

TRƯỜNG ĐẠI HỌC THỦY LỢI  
**KHOA CÔNG NGHỆ THÔNG TIN**

# **MẠNG MÁY TÍNH**

## **Chương 2. Tầng ứng dụng**

ThS. Nguyễn Thị Phương Thảo  
Email: [thaont@tlu.edu.vn](mailto:thaont@tlu.edu.vn)

# Chương 2: Tầng ứng dụng

1. Nguyên lý của ứng dụng mạng
2. Web và HTTP
3. Truyền tập tin: FTP
4. Thư điện tử
5. Hệ thống tên miền: DNS

# 1. Nguyên lý của ứng dụng mạng

## a. Kiến trúc ứng dụng mạng

### □ Ứng dụng mạng (network application)

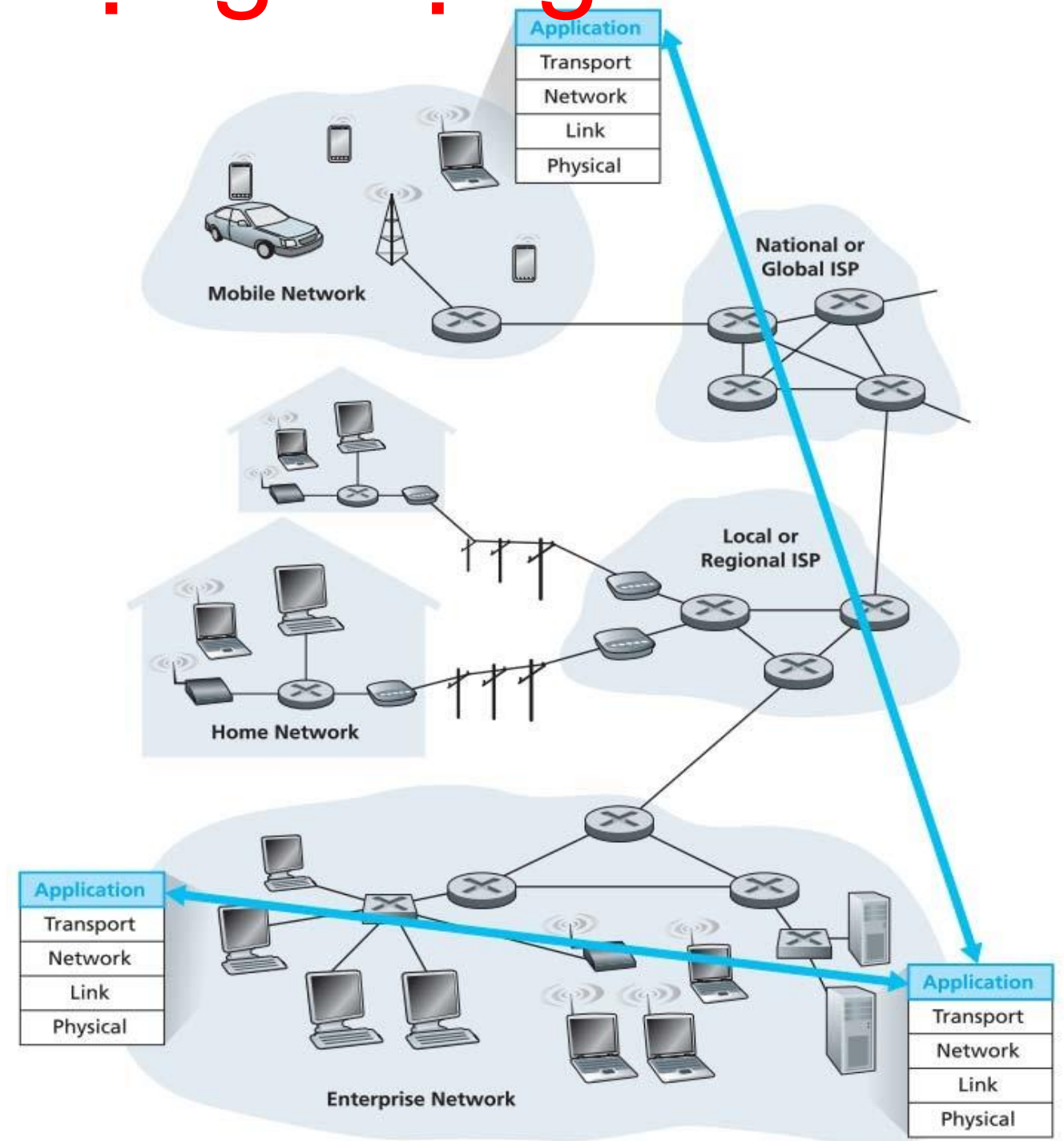
- Chạy trên các host
- Giao tiếp với host khác qua mạng

### □ Phát triển ứng dụng mạng

- SD các NNLT khác nhau
- Không cần viết cho các tb phần lõi mạng
- Xây dựng **kiến trúc ứng dụng**

### □ Kiến trúc ứng dụng

- Kiến trúc Client-Server
- Kiến trúc P2P
- Kiến trúc lai

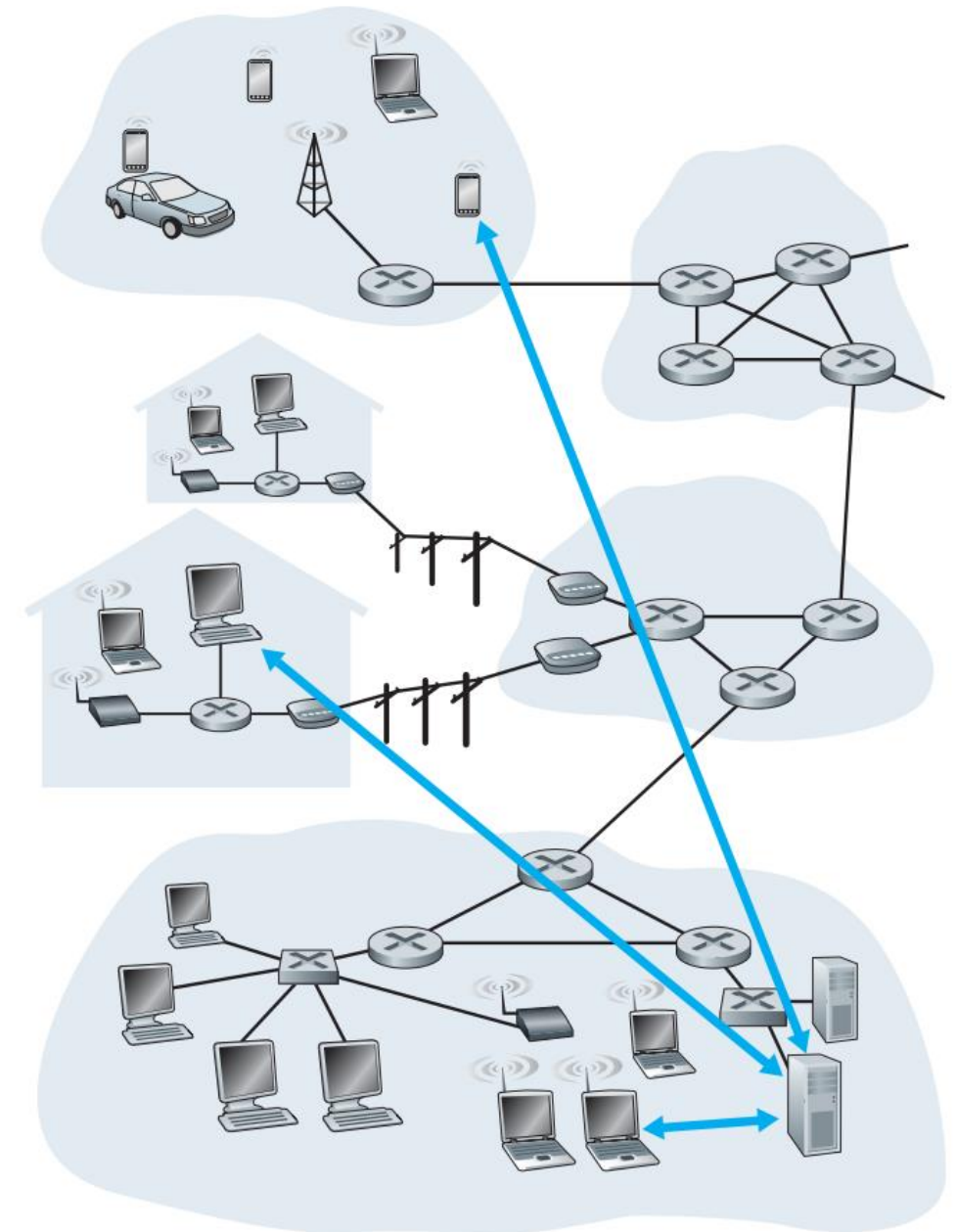


# 1. Nguyên lý của ứng dụng mạng

## a. Kiến trúc ứng dụng mạng

### □ Kiến trúc client-server

- Ứng dụng: 2 loại: client program, server program
- Host:
  - Client: máy khách
    - Gửi yêu cầu truy cập dịch vụ đến máy chủ
    - các client không trực tiếp giao tiếp với nhau
  - Server: máy chủ,
    - Phục vụ yêu cầu từ máy khách
    - Có một địa chỉ cố định, nhiều người biết: địa chỉ IP
    - Luôn sẵn sàng
    - Phục vụ nhiều client cùng lúc
- Vd: Web, FTP, email. Telnet
- Data center



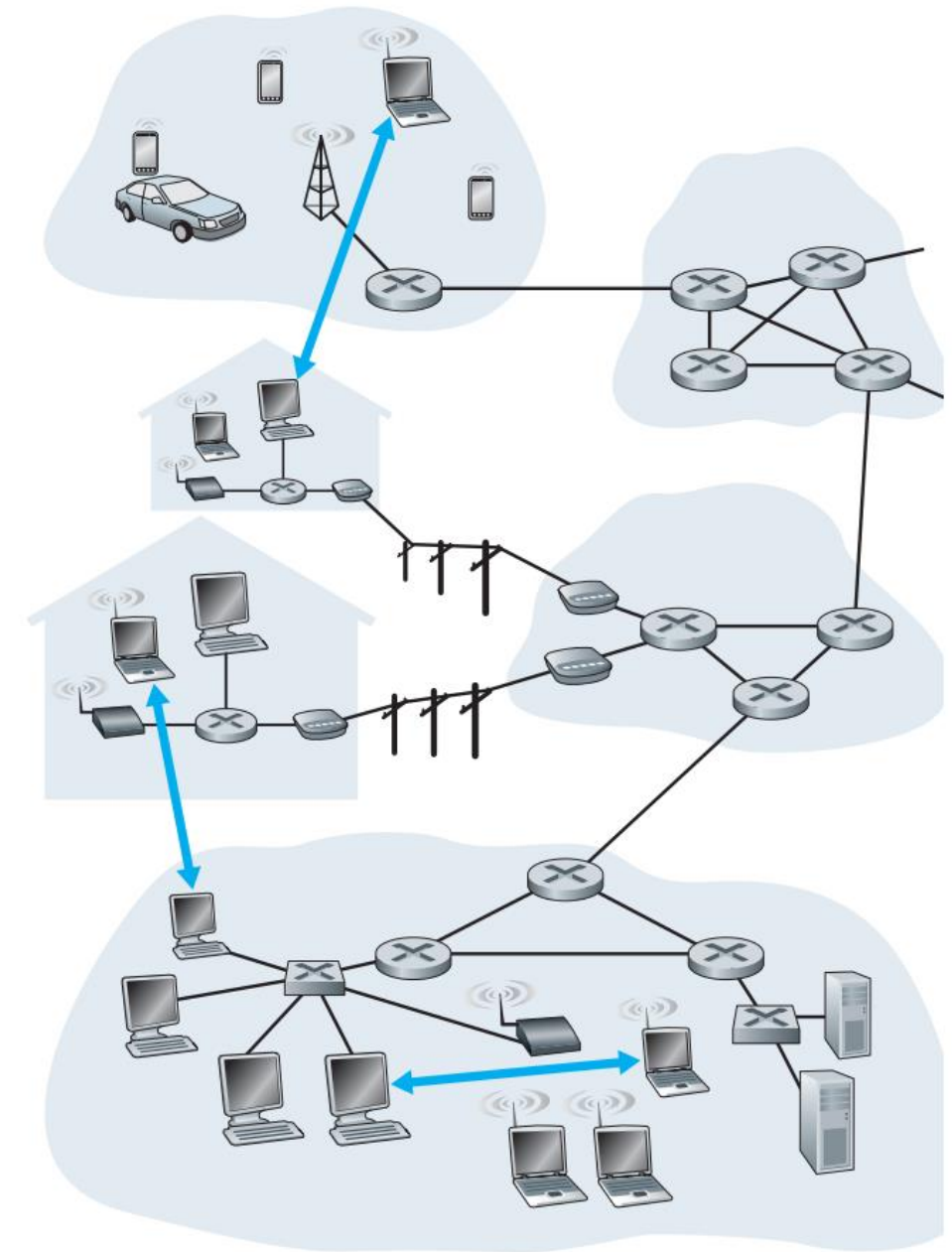
a. Client-server architecture

# 1. Nguyên lý của ứng dụng mạng

## a. Kiến trúc ứng dụng mạng

### □ Kiến trúc Peer-to-peer

- Ứng dụng: 1 loại chương trình cho máy người sử dụng:
  - Các máy ngang hàng nhau: Peer
  - Các Peer giao tiếp trực tiếp: Peer to peer
  - Không cần 1 máy chủ cung cấp dịch vụ
- Ví dụ: file sharing, điện thoại Internet
- Một vài ứng dụng hiện nay là ứng dụng lai: tin nhắn trực tuyến,...

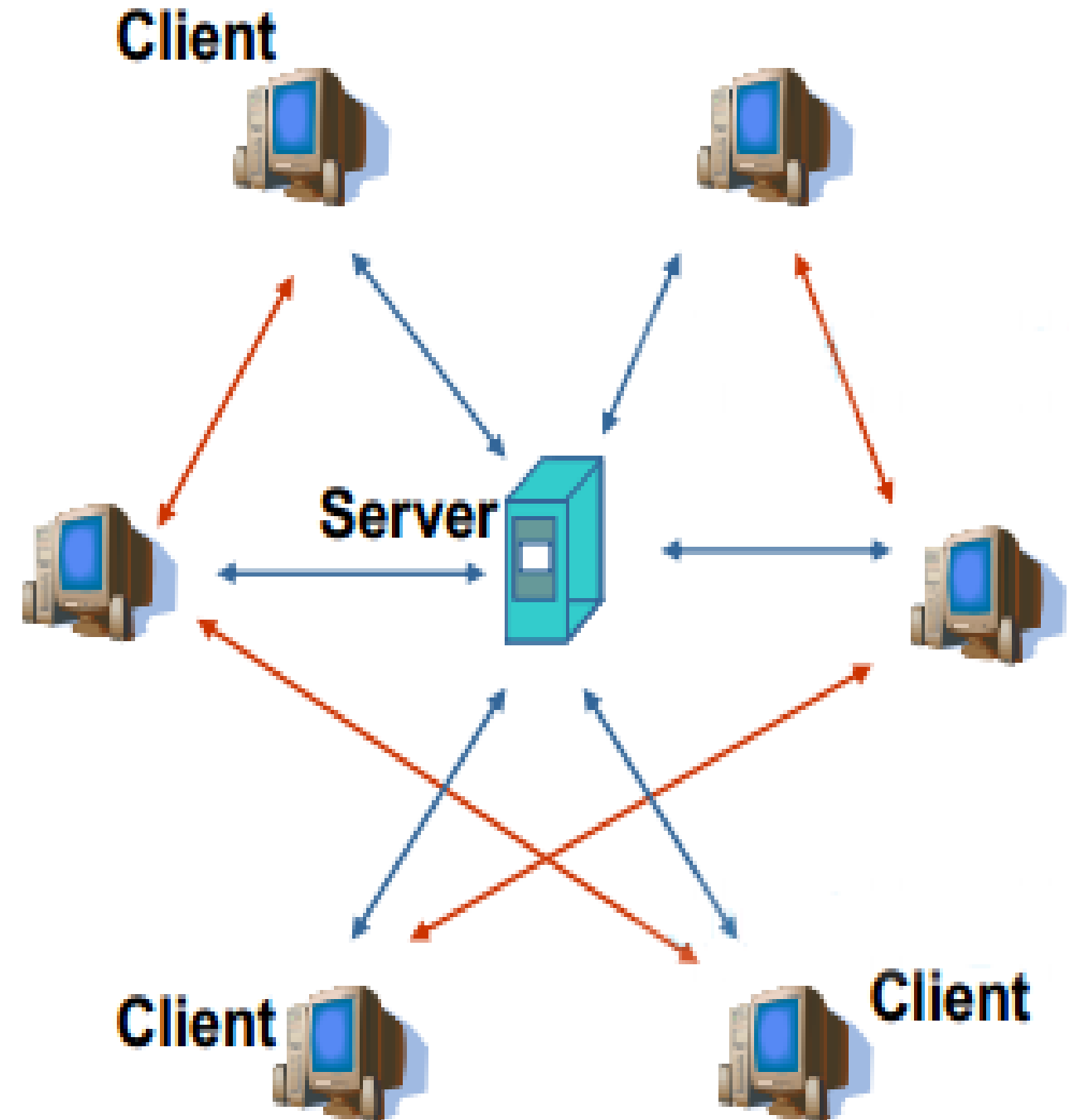


b. Peer-to-peer architecture

# 1. Nguyên lý của ứng dụng mạng

## a. Kiến trúc ứng dụng mạng

- ❑ Mô hình lai
  - ❑ Một máy chủ trung tâm để quản lý NSD, thông tin tìm kiếm
  - ❑ Các máy khách sẽ trực tiếp giao tiếp với nhau sau khi đăng nhập
  - ❑ Ví dụ: Skype
    - Máy chủ Skype quản lý các phiên đăng nhập, mật khẩu, ..
    - Sau khi kết nối, các máy sẽ gọi VoIP trực tiếp cho nhau

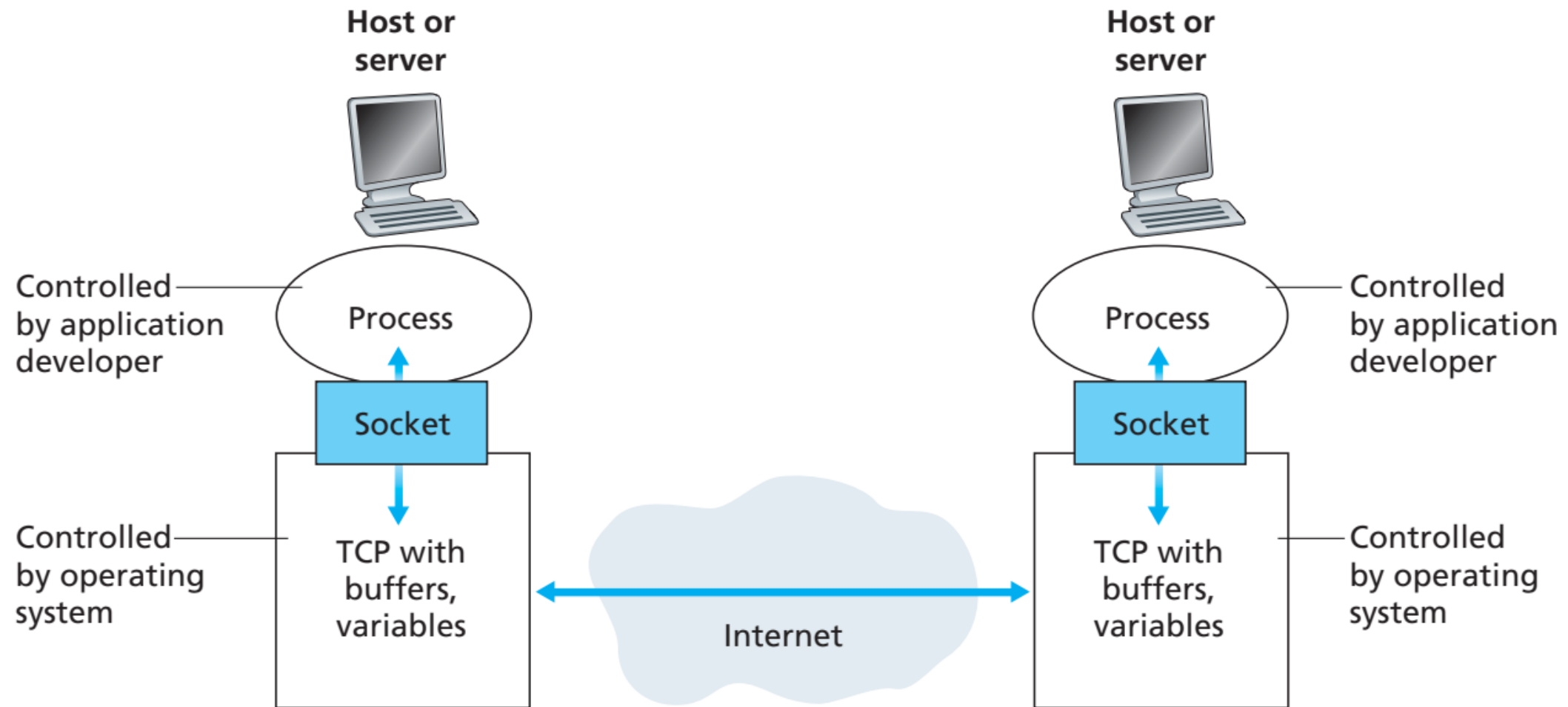




# 1. Nguyên lý của ứng dụng mạng

## b. Giao tiếp tiến trình

- ❑ Tiến trình: về mặt HĐH, một chương trình đang chạy được gọi là một tiến trình
- ❑ Các tiến trình của 2 thiết bị đầu cuối giao tiếp với nhau bằng cách chuyển các bản tin (message) qua MMT



# 1. Nguyên lý của ứng dụng mạng

## b. Giao tiếp tiến trình

- ❑ Tiến trình gửi thông điệp và nhận thông điệp từ mạng qua một giao diện phần mềm gọi là socket
  - Tiến trình Client: tiến trình khởi tạo giao tiếp
  - Tiến trình Server: tiến trình chờ để tiến trình khác liên lạc
- ❑ Socket là giao diện giữa tiến trình ứng dụng và tầng giao vận: còn đc gọi là API (Application Programming Interface)



# 1. Nguyên lý của ứng dụng mạng

## b. Giao tiếp tiến trình

### □ Địa chỉ tiến trình

- Khi một gói tin được gửi tới một tiến trình trên host. Để xác định ***tiến trình nhận***, cần có hai thông tin sau:

- Địa chỉ của host → **Địa chỉ IP (IP address) (Chương 4)**
  - *Là một chuỗi bit 32b giúp xác định một host duy nhất*
- Thông tin nào đó giúp xác định tiến trình nhận trong các các tiến trình của host đích → **Địa chỉ cổng (Port Number)**
  - *Vì một host có thể đang chạy nhiều ứng dụng mạng. Vd: Web server: 80, mail server: 25*
  - *<http://www.iana.org>*

# 1. Nguyên lý của ứng dụng mạng

## c. Các dịch vụ tầng giao vận dành cho ứng dụng

- ❑ Socket là giao diện giữa ứng dụng và các giao thức tầng giao vận
- ❑ Khi phát triển ứng dụng, LTV lựa chọn các giao thức tầng giao vận phù hợp
- ❑ Các dịch vụ tầng giao vận cung cấp?
  - Truyền dữ liệu tin cậy:
  - Thông lượng:
  - Thời gian
  - An toàn bảo mật

# 1. Nguyên lý của ứng dụng mạng

## c. Các dịch vụ tầng giao vận dành cho ứng dụng

### □ Truyền dữ liệu tin cậy (**reliable data transfer**)

- Gói tin có thể bị thất lạc trong mạng máy tính. Khi nào?
- Đối với nhiều ứng dụng, thất lạc dữ liệu có thể gây hậu quả không chấp nhận được. Ứng dụng nào? Tại sao?
- Dịch vụ truyền dữ liệu đảm bảo (tin cậy): có cơ chế để đảm bảo dữ liệu được chuyển đúng và đầy đủ tới ứng dụng nhận
- Tầng giao vận cung cấp cả dịch vụ truyền tin không tin cậy dành cho một số ứng dụng nhất định
  - Ví dụ ứng dụng truyền âm thanh, video, hình ảnh

# 1. Nguyên lý của ứng dụng mạng

## c. Các dịch vụ tầng giao vận dành cho ứng dụng

### □ Thông lượng

- Thông lượng là tốc độ mà tiến trình gửi có thể chuyển các bit tới tiến trình nhận.
- Thông lượng khả dụng có thể thay đổi theo thời gian. Tại sao?
- Tầng giao vận có thể cung cấp dịch vụ đảm bảo thông lượng khả dụng
  - Ứng dụng có thể yêu cầu thông lượng khả dụng  $r$  bit/giây và tầng giao vận sẽ đảm bảo rằng thông lượng khả dụng luôn thấp nhất là  $r$  bit/giây
- Ứng dụng cần đảm bảo yêu cầu thông lượng gọi là ***ứng dụng nhạy cảm băng thông (bandwidth-sensitive applications)***
  - Ví dụ, ứng dụng điện thoại Internet cần 32 kbps
- **Elastic application** có thể sử dụng băng thông ít hoặc nhiều tùy theo thông lượng khả dụng. Ví dụ elastic apps?

# 1. Nguyên lý của ứng dụng mạng

## c. Các dịch vụ tầng giao vận dành cho ứng dụng

### □ Thời gian

- Đảm bảo thời gian: ví dụ mỗi bit gửi từ socket gửi tới socket nhận không chậm hơn 10msec
- Dịch vụ kiểu này cần cho các ứng dụng thời gian thực có tính tương tác, ví dụ Internet telephony, virtual environment, teleconferencing, và multiplayer game
- Trong trò chơi nhiều người chơi hoặc môi trường tương tác ảo, độ trễ lớn giữa hành động và thể hiện được hành động đó trong môi trường (ví dụ, người chơi khác quan sát thấy) sẽ làm giảm tính thực tế của ứng dụng

# 1. Nguyên lý của ứng dụng mạng

## c. Các dịch vụ tầng giao vận dành cho ứng dụng

### □ An toàn bảo mật

- Giao thức tầng giao vận có thể cung cấp một hoặc nhiều dịch vụ an toàn bảo mật
- Ví dụ, tại nút gửi, giao thức tầng giao vận có thể mã hóa dữ liệu mà tiến trình gửi truyền đi, và tại nút nhận, giao thức tầng giao vận có thể giải mã dữ liệu trước khi chuyển tới tiến trình nhận
- Những dịch vụ như vậy có thể cung cấp tính bí mật giữa hai tiến trình, thậm chí nếu dữ liệu bị quan sát khi truyền từ tiến trình gửi tới tiến trình nhận
- Giao thức tầng giao vận có thể cung cấp các dịch vụ an toàn bảo mật khác ngoài tính bí mật, ví dụ như tính toàn vẹn dữ liệu, tính xác thực

# 1. Nguyên lý của ứng dụng mạng

## d. Dịch vụ tầng giao vận

Application	Data Loss	Throughput	Time-Sensitive
File transfer/download	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps–1 Mbps Video: 10 kbps–5 Mbps	Yes: 100s of msec
Streaming stored audio/video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps–10 kbps	Yes: 100s of msec
Instant messaging	No loss	Elastic	Yes and no



# 1. Nguyên lý của ứng dụng mạng

## d. Dịch vụ tầng giao vận

- ❑ **Dịch vụ TCP:** Mô hình dịch vụ TCP bao gồm dịch vụ hướng kết nối và dịch vụ truyền dữ liệu tin cậy
- ❑ **Dịch vụ UDP:** UDP là không hướng kết nối. UDP cung cấp dịch vụ truyền dữ liệu không tin cậy
- ❑ Các dịch vụ không được cung cấp bởi các giao thức tầng giao vận của Internet
  - **An toàn bảo mật:** TCP cung cấp truyền dữ liệu tin cậy giữa hai nút và có thể dễ dàng bổ sung SSL tại tầng ứng dụng để cung cấp dịch vụ an toàn bảo mật
  - Đảm bảo thông lượng và thời gian là dịch vụ không được cung cấp.
  - Internet ngày nay có thể cung cấp các dịch vụ chấp nhận được *cho các ứng dụng yêu cầu thời gian* nhưng không thể đảm bảo về thông lượng hoặc thời gian

# Dịch vụ tầng giao vận của Internet

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 5321]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (e.g., YouTube)	TCP
Internet telephony	SIP [RFC 3261], RTP [RFC 3550], or proprietary (e.g., Skype)	UDP or TCP

# 1. Nguyên lý của ứng dụng mạng

## e. Giao thức tầng ứng dụng

- ❑ Giao thức tầng ứng dụng định nghĩa các quy tắc giao tiếp
  - Kiểu của bản tin, ví dụ bản tin yêu cầu (request message), bản tin trả lời (response message)
  - Cú pháp của các kiểu bản tin, ví dụ các trường thông tin trong bản tin và cách thức phân chia các trường
  - Ngữ nghĩa của các trường: thông tin trong trường này dùng để làm gì
  - Nguyên tắc để xác định một tiến trình gửi và trả lời gói tin khi nào và như thế nào.
- ❑ Một số giao thức tầng ứng dụng được đặc tả trong các RFC:
  - Giao thức tầng ứng dụng Web: HTTP (HyperText Transfer Protocol [RFC 2616])
- ❑ Nhiều giao thức tầng ứng dụng là có bản quyền sở hữu
  - Giao thức tầng ứng dụng của Skype

# Chương 2: Tầng ứng dụng

1. Đặc điểm của ứng dụng mạng
2. Web và HTTP
3. Truyền tập tin: FTP
4. Thư điện tử
5. Hệ thống tên miền: DNS

## 2. Web và HTTP

### a. Tổng quan về HTTP

- ❑ WWW (World Wide Web): Berners-Lee 1994
  - Trao đổi dữ liệu siêu văn bản HTML (HyperText Markup Language) trên Internet
- ❑ Đặc điểm mới thu hút người sử dụng là Web hoạt động theo yêu cầu *(on demand)*
- ❑ Mọi người có thể trở thành **người xuất bản** với chi phí rất thấp

# Một số thuật ngữ về Web

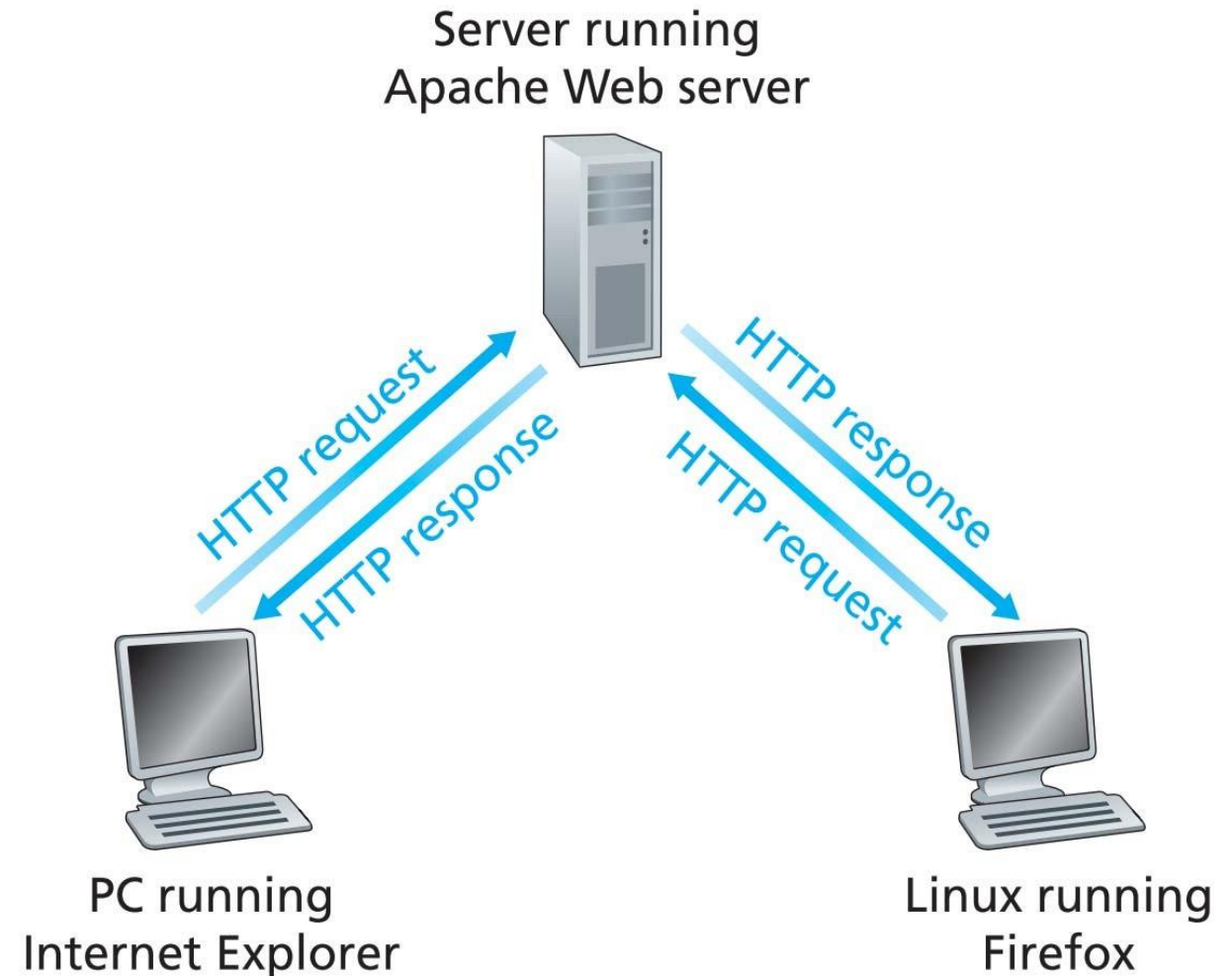
- ❑ **Trang Web (Webpage** – hay còn gọi là một tập tin) chứa các ***đối tượng***.
  - ❑ ***Đối tượng*** là một file - chẳng hạn như file HTML, hình ảnh JPEG, hoặc một video clip, ....
  - ❑ Mỗi đối tượng được xác định qua địa chỉ URL duy nhất
- ❑ **URL (Uniform Resource Locator**, nghĩa tiếng Việt: định vị tài nguyên thống nhất)
  - ❑ được gọi một cách thông thường là một địa chỉ web)
  - ❑ là một tham chiếu đến tài nguyên web chỉ định vị trí của nó trên một mạng máy tính để truy xuất nó.
  - ❑ Mỗi URL có hai thành phần: tên của máy chủ (hostname) và tên đường dẫn của đối tượng.

☐ Ví dụ: URL: <http://www.someSchool.edu/someDepartment/picture.gif>

tên máy chủ                      đường dẫn

# Giới thiệu về HTTP

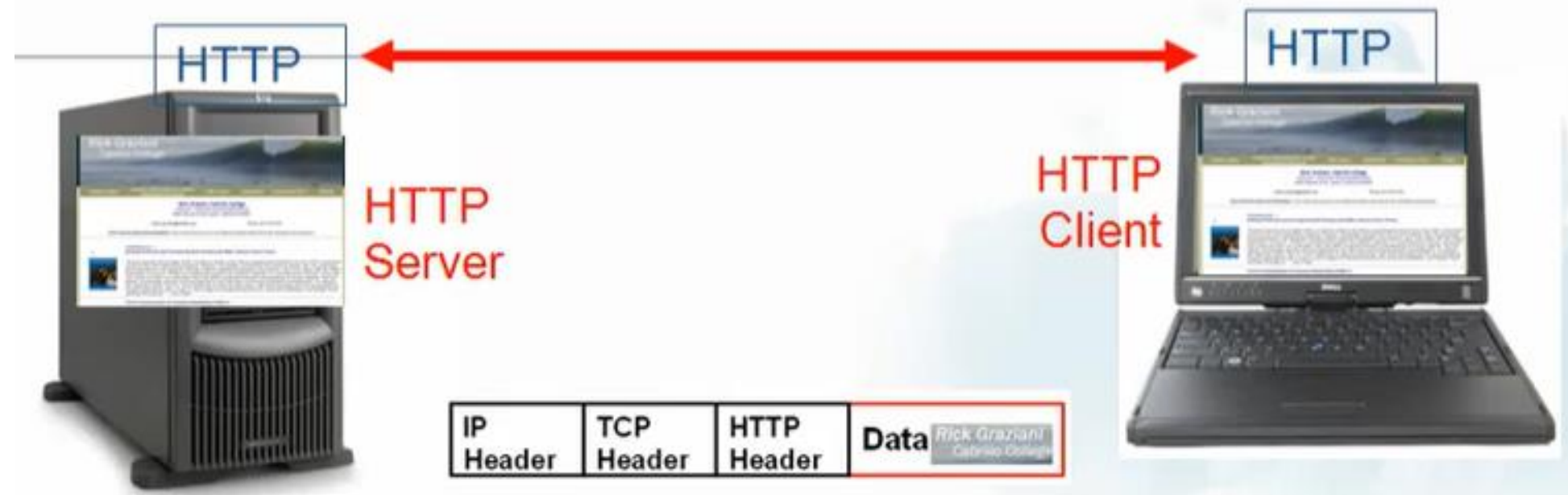
- ❑ HTTP (HyperText Transfer Protocol):  
Giao thức tầng ứng dụng của Web
  - RFC 1945 và RFC 2616
  - RFC 7540 (May 2015)
- ❑ HTTP sử dụng TCP làm giao thức tầng giao vận
- ❑ Mô hình Client-Server
  - ❑ Client: yêu cầu truy nhập tới các trang web (đối tượng web)
  - ❑ Server: nhận yêu cầu và trả lời cho client
- ❑ HTTP là giao thức không trạng thái (**stateless protocol**)



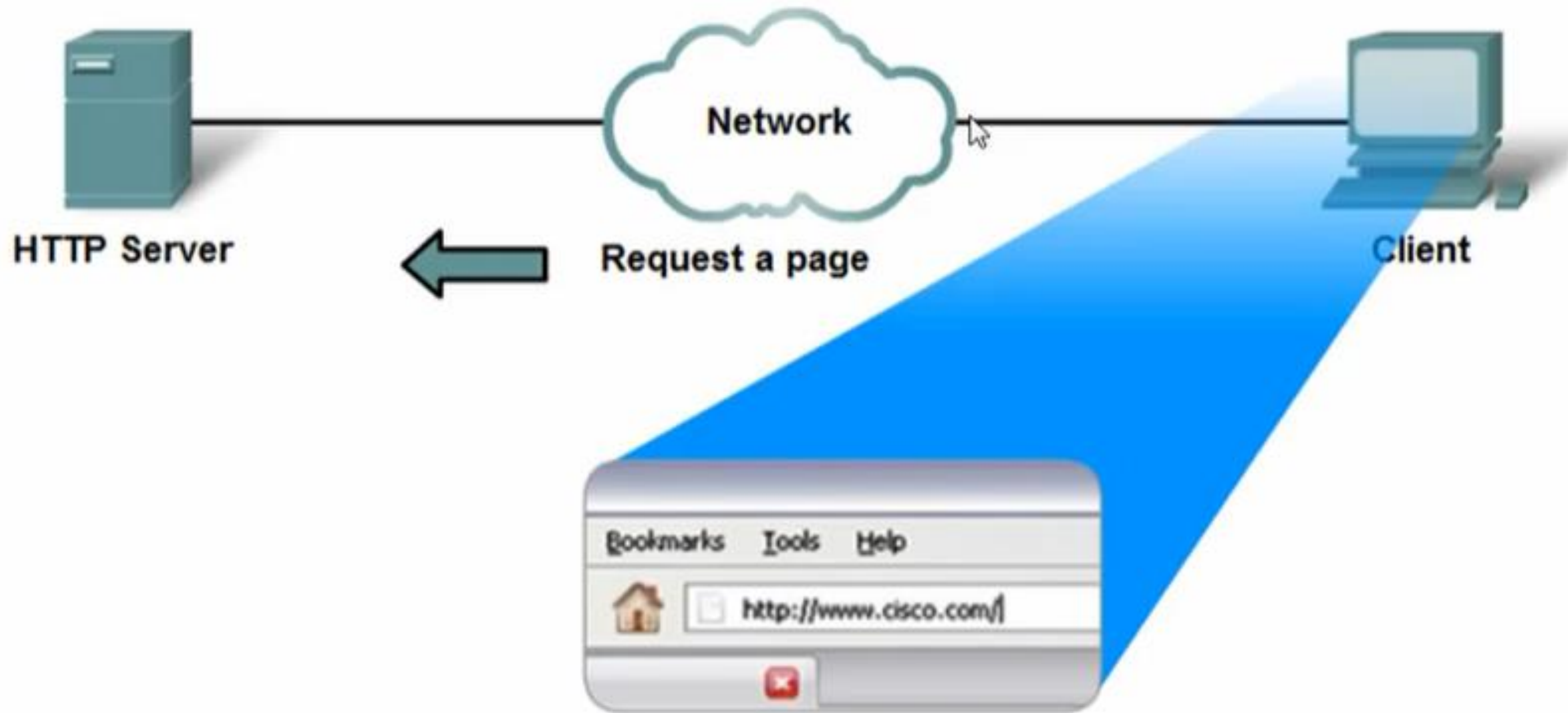


# HTTP hoạt động như thế nào?

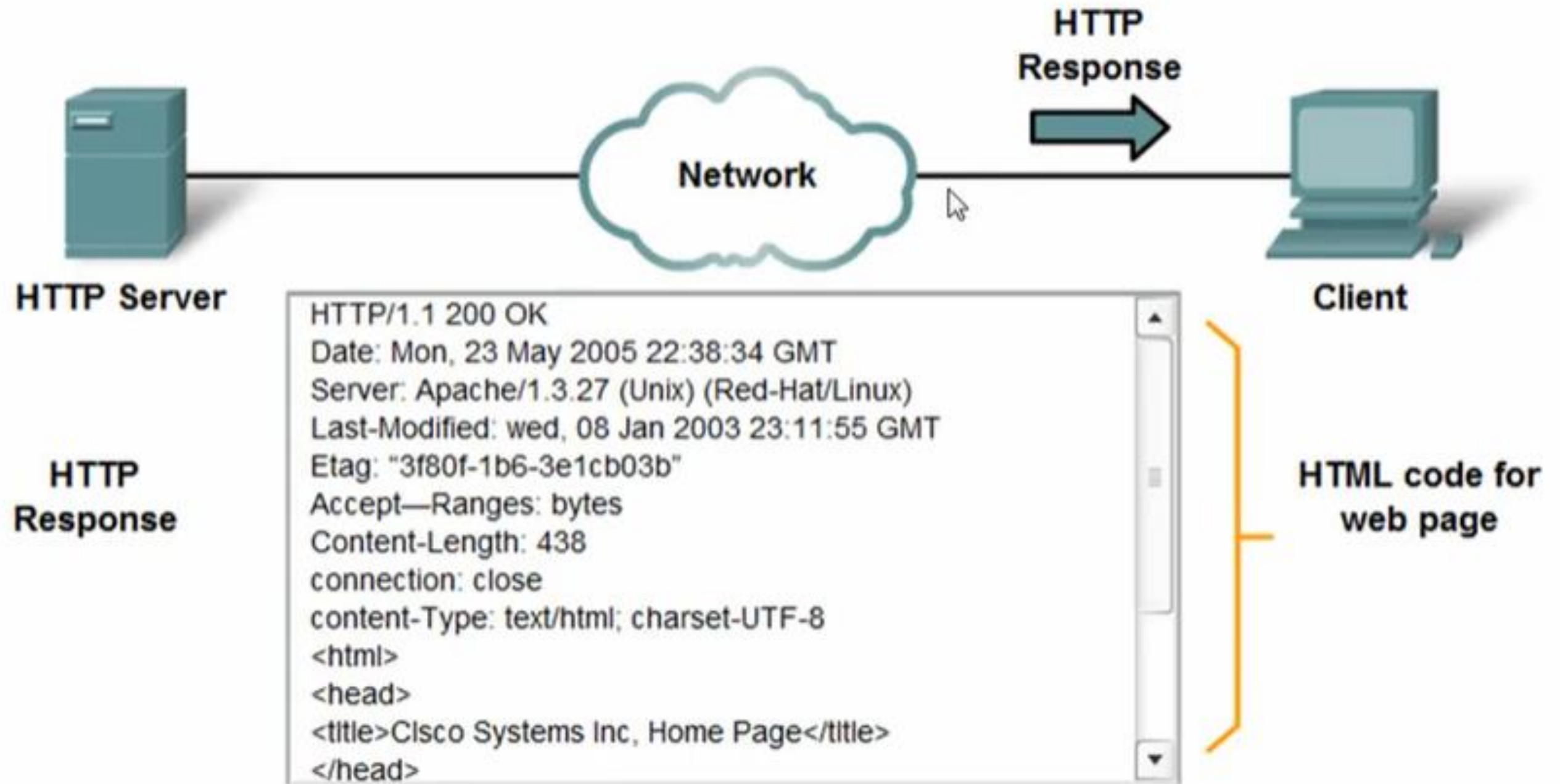
- ❑ Server mở một TCP socket chờ tại cổng 80
- ❑ Client khởi tạo một kết nối TCP tới server
- ❑ Server chấp nhận yêu cầu tạo kết nối
- ❑ Trao đổi thông điệp HTTP
  - HTTP request
  - HTTP Response
- ❑ Đóng kết nối TCP



# HTTP Protocol

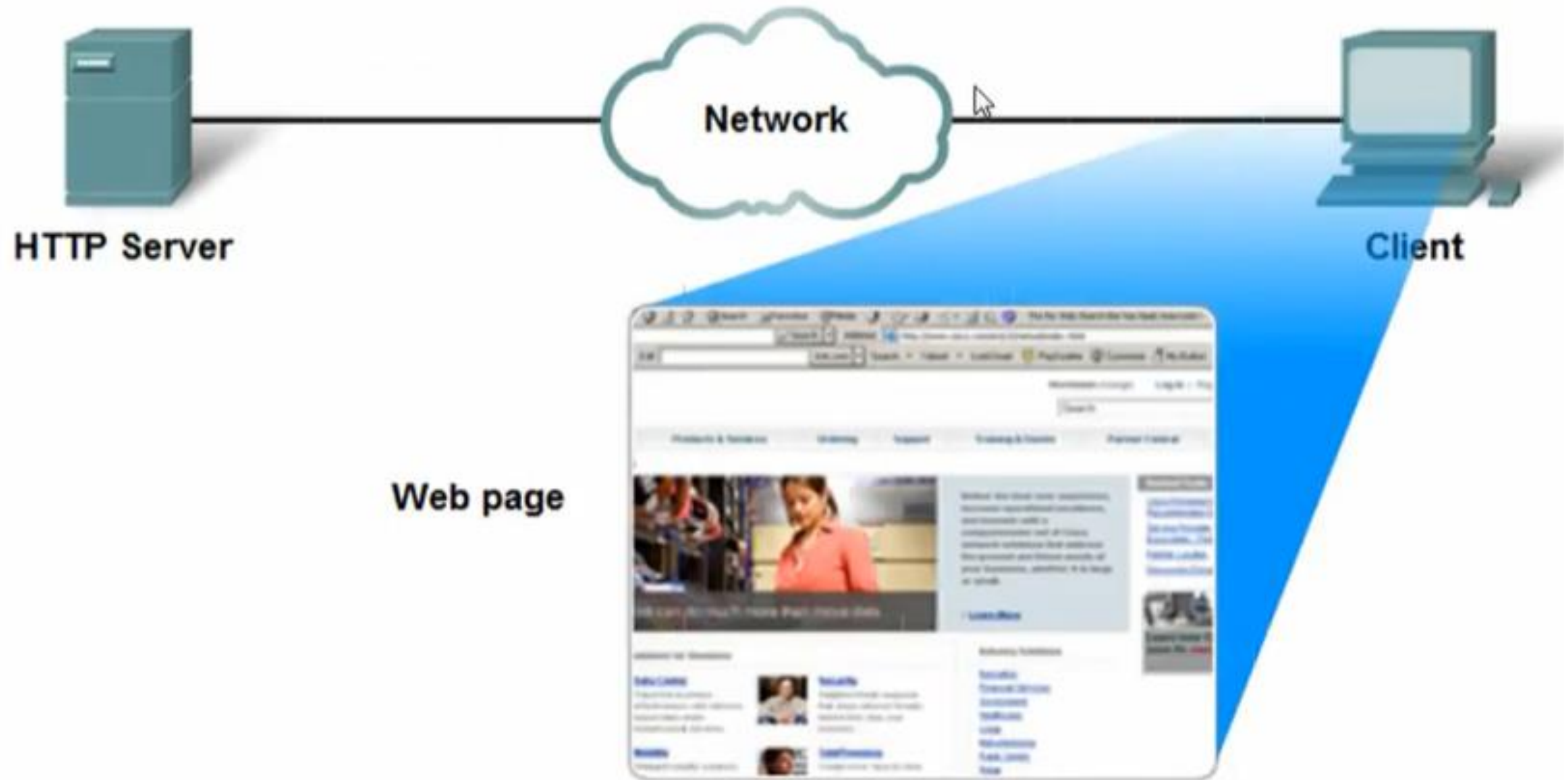


## HTTP Protocol



**In response to the request, the HTTP server returns code for a web page.**

# HTTP Protocol



The browser interprets the HTML code and displays a web page.

## b. Kết nối Non-Persistent và Persistent

### *Kiểu không giữ kết nối (non-persistent HTTP)*

- ❑ Một đối tượng (object) gửi qua kết nối TCP
  - Sau đó kết nối TCP đóng
- ❑ Tải nhiều object cần nhiều kết nối

### *Kiểu giữ kết nối (persistent HTTP)*

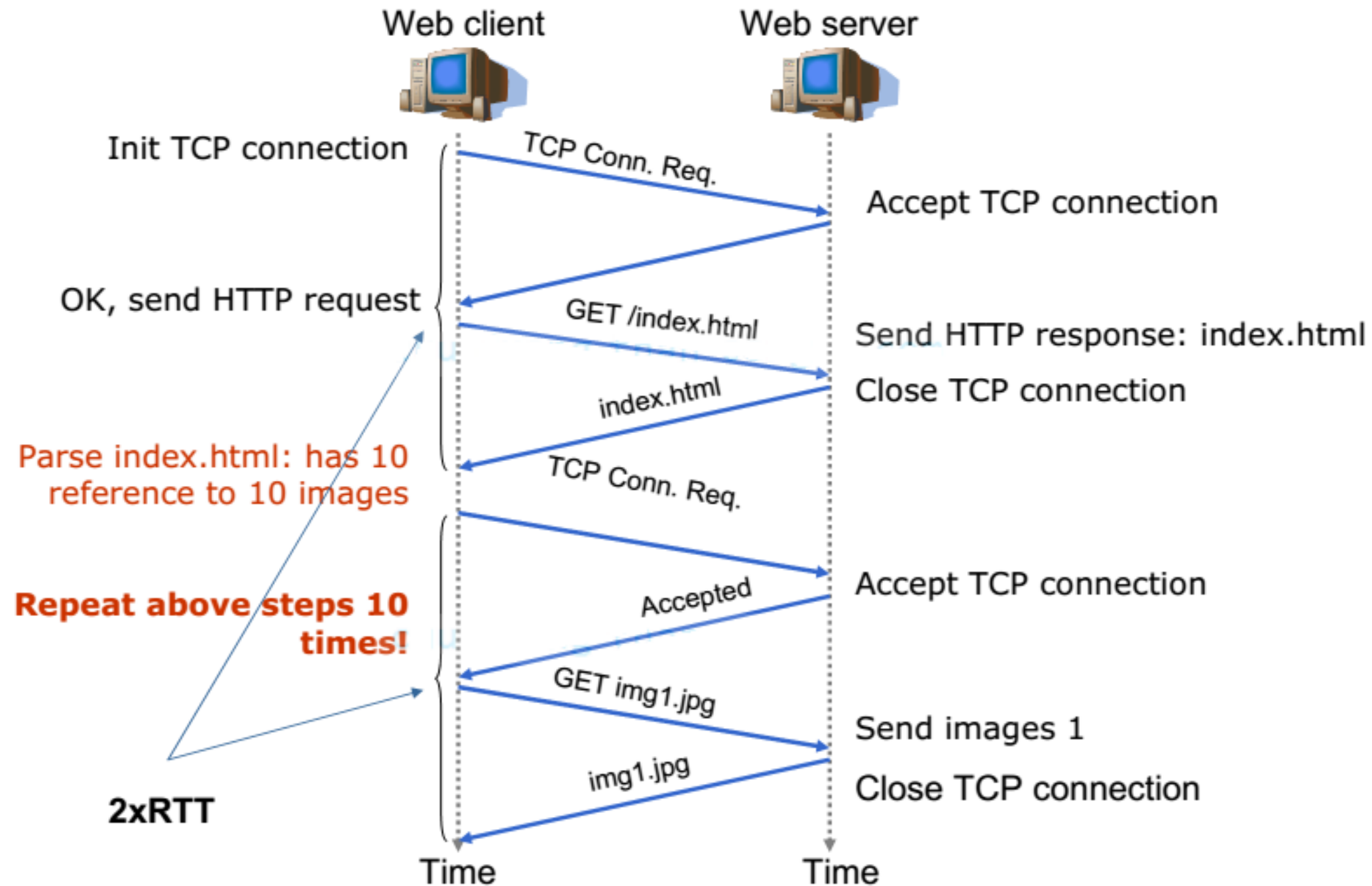
- ❑ Nhiều objects có thể gửi qua một kết nối TCP giữa client và server



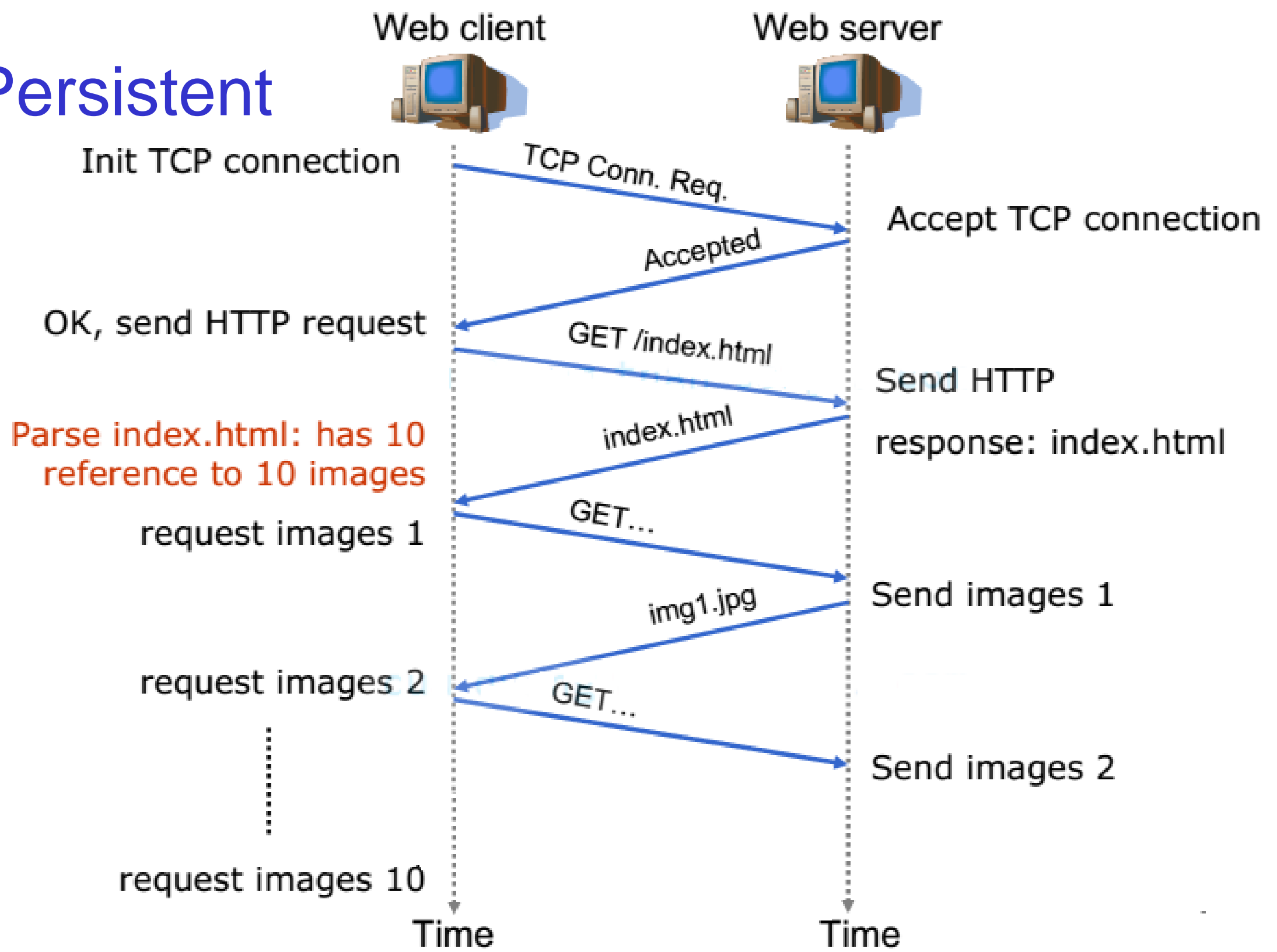
# Kết nối Non-Persistent

www.someSchool.edu/someDepartment/index.html

(chứa văn bản,  
và tham chiếu tới 10  
ảnh jpg)



# Kết nối Persistent





# Khuôn dạng HTTP request message

- ASCII

request line  
(GET, POST,  
HEAD commands)

header  
lines

carriage return,

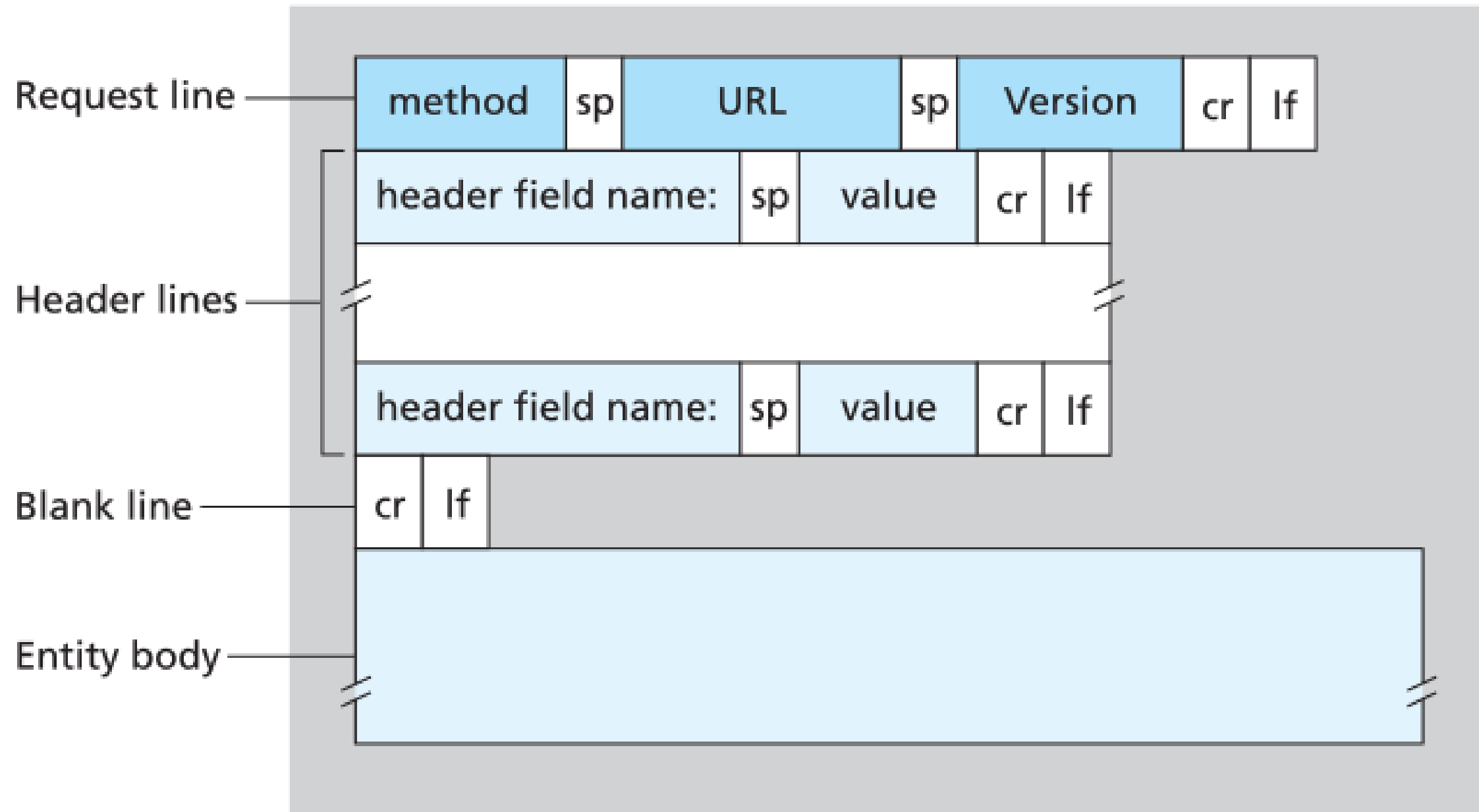
```
GET /somedir/page.html HTTP/1.1 \r\n
Host: www.someschool.edu \r\n
Connection: close \r\n
User-agent: Mozilla/5.0 \r\n
Accept-language: Fr \r\n
\r\n
```

carriage return character  
line-feed character

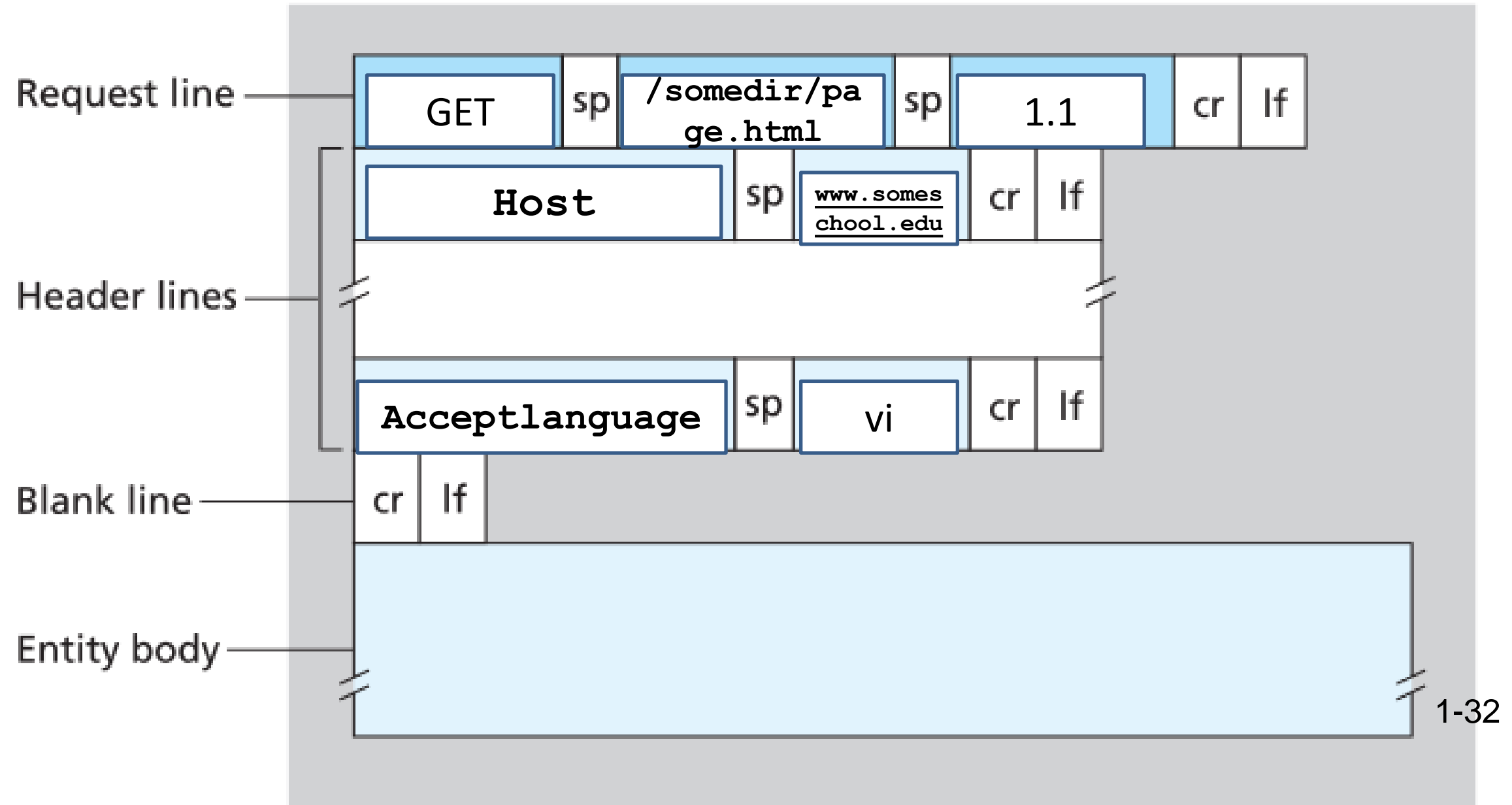
line feed ở đầu dòng chỉ ra kết thúc phần header

Địa chỉ URL của văn bản mà trình duyệt yêu cầu?

# Khuôn dạng HTTP request message



# Khuôn dạng HTTP request message



# Khuôn dạng HTTP response message

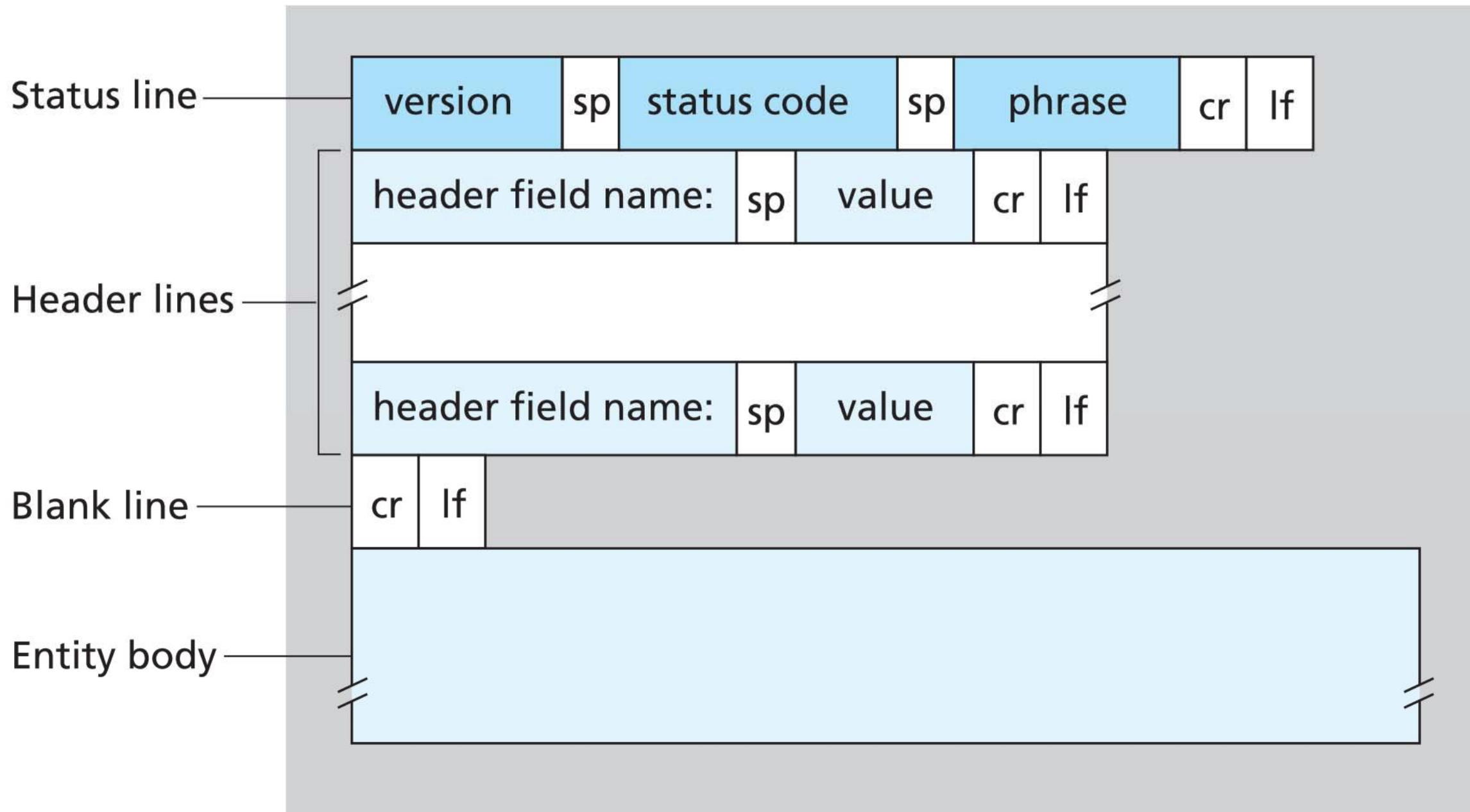
status line  
(protocol  
status code  
status phrase)

header  
lines

```
HTTP/1.1 200 OK \r\n
Connection: close \r\n
Date: Tue, 09 Aug 2011 15:44:04 GMT \r\n
Server: Apache/2.2.3 (CentOS) \r\n
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT\r\n
Content-Length: 6821 \r\n
Content-Type: text/html \r\n
\r\n
data data data data data ...
```

data, e.g.,  
requested  
HTML file

# Khuôn dạng HTTP response message



# Một số mã trạng thái phổ biến

- **200 OK**: Yêu cầu thành công và thông tin được trả về trong phản hồi.
- **301 Moved Permanently**: Đối tượng được yêu cầu đã bị di chuyển vĩnh viễn; URL mới được chỉ định trong trường Location (vị trí) của thông điệp phản hồi. Phần mềm máy khách sẽ tự động truy xuất URL mới.
- **400 Bad Request**: Server không hiểu được yêu cầu từ client.
- **404 Not Found**: Tài liệu được yêu cầu không tồn tại trên máy chủ.
- **505 HTTP Version Not Supported**: Phiên bản giao thức HTTP được yêu cầu không được máy chủ hỗ trợ.

# Quan sát HTTP message:

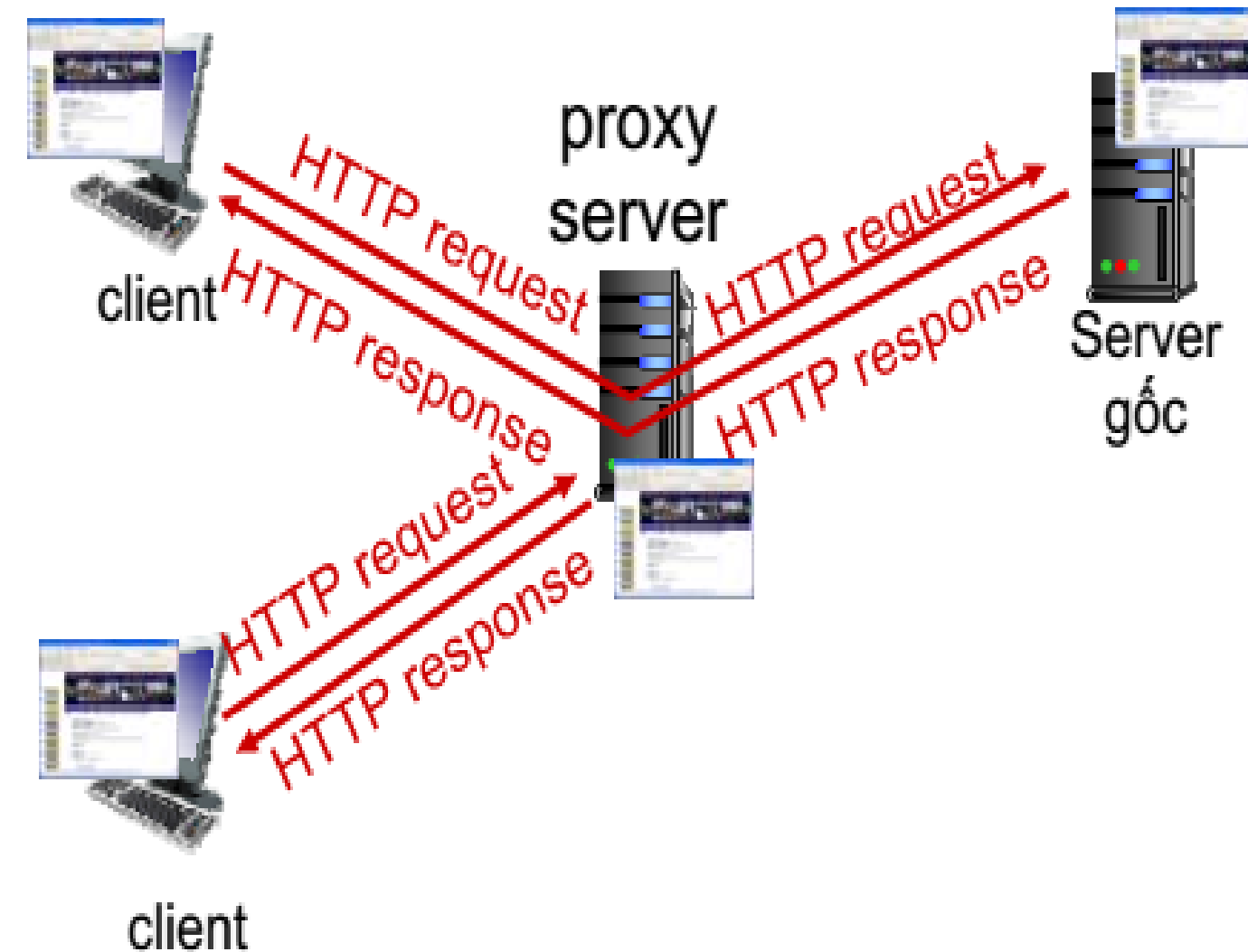
## Tạo HTTP request dùng trình duyệt

1. Chạy phần mềm Wireshark để ghi lại dữ liệu gửi nhận (Chọn Capture -> Start)
2. Mở trình duyệt vào trang web:  
`www.tlu.edu.vn`
3. Dừng việc ghi dữ liệu của phần mềm Wireshark
4. Gõ Filter: `http.host == "tlu.edu.vn"`
5. Quan sát HTTP message



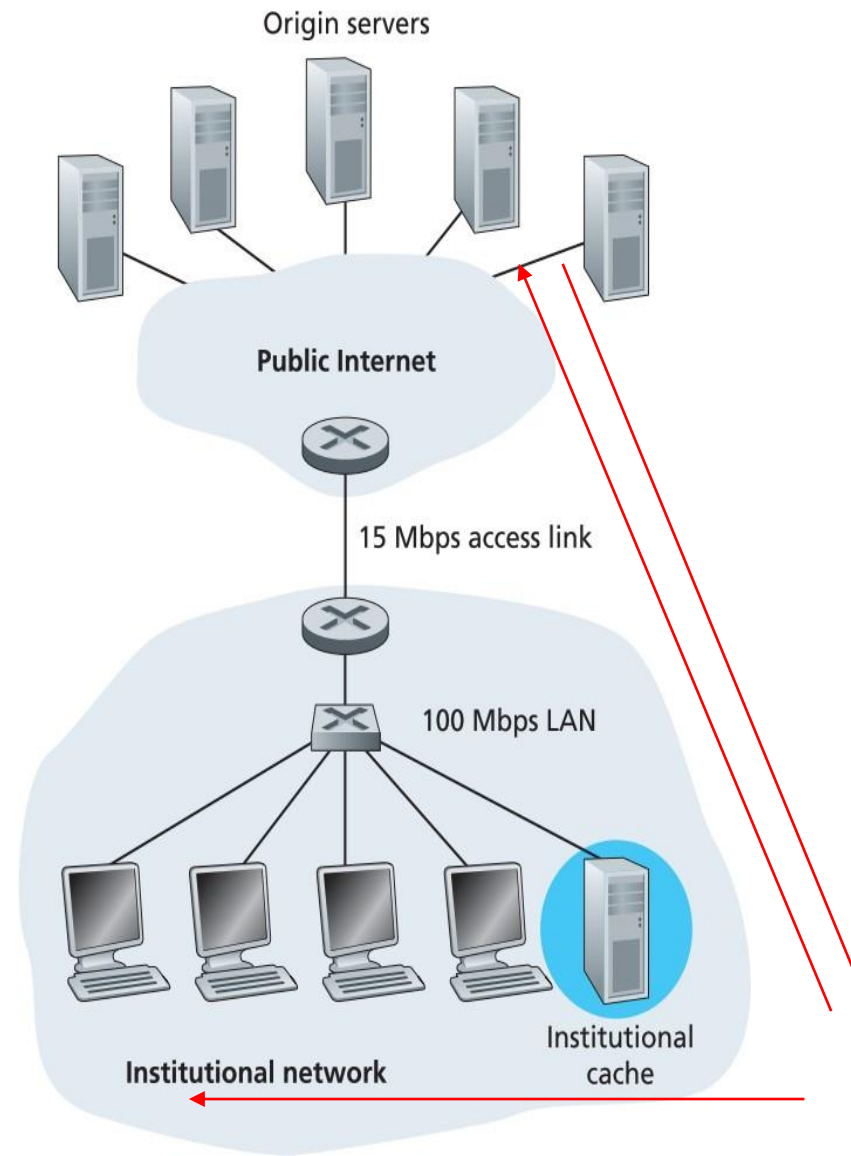
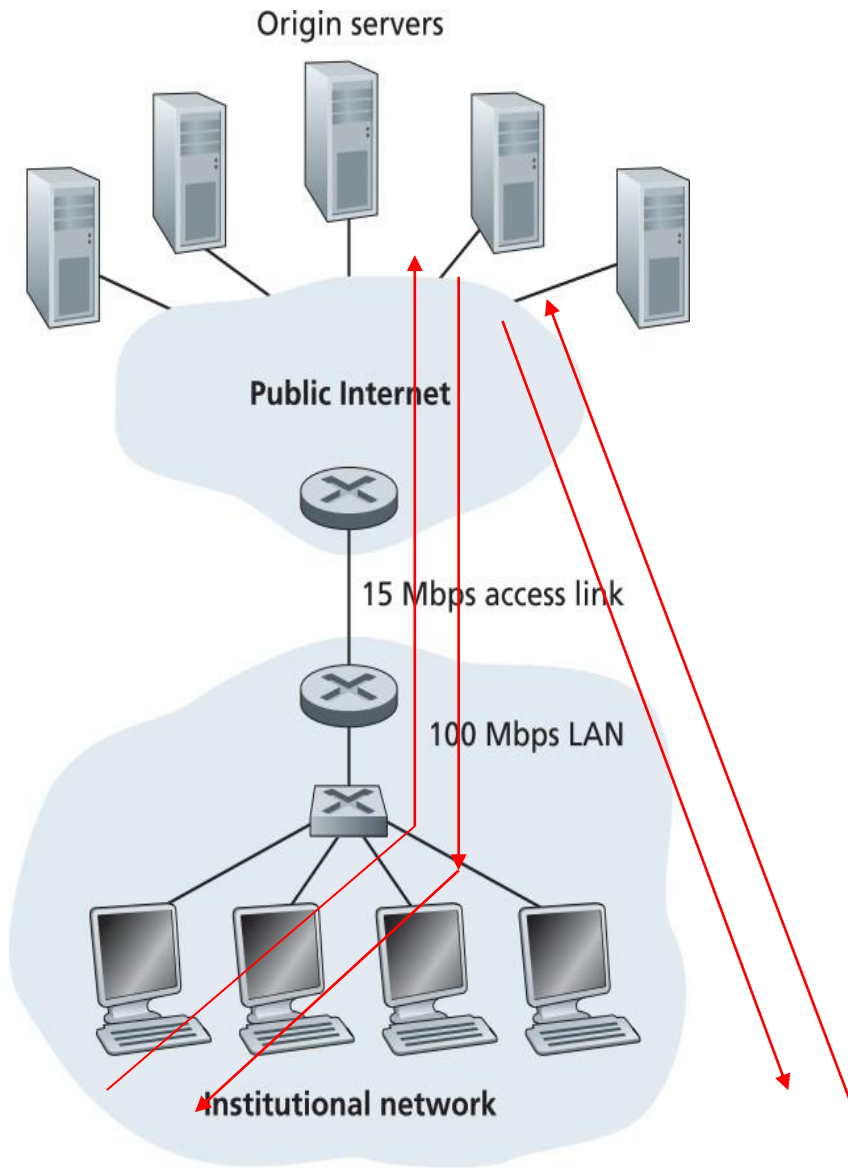
# Bộ đệm Web

- Bộ đệm Web (gọi là proxy server): có ổ lưu trữ riêng và giữ các bản sao của các đối tượng được yêu cầu gần đây.
- Trình duyệt gửi yêu cầu đối tượng web đến proxy server
  - Hit: Proxy trả đối tượng web cho trình duyệt
  - Miss: Proxy gửi yêu cầu tới máy chủ web, trả lời trình duyệt và lưu đối tượng web



# Bộ đệm Web

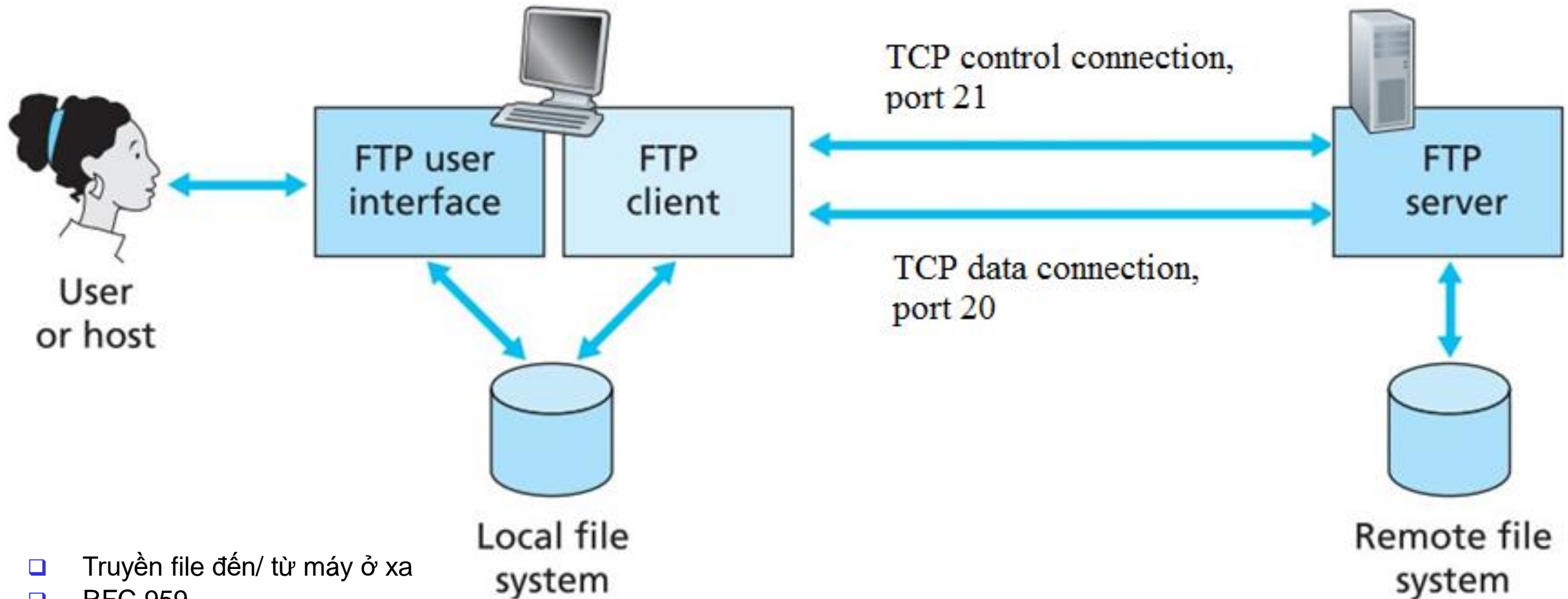
- ❑ Giảm lưu lượng Web trên đường truy cập ra Internet
- ❑ Giảm thời gian đáp ứng cho yêu cầu của client



# Chương 2: Tầng ứng dụng

1. Đặc điểm của ứng dụng mạng
2. Web và HTTP
3. Truyền tập tin: FTP
4. Thư điện tử
5. Hệ thống tên miền: DNS

# Truyền tập tin: FTP



- ❑ Truyền file đến/ từ máy ở xa
- ❑ RFC 959
- ❑ Mô hình client/server
- ❑ Kết nối điều khiển và kết nối dữ liệu riêng biệt
- ❑ FTP server duy trì trạng thái: thư mục hiện tại, xác thực trước đó
  - HTTP không duy trì trạng thái

# Quá trình thiết lập kết nối FTP

Khi một máy (local host) muốn truyền file đến một máy ở xa (remote host):

1. Cung cấp hostname cho FTP client
2. FTP client thiết lập kết nối TCP đến FTP server
3. User gửi thông tin xác thực (username và pass) đến server qua kết nối
4. FTP server sử dụng thông tin đó để xác thực
5. Truyền file

FTP:

❑ Sử dụng hai kết nối song song:

- ❑ Kết nối điều khiển: truyền thông tin điều khiển: xác thực, lệnh thay đổi đường dẫn, lệnh “put” hoặc “get” file
- ❑ Kết nối dữ liệu: khi truyền file, đóng sau khi xong

# Bản tin và phản hồi

- USER username: Gửi thông tin xác thực: username đến server.
- PASS password: Gửi thông tin xác thực: password đến server.
- LIST: Hỏi server danh sách các file trong thư mục hiện tại. Danh sách file sẽ được phía server gửi lại bằng một **kết nối dữ liệu**.
- RETR filename: Lấy một file từ thư mục hiện tại của phía server qua một **kết nối dữ liệu** mới.
- STOR filename: Đưa một file vào thư mục hiện tại của host từ xa.
- 331 Username OK, password required
- 125 Data connection already open; transfer starting
- 425 Can't open data connection
- 452 Error writing file

# Chương 2: Tầng ứng dụng

1. Đặc điểm của ứng dụng mạng
2. Web và HTTP
3. Truyền tập tin: FTP
4. Thư điện tử
5. Hệ thống tên miền: DNS

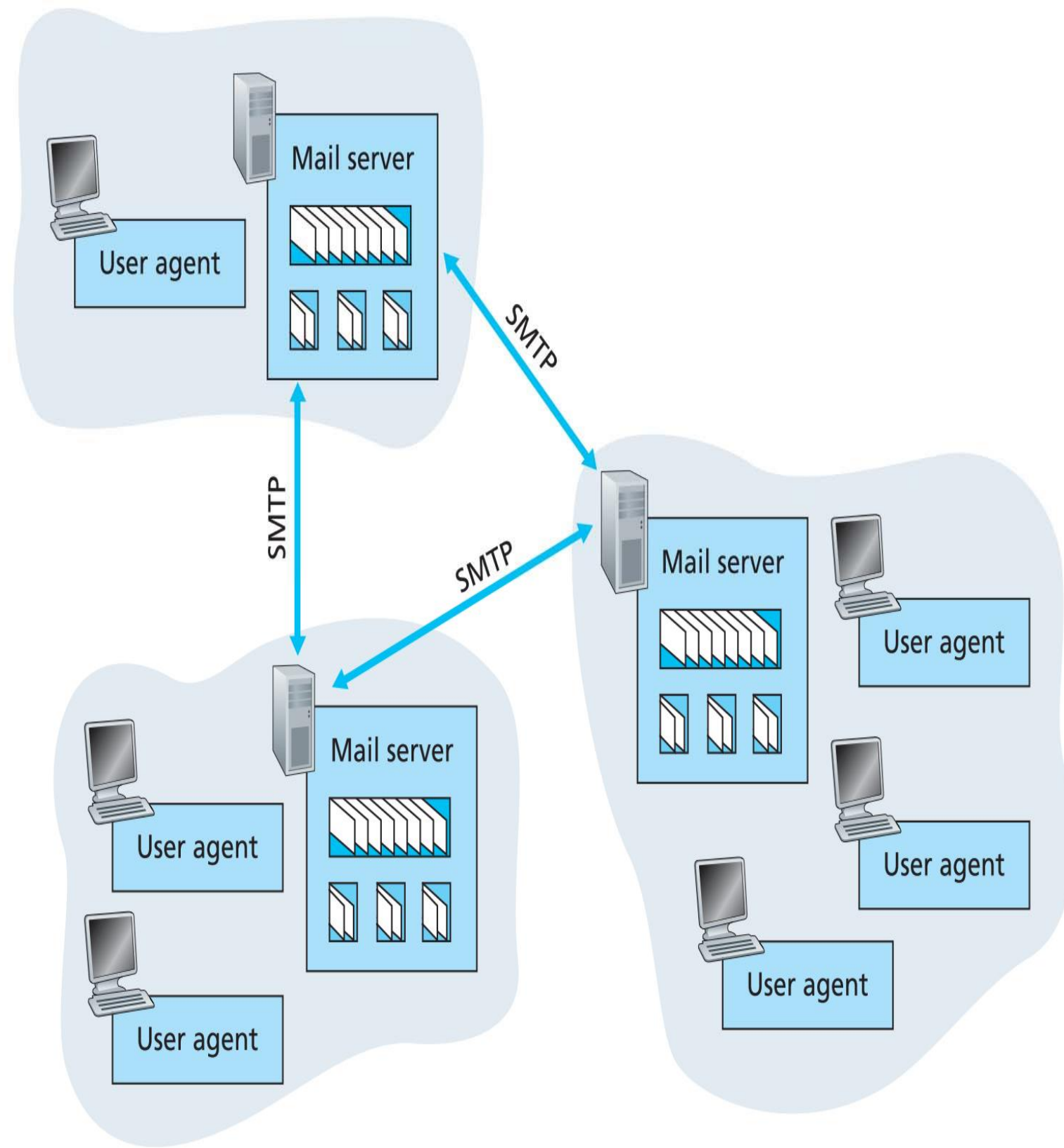
# Thư điện tử

## ❑ Ba thành phần chính

- User agent
- Mail server
- SMTP (simple mail transfer protocol)

## ❑ User agent:

- ❑ Chương trình giao tiếp người dùng, gọi là mail reader – trình đọc thư
- ❑ Soạn thảo, gửi, đọc và trả lời thư





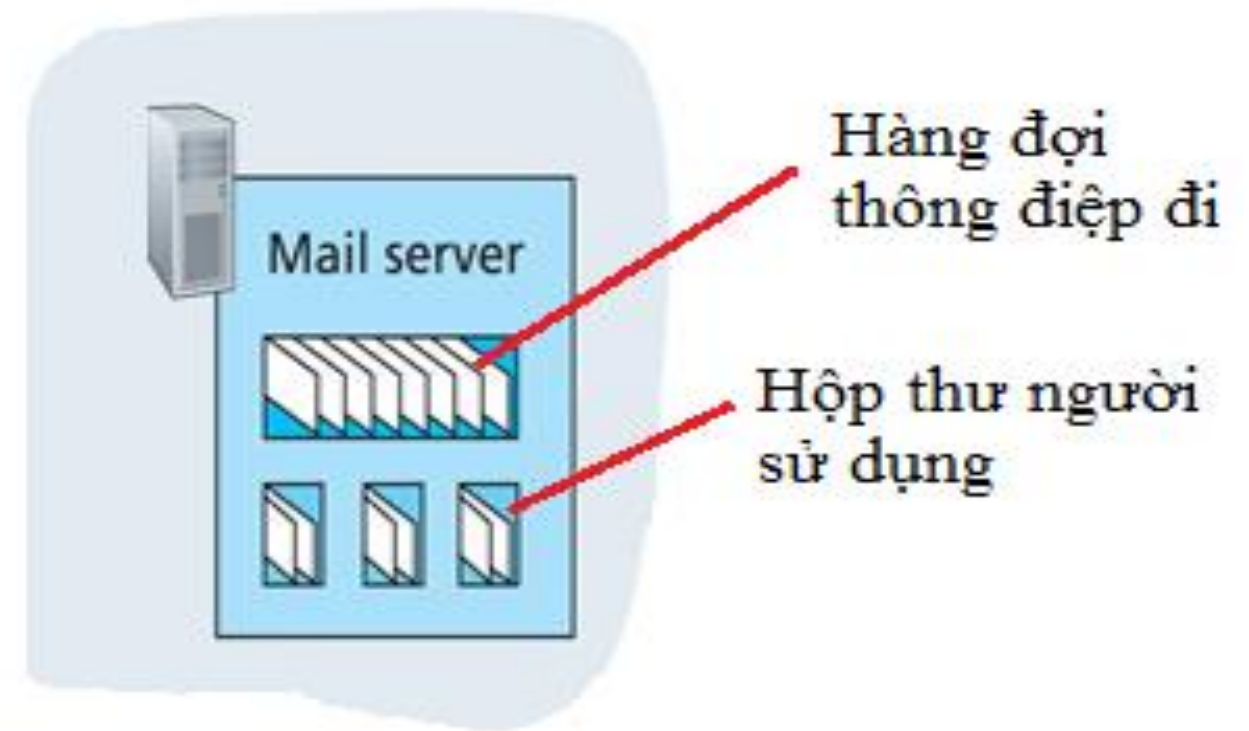
# Thư điện tử

## ❑ Mail server

- Hộp thư (mail box) chứa thông điệp đến user
- Hàng đợi chứa các thông điệp chuẩn bị gửi

## ❑ Giao thức SMTP giữa các mail server để gửi các thông điệp email

- Client: mail server gửi
- Server: mail server nhận

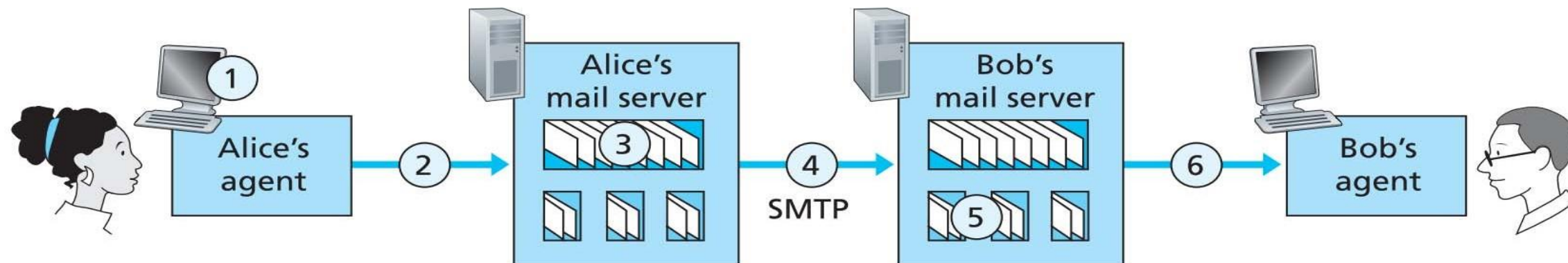


# Giao thức SMTP

- ❑ RFC 2821
- ❑ Sử dụng TCP để truyền tin cậy bản tin email từ client tới cổng 25 server
- ❑ Truyền trực tiếp: server gửi tới server nhận
- ❑ 3 giai đoạn truyền: bắt tay, truyền thông điệp, đóng
- ❑ giao tiếp kiểu lệnh/ phản hồi (giống HTTP, FTP)
  - **Lệnh:** văn bản ASCII
  - **Phản hồi:** mã trạng thái (status code) và cụm từ (phrase)
- ❑ Thông điệp phải ở dạng mã ASCII 7 bit

# Tình huống: Alice gửi thông điệp đến Bob

- 1) Alice dùng một UA để soạn thảo thông điệp “gửi đến” bob@some school.edu
- 2) UA của Alice gửi thông điệp đến mail server của cô ta; thông điệp được đặt trong hàng đợi
- 3) Phần client của SMTP server mở kết nối TCP với mail server của Bob
- 4) Phần client của SMTP server gửi thông điệp của Alice trên kết nối TCP
- 5) Mail server của Bob đặt thông điệp đó trong hộp thư của Bob
- 6) Bob kích hoạt user agent của anh ta để đọc thông điệp



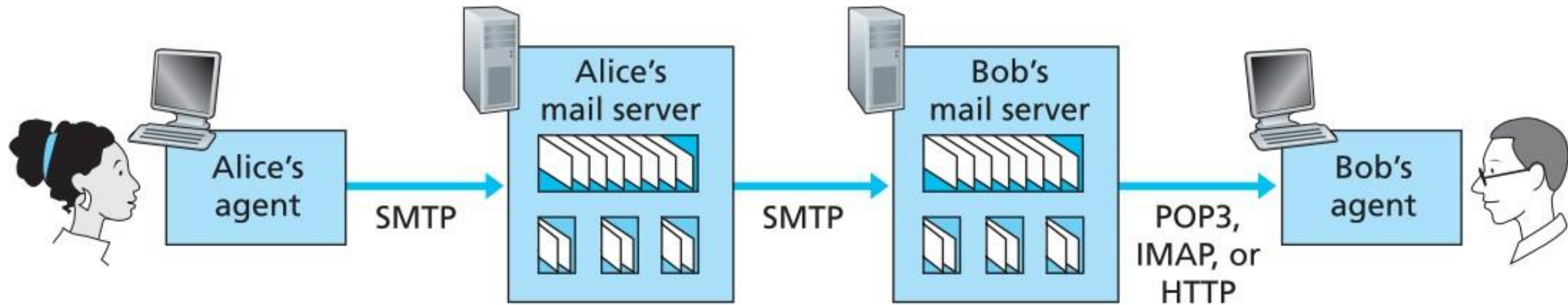
# SMTP

- ❑ SMTP sử dụng persistent connection
- ❑ SMTP yêu cầu bản tin (header & body) là 7-bit ASCII
- ❑ SMTP server sử dụng CRLF.CRLF để xác định kết thúc bản tin

## so sánh với HTTP:

- ❑ cùng giao tiếp dùng lệnh/phản hồi, mã trạng thái dạng ASCII
- ❑ HTTP: mỗi object chứa trong bản tin riêng
- ❑ SMTP: nhiều object được gửi trong một bản tin

# Giao thức truy cập thư điện tử



- ❑ **SMTP**: chuyển email tới server của người nhận
- ❑ Giao thức truy cập thư điện tử (mail access protocol): lấy email từ server
  - **POP** (Post Office Protocol [RFC 1939]): đăng nhập và lấy thư về
  - **IMAP** (Internet Mail Access Protocol [RFC 1730]): có nhiều đặc tính hơn, ví dụ thao tác với email trên server
  - **HTTP**: gmail, Hotmail, Yahoo Mail,...

# Chương 2: Tầng ứng dụng

1. Đặc điểm của ứng dụng mạng
2. Web và HTTP
3. Truyền tập tin: FTP
4. Thư điện tử
5. Hệ thống tên miền: DNS

# Hệ thống tên miền: DNS

- ❑ Con người: có nhiều cách để nhận dạng
  - ❑ Tên, số CMND
- ❑ Host, router trên Internet
  - ❑ Địa chỉ IP: ví dụ 203.113.135.55
  - ❑ Tên: ví dụ www.tlu.edu.vn
- ❑ Ánh xạ giữa địa chỉ IP và tên?
- ❑ DNS (domain name system)
  - ❑ Cơ sở dữ liệu phân tán được cài đặt trên một hệ thống phân cấp các máy phục vụ tên (name server)
  - ❑ Là giao thức tầng ứng dụng cho phép máy tính, máy chủ sử dụng để ánh xạ địa chỉ IP-tên

1. The same user machine runs the client side of the DNS application.
2. The browser extracts the hostname, `www.someschool.edu`, from the URL and passes the hostname to the client side of the DNS application.
3. The DNS client sends a query containing the hostname to a DNS server (UDP, port 53).
4. The DNS client eventually receives a reply, which includes the IP address for the hostname.
5. Once the browser receives the IP address from DNS, it can initiate a TCP connection to the HTTP server process located at port 80 at that IP address.



# Tổng quan về hoạt động của DNS

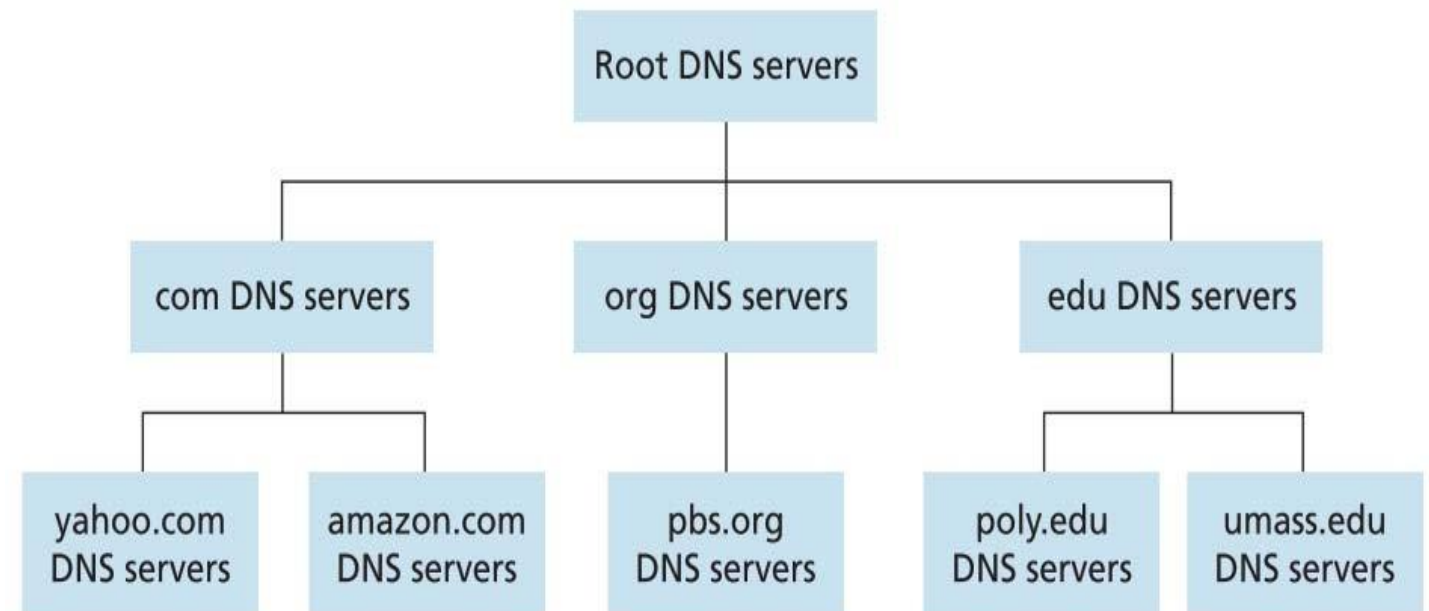
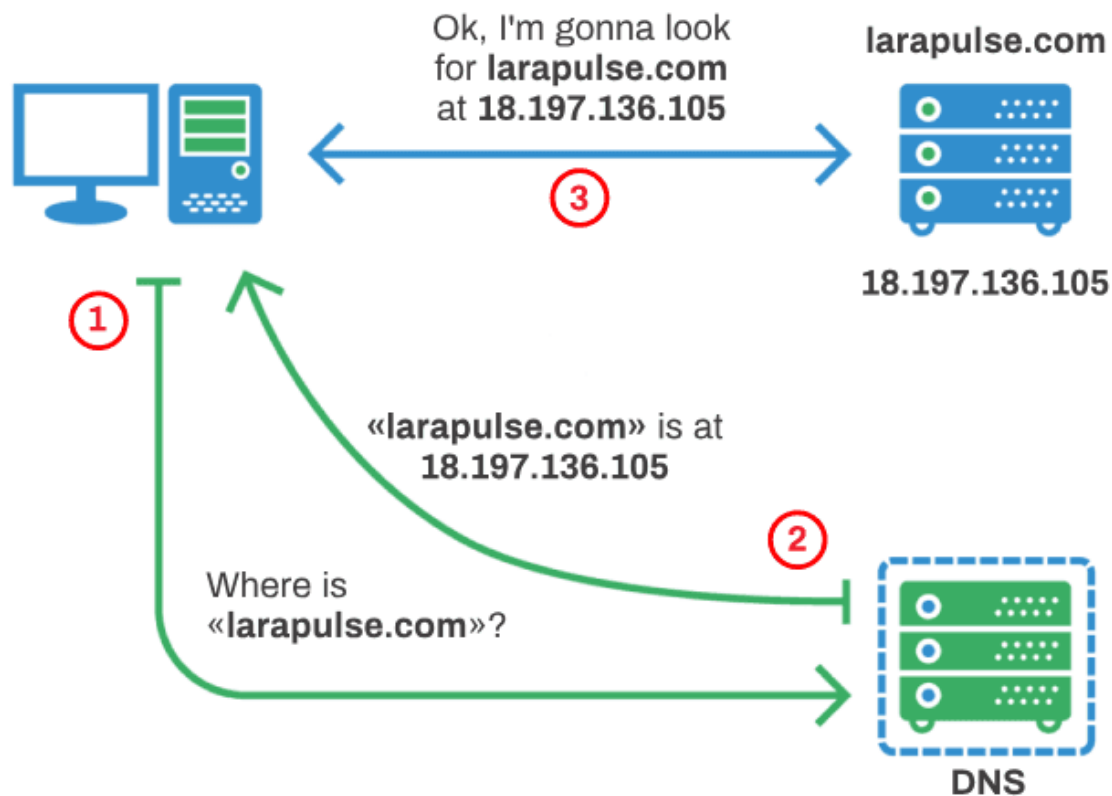
1. The same user machine runs the client side of the DNS application.

The browser extracts the hostname, `www.someschool.edu`, from the URL and passes the hostname to the client side of the DNS application.

The DNS client sends a query containing the hostname to a DNS server (UDP, port 53).

2. The DNS client eventually receives a reply, which includes the IP address for the hostname.

3. Once the browser receives the IP address from DNS, it can initiate a TCP connection to the HTTP server process located at port 80 at that IP address.



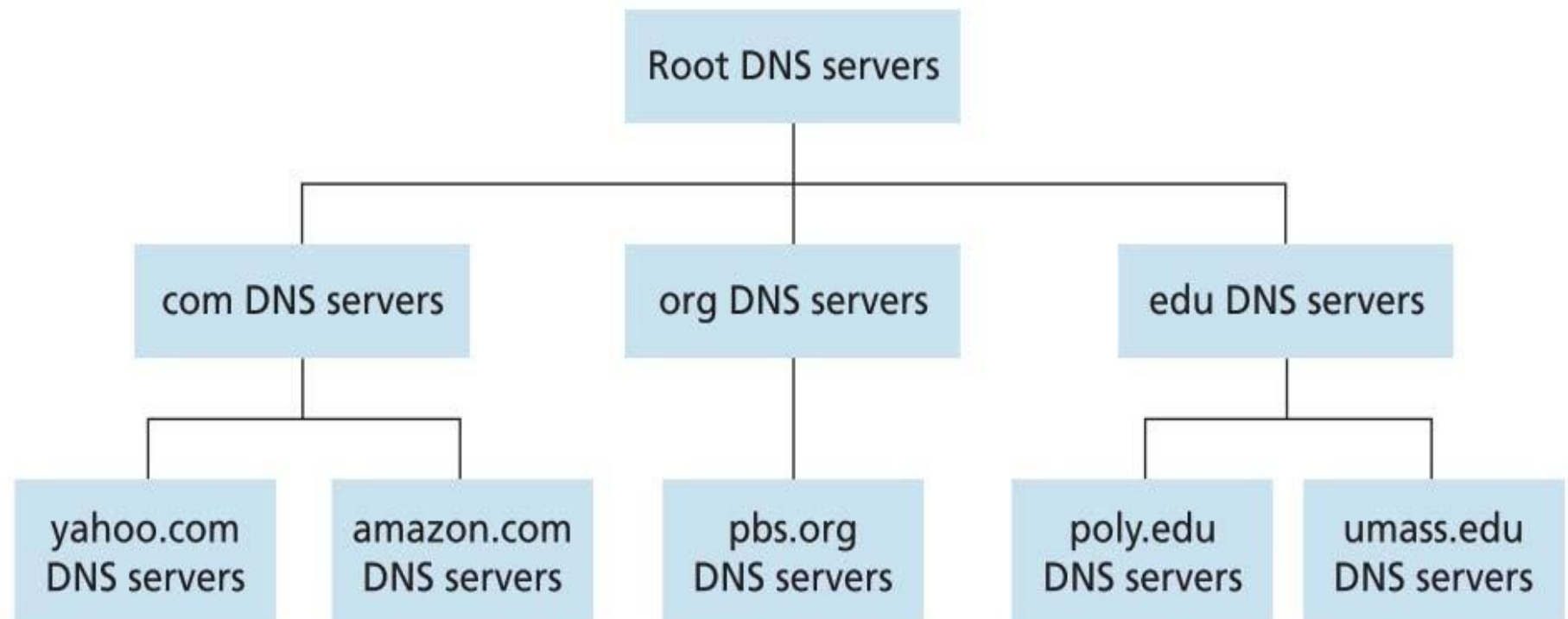


# Tổng quan về hoạt động của DNS

❑ DNS gồm nhiều server, theo mô hình **phân cấp** và **phân tán**

❑ Phân cấp: 3 cấp

1. Root DNS server: 13 địa điểm, hầu hết ở Bắc Mỹ.
2. Top-level domain DNS server (TLD server): com, org, net, edu, gov, uk, fr, ca, jp,...
3. Authoritative DNS server: Chứa bản ghi về IP các host (web server, mail server) của một tổ chức

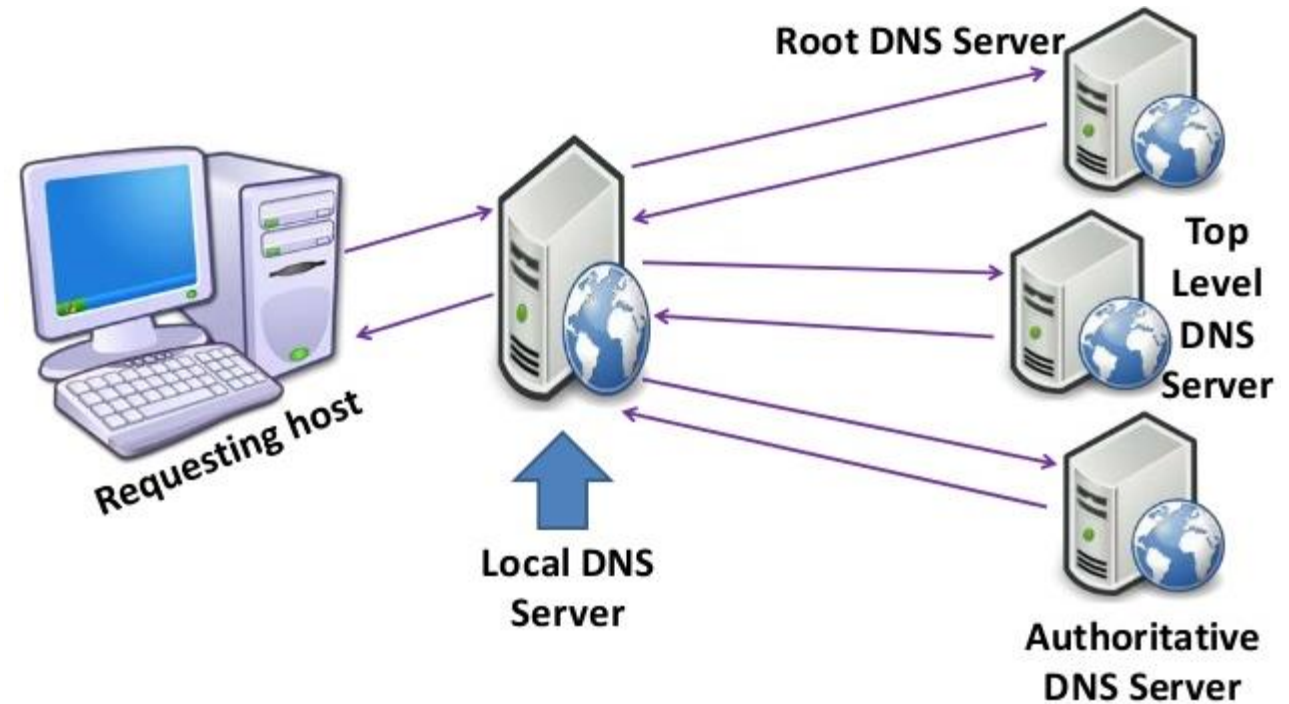


# Tổng quan về hoạt động của DNS

- Ngoài ra, để cải thiện hiệu quả của việc truy vấn DNS, các ISP còn sử dụng thêm một **Local DNS server – Máy chủ DNS địa phương**:

- **Máy chủ tên miền địa phương**

- Mỗi ISP như trường học, công ty hoặc ISP địa phương đều có thể có một local name server, còn được gọi là máy chủ tên miền mặc định (default name server)
- Khi host kết nối đến ISP, ISP gửi cho host một địa chỉ IP của local DNS server của nó



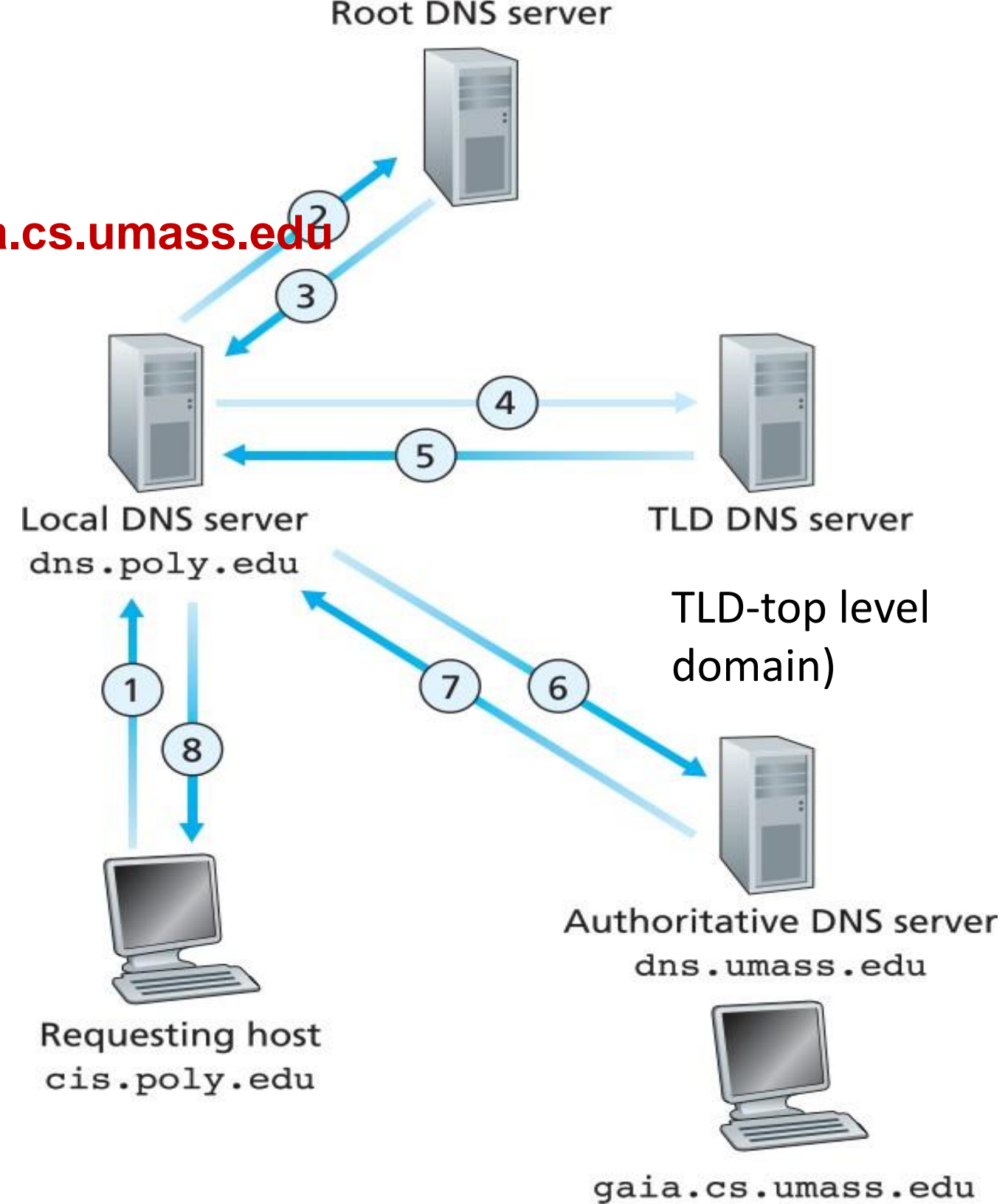
# Ví dụ truy vấn tên miền

**Máy trạm tại cis.poly.edu muốn địa chỉ IP của gaia.cs.umass.edu**

- 2 cách: tuần tự và đệ quy

## Truy vấn tuần tự

1. Host gửi yêu cầu truy vấn đến Local DNS server
2. Local DNS server gửi truy vấn đến **Root DNS server**
3. **Root DNS server** trả lại danh sách địa chỉ IP của các TLD DNS server của tên miền .edu
4. Local DNS server gửi truy vấn đến **TLD DNS server (.edu)**
5. **TLD DNS server (.edu)** xác định địa chỉ IP của authoritative DNS server (umass.edu)
6. Local DNS server gửi truy vấn đến **authoritative DNS server (umass.edu)**
7. **Authoritative DNS server (umass.edu)** xác định địa chỉ IP của **gaia.cs.umass.edu** và gửi lại cho Local DNS server
8. Local DNS server gửi ánh xạ địa chỉ IP cho host



# Ví dụ truy vấn tên miền (tiếp)

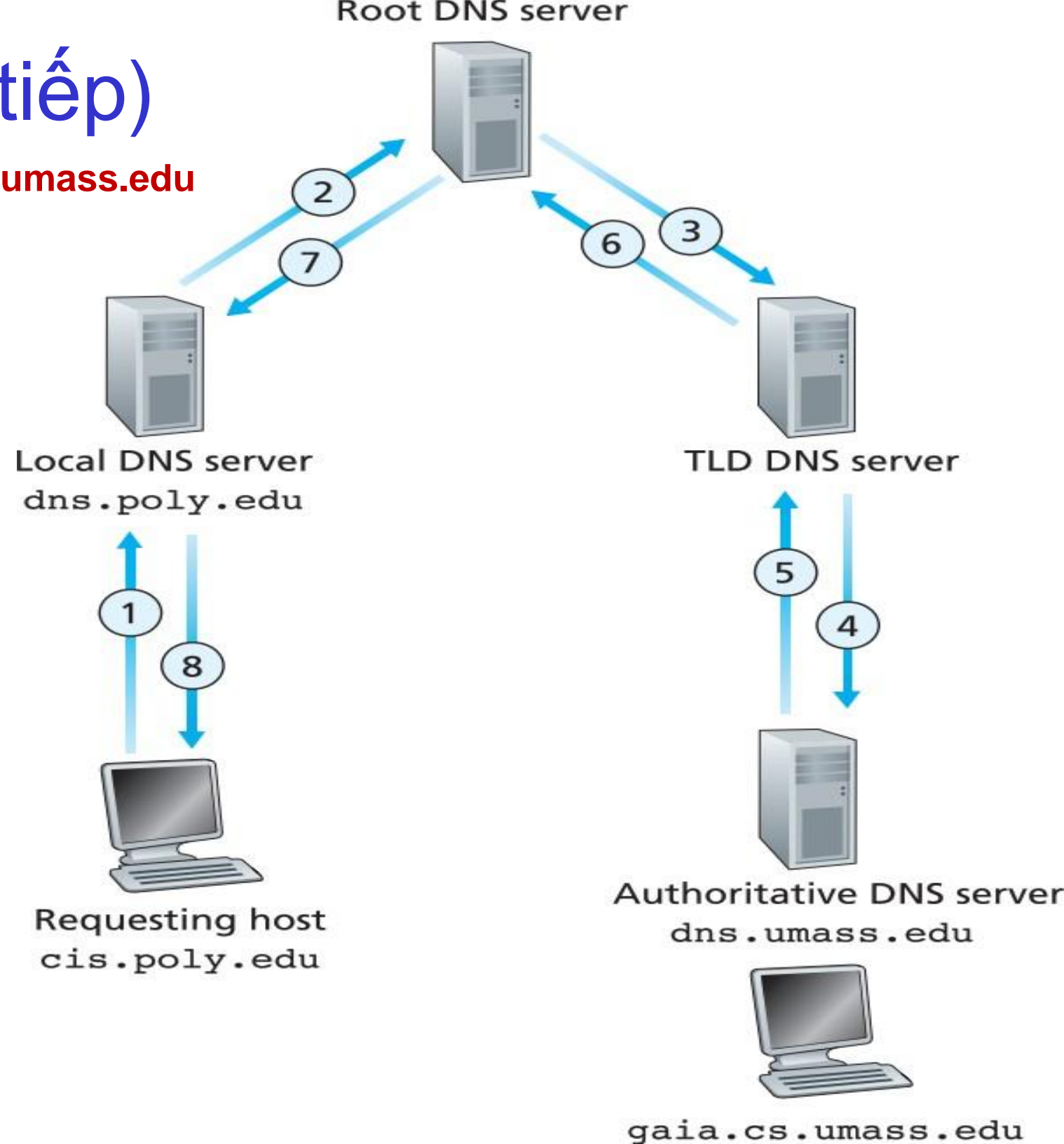
Máy trạm tại **cis.poly.edu** muốn địa chỉ IP của **gaia.cs.umass.edu**

## Truy vấn đệ quy

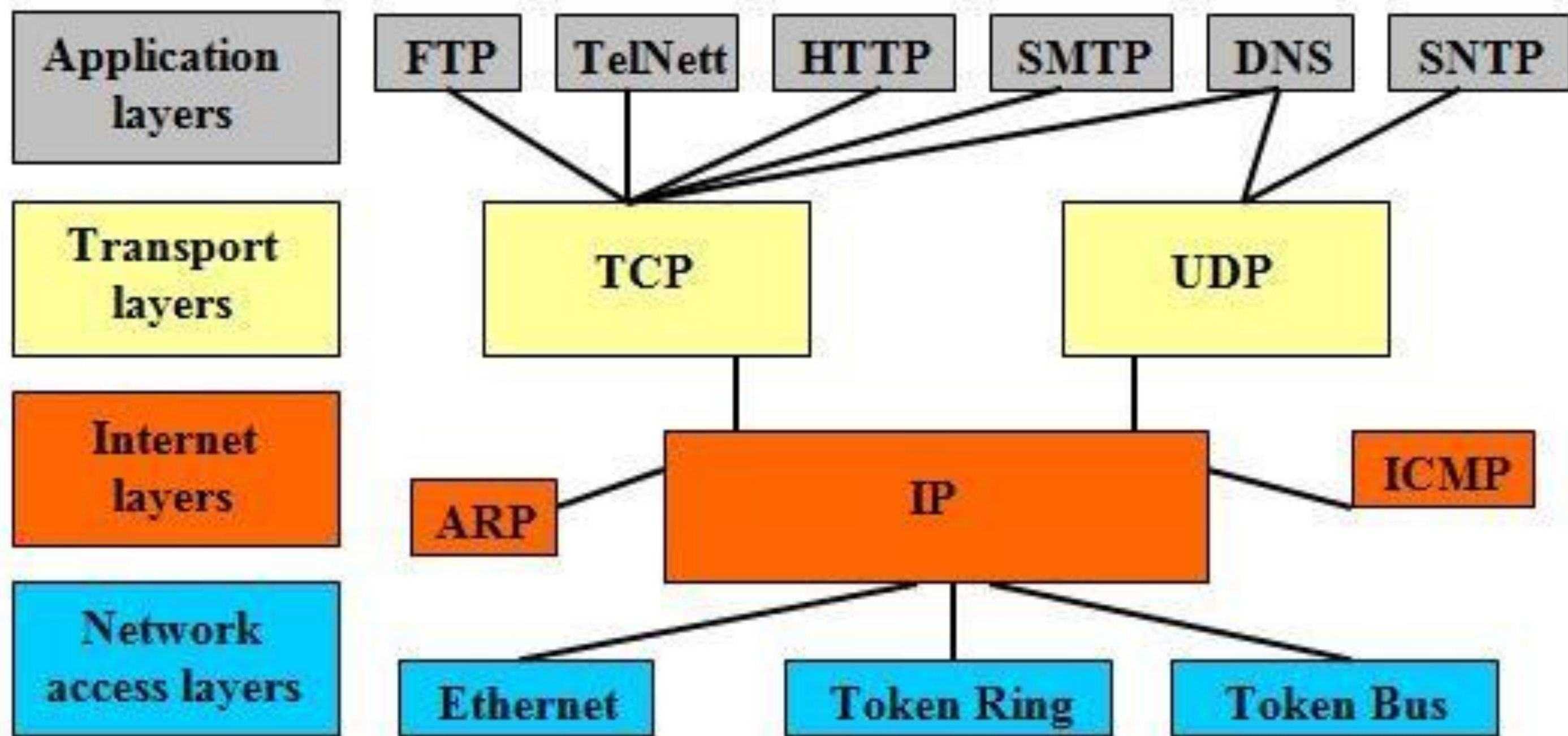
- ❑ Việc truy vấn sẽ được thực hiện bởi các DNS server
- ❑ Không thường được sử dụng trong thực tế

## DNS caching – Bộ nhớ đệm DNS

- ❑ Các DNS server trong hệ thống DNS sẽ lưu thông tin truy vấn được trong một khoảng thời gian nhất định (thường là 2 ngày) để giảm lưu lượng truy vấn DNS và tăng hiệu quả hệ thống







# Tóm tắt

- ❑ Kiến trúc ứng dụng
  - Client-server
  - P2P
- ❑ Yêu cầu dịch vụ ứng dụng:
  - Truyền tin cậy, độ trễ
- ❑ Mô hình dịch vụ giao vận của Internet
  - Tin cậy: TCP
  - không tin cậy: UDP
- ❑ Giao thức:
  - HTTP
  - FTP
  - SMTP, POP, IMAP
  - DNS

# Mạng máy tính

- Hình ảnh và nội dung trong bài giảng này có tham khảo từ sách và bài giảng của TS. J.F. Kurose and GS. K.W. Ross

# Bài tập

## 2.1. Bài 1

*Cho chuỗi các kí tự mà phần mềm Wireshark đã ghi lại khi trình duyệt gửi bản tin HTTP GET. Kí tự <cr><lf> là kí tự về đầu dòng và xuống dòng*

**GET /cs453/index.html HTTP/1.1<cr><lf> Host: gaia.cs.umass.edu<cr><lf> User-Agent: Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gecko/20040804 Netscape/7.2 (ax)<cr><lf> Accept::ext/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) Xác định URL của văn bản mà trình duyệt đã yêu cầu?*
- b) Chỉ ra phiên bản của HTTP mà trình duyệt đang dùng?*
- c) Trình duyệt yêu cầu kết nối kiểu non-persistent hay persistent?*
- d) Xác định địa chỉ IP của nút mạng mà trình duyệt đang chạy?*



## 2.2. Bài 2

*Đoạn văn bản dưới là phản hồi của server khi nhận được gói tin HTTP GET trong bài 1.*

**HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008 12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46 GMT<cr><lf>ETag: “526c3-f22-a88a4c80”<cr><lf>Accept-Ranges: bytes<cr><lf>Content-Length: 3874<cr><lf>Keep-Alive: timeout=max=100<cr><lf>Connection: Keep-Alive<cr><lf>Content-Type: text/html; charset=ISO-8859- 1<cr><lf><cr><lf><!doctype html public “-//w3c//dtd html 4.0 transitional//en”><lf><html><lf><head><lf> <meta http-equiv=”Content-Type” content=”text/html; charset=iso-8859-1”><lf> <metaname=”GENERATOR” content=”Mozilla/4.79 [en] (Windows NT 5.0; U Netscape)”><lf> <title>CMPSCI 453 / 591 / NTU-ST550A Spring 2005 homepage</title><lf></head><lf> <bỏ qua các phần sau của gói tin>**

- a) Server tìm thấy văn bản hay không?*
- b) Xác định thời gian mà văn bản trả lời được cung cấp?*
- c) Xác định thời gian cuối cùng mà văn bản được sửa đổi?*
- d) Xác định số byte trong văn bản trả lời?*
- e) Xác định 5 byte đầu tiên trong văn bản trả lời?*
- f) Có phải server chấp nhận kiểu kết nối persistent?*