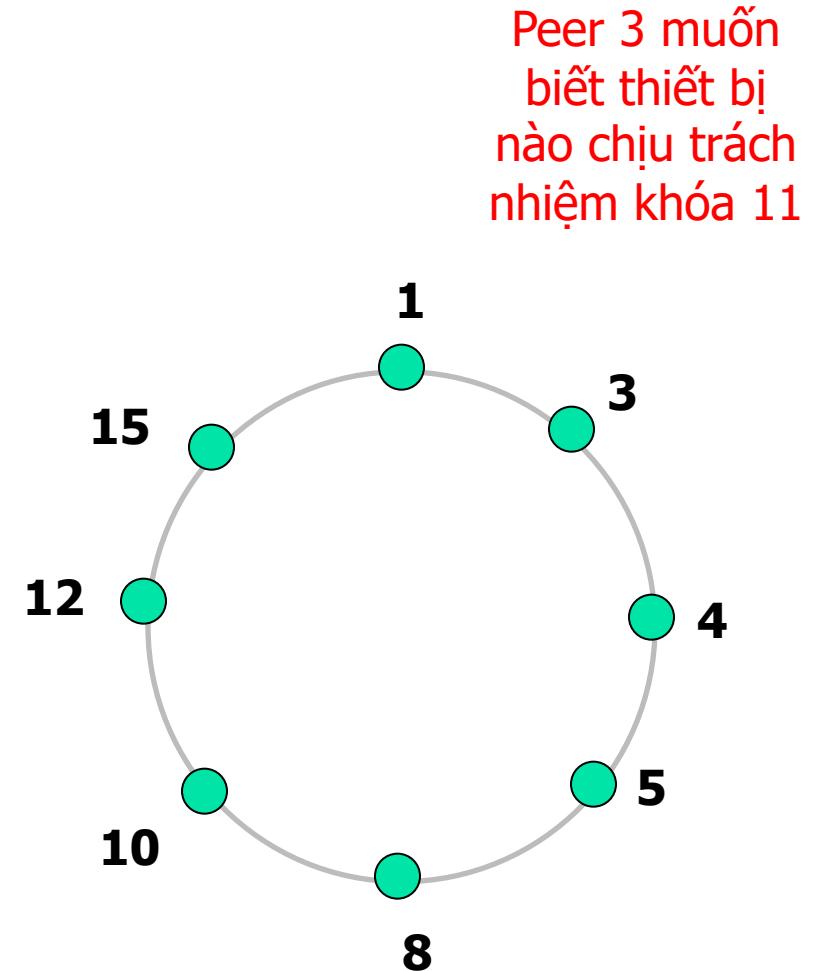
- Để giả quyết vấn đề quy mô, xem xét tổ chức các thiết bị ngang hàng thành một vòng
- Mỗi thiết bị ngang hàng chỉ duy trì theo dõi thiết bị ngay sau nó ($\text{modulo } 2^n$).
- Ví dụ: $n = 4$ và cũng có 8 thiết bị ngang hàng như ví dụ trước. Thiết bị ngang hàng 5 biết địa chỉ IP và định danh của thiết bị ngang hàng 8 nhưng không cần thiết phải biết về các thiết bị ngang hàng khác có thể có trong DHT
 - ❖ Việc bố trí vòng các thiết bị ngang hàng như vậy là trường hợp đặc biệt của *mạng chồng phủ* (overlay network)
 - ❖ Trong mạng chồng phủ, các thiết bị ngang hàng tạo thành mạng logic trùu tượng lưu trú bên trên mạng máy tính “nền” bao gồm từ các liên kết vật lý, bộ định tuyến, và máy chủ.
 - ❖ Các liên kết trong mạng che phủ không phải là các liên kết vật lý, mà chỉ đơn giản là các liên kết ảo giữa các cặp thiết bị ngang hàng.



Bảng hàng băm DHT

DHT vòng

- Mỗi peer chỉ biết thiết bị ngay sau nó.
- Peer 3 muốn biết peer nào chịu trách nhiệm khóa 11: Peer 3 gửi bản tin đến peer 4; peer 4 gửi bản tin đến peer 5; ... cho đến peer 12 chịu trách nhiệm khóa 11. Peer 12 gửi bản tin trả lời đến peer 3.
- Giảm số thông tin mỗi peer phải quản lý.
- Số lượng bản tin gửi đi lớn ($N/2$).

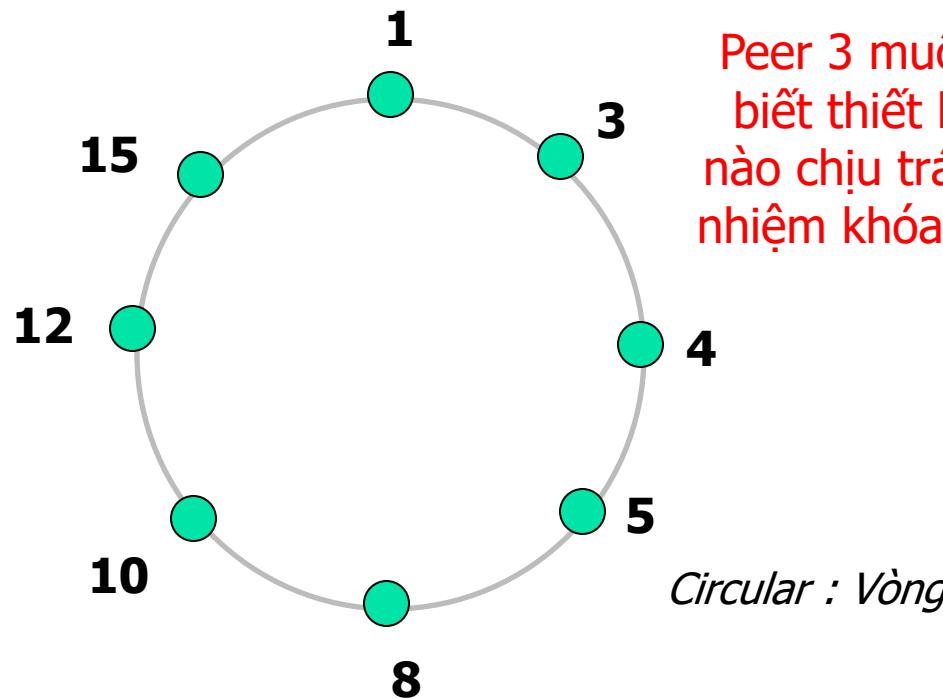


Bảng hàng băm DHT

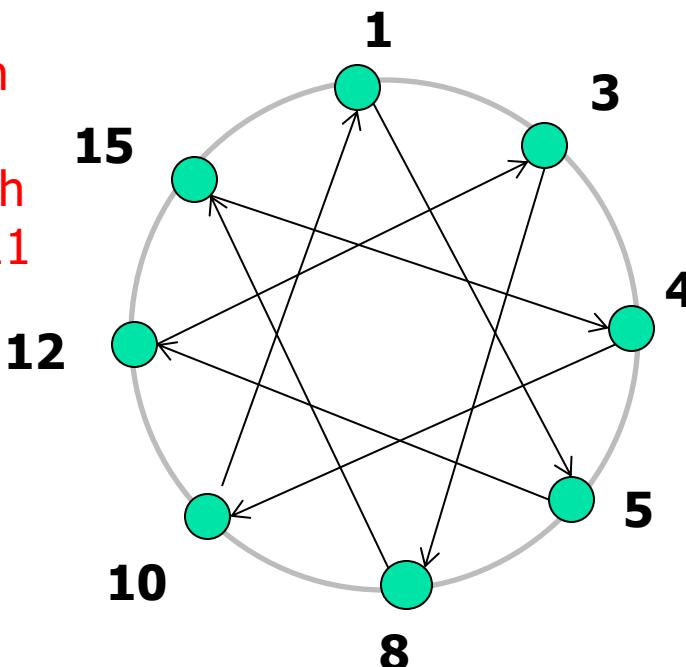
DHT vòng (1)

Mạng che phủ: biểu đồ

- Cạnh giữa thiết bị ngang hàng X và Y là kết nối TCP
- Tất cả các thiết bị ngang hàng đang hoạt động và các cạnh tạo thành mạng che phủ
- Cạnh: liên kết ảo (*không phải kết nối vật lý thực tế*)



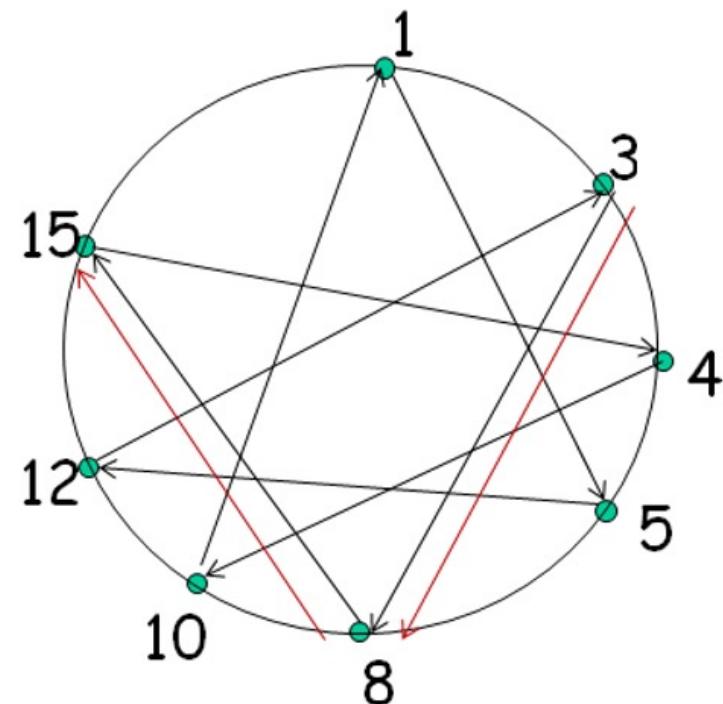
Peer 3 muốn
biết thiết bị
nào chịu trách
nhiệm khóa 11



Bảng hàng băm DHT

DHT vòng với các đường tắt

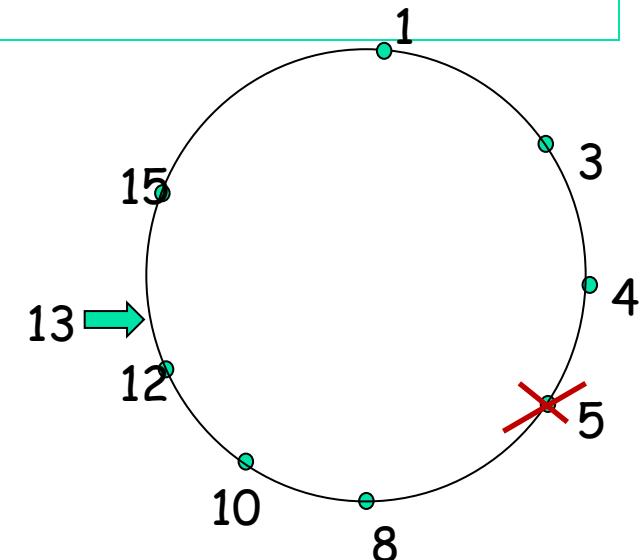
- Tìm điểm cân bằng giữa số lượng thông tin mỗi peer phải theo dõi và số lượng bản tin truyền trên mạng: bổ sung các đường kết nối tắt.
- Mỗi thiết bị ngang hàng không chỉ theo dõi thiết bị ngay sau của nó, mà cả một số lượng nhỏ các thiết bị ngang hàng được nối tắt rải rác trên vòng tròn
- Các đường tắt được sử dụng để tiến hành định tuyến các bản tin truy vấn.
- Peer 3 – peer 4 – peer 10 – peer 12.
=> các đường nối tắt có thể giảm đáng kể số lượng bản tin sử dụng để xử lý truy vấn



Bảng hàng băm DHT

Peer churn

- P2P: thiết bị ngang hàng có thể vào và ra không báo trước. Do đó, khi thiết kế DHT, chúng ta phải quan tâm đến duy trì che phủ DHT bằng sự có mặt của *peer churn*
- Để xử lý peer churn, chúng ta sẽ yêu cầu mỗi thiết bị ngang hàng theo dõi (tức là biết địa chỉ IP) của thiết bị ngay sau thứ nhất và thứ hai của nó
 - ví dụ, thiết bị ngang hàng 4 bấy giờ theo dõi cả thiết bị ngang hàng 5 và 8.
- Yêu cầu mỗi thiết bị ngang hàng định kì kiểm tra hai thiết bị ngay sau của nó còn sống (ping)



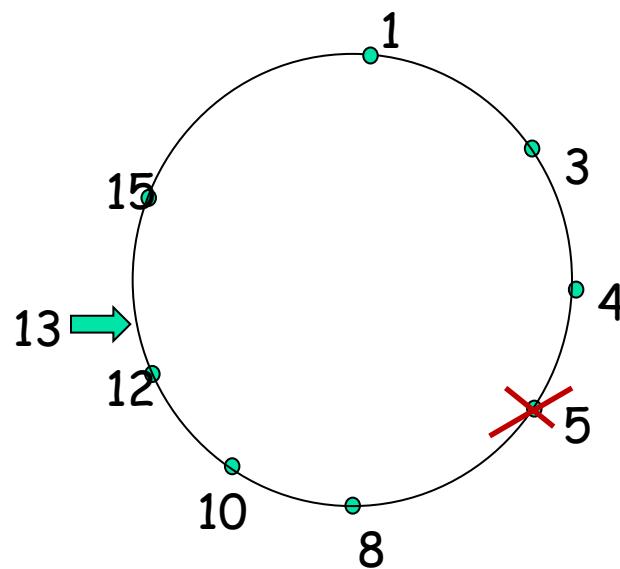
Bảng hàng băm DHT

Peer churn

- DHT được duy trì như thế nào khi một thiết bị ngang hàng đột ngột rời bỏ/gia nhập

Peer rời bỏ (peer 5):

- 2 thiết bị ngang hàng đằng trước thiết bị ngang hàng bỏ đi (4 và 3) biết được 5 đã bỏ đi (ping không phản hồi)
- Thiết bị 4 và 3 vì vậy cần cập nhật thông tin thiết bị ngay sau của chúng.
- Thiết bị ngang hàng 4 sau đó yêu cầu định danh và địa chỉ IP của thiết bị ngay sau thứ hai của nó (thiết bị ngang hàng 10).



Peer gia nhập (peer 13):

- Chỉ biết peer 1.
- Peer 1 gửi bản tin đến peer 13 qua các peer “Thiết bị ngay trước và sau của peer 13 là ai?”;
- Peer 12 nhận được bản tin và biết được nó là peer ngay trước của peer 13 và peer 15 là peer ngay sau của nó.

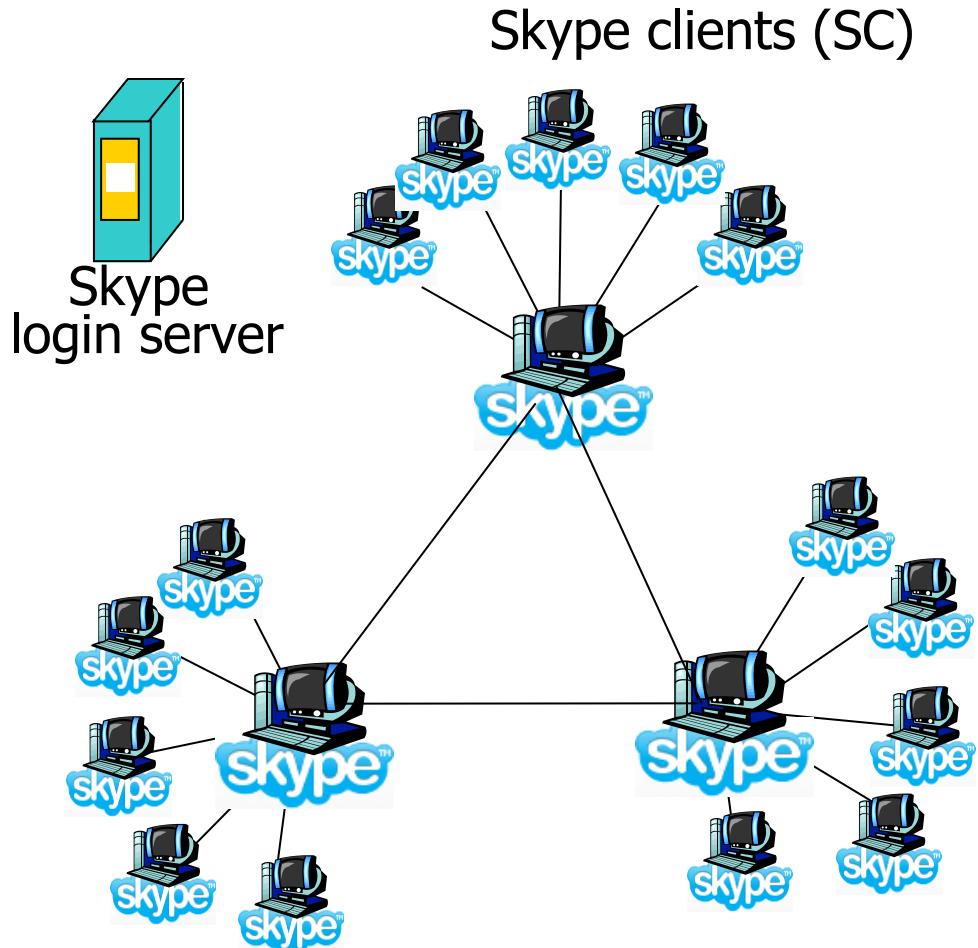
Bảng hàng băm DHT

Ứng dụng DHT vào BitTorrent

- BitTorrent sử dụng Kademlia DHT để tạo lập bộ theo dõi phân tán.
- Khóa-key: định danh của torrent;
- Giá trị-value: địa chỉ IP của tất cả các peer đang tham gia trong torrent.
- Peer mới gia nhập: truy vấn định danh torrent, xác định peer chịu trách nhiệm theo dõi.

Thoại Internet P2P Skype

- Bản chất P2P: các cặp người sử dụng kết nối với nhau.
- Giao thức lớp ứng dụng độc quyền
- Mạng che phủ phân cấp với các siêu nút SN
- Chỉ số ánh xạ tên người sử dụng với địa chỉ IP; được phân tán trên các SN.



Thoại Internet P2P Skype

Sử dụng peer như trạm chuyển tiếp

- Khi A và B đều ở đằng sau “NATs”.
 - NAT ngăn chặn peer bên ngoài khởi tạo cuộc gọi vào peer bên trong
- Giải pháp:
 - Sử dụng các siêu nút SN của A và B để chuyển tiếp
 - Từng peer khởi tạo phiên với trạm chuyển tiếp.
 - Các Peers có thể truyền thông qua NAT sử dụng chuyển

