

Trường Đại Học Bách Khoa Tp.HCM  
Hệ Đào Tạo Từ Xa  
Khoa Khoa Học và Kỹ Thuật Máy Tính

---

Mạng máy tính căn bản

# Bài giảng 3: Tầng ứng dụng

---

## **Tham khảo:**

Chương 2: “Computer Networking – A top-down approach”  
Kurose & Ross, 5<sup>th</sup> ed., Addison Wesley, 2010.

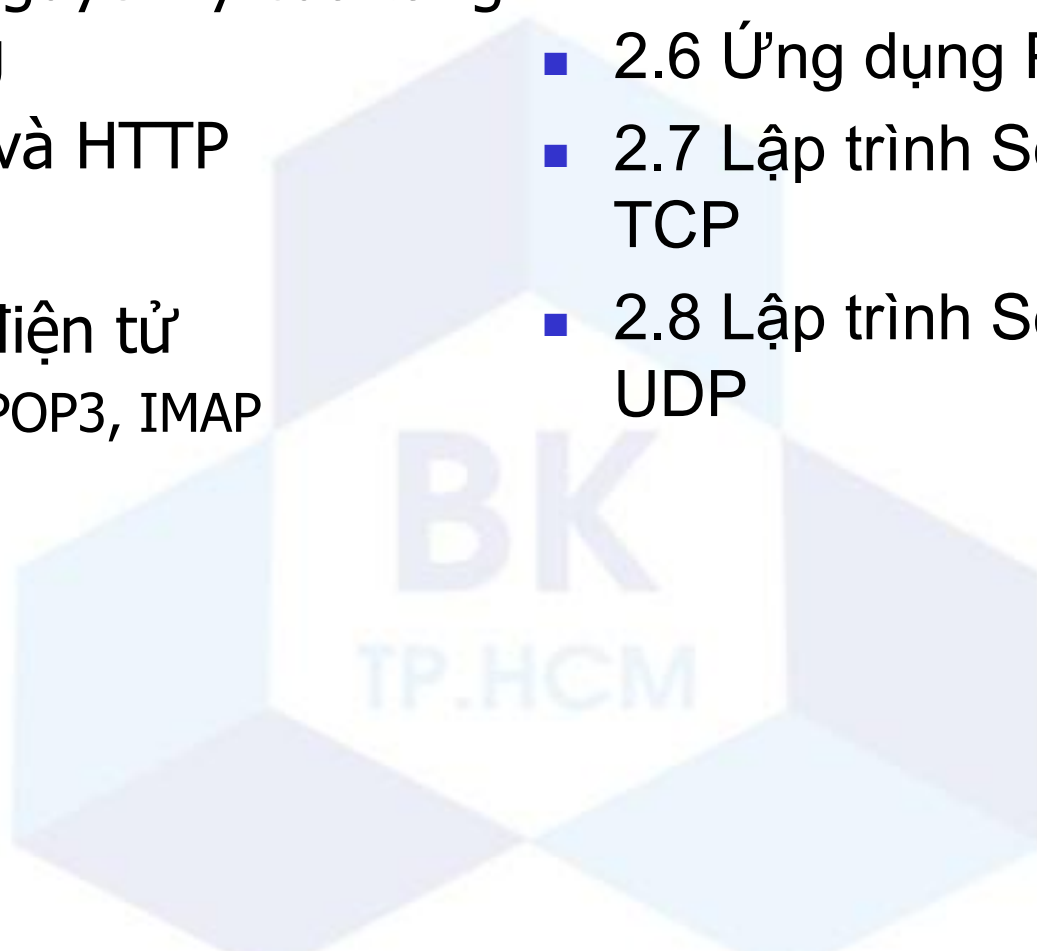
# Chapter 2: Tầng ứng dụng

## Mục tiêu:

- các khía cạnh thuộc khái niệm và hiện thực của các giao thức ứng dụng mạng
  - mô hình dịch vụ tầng ứng dụng
  - mô hình khách-chủ
  - mô hình ngang hàng một-một
- Tìm hiểu giao thức bằng cách xem xét các giao thức tầng ứng dụng phổ biến
  - HTTP
  - FTP
  - SMTP / POP3 / IMAP
  - DNS
- Lập trình ứng dụng mạng
  - socket API
  - socket = hốc kết nối?

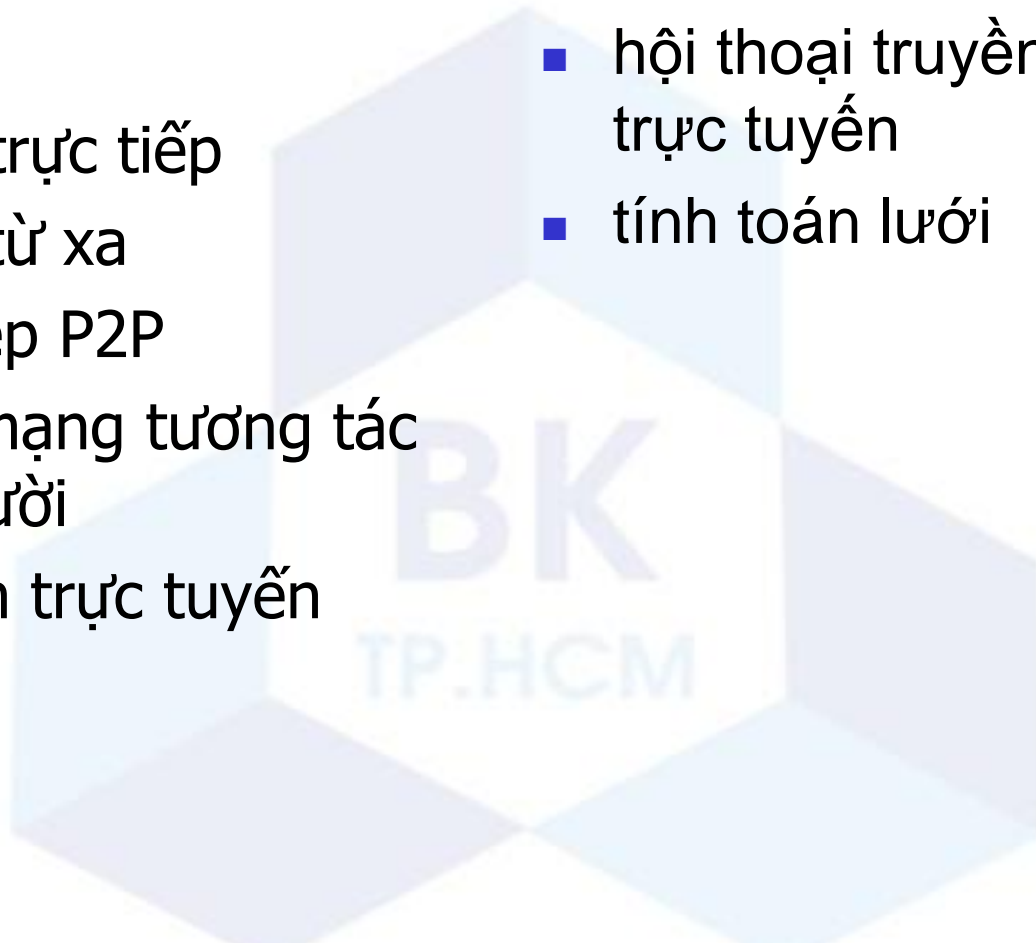
# Chương 2: Tầng ứng dụng

- 2.1 Các nguyên lý của tầng ứng dụng
- 2.2 Web và HTTP
- 2.3 FTP
- 2.4 Thư điện tử
  - SMTP, POP3, IMAP
- 2.5 DNS
- 2.6 Ứng dụng P2P
- 2.7 Lập trình Socket với TCP
- 2.8 Lập trình Socket với UDP



# Ví dụ về ứng dụng mạng

- e-mail
- web
- nhắn tin trực tiếp
- truy cập từ xa
- chia sẻ tệp P2P
- trò chơi mạng tương tác nhiều người
- xem phim trực tuyến
- chat tiếng
- hội thoại truyền hình trực tuyến
- tính toán lưới



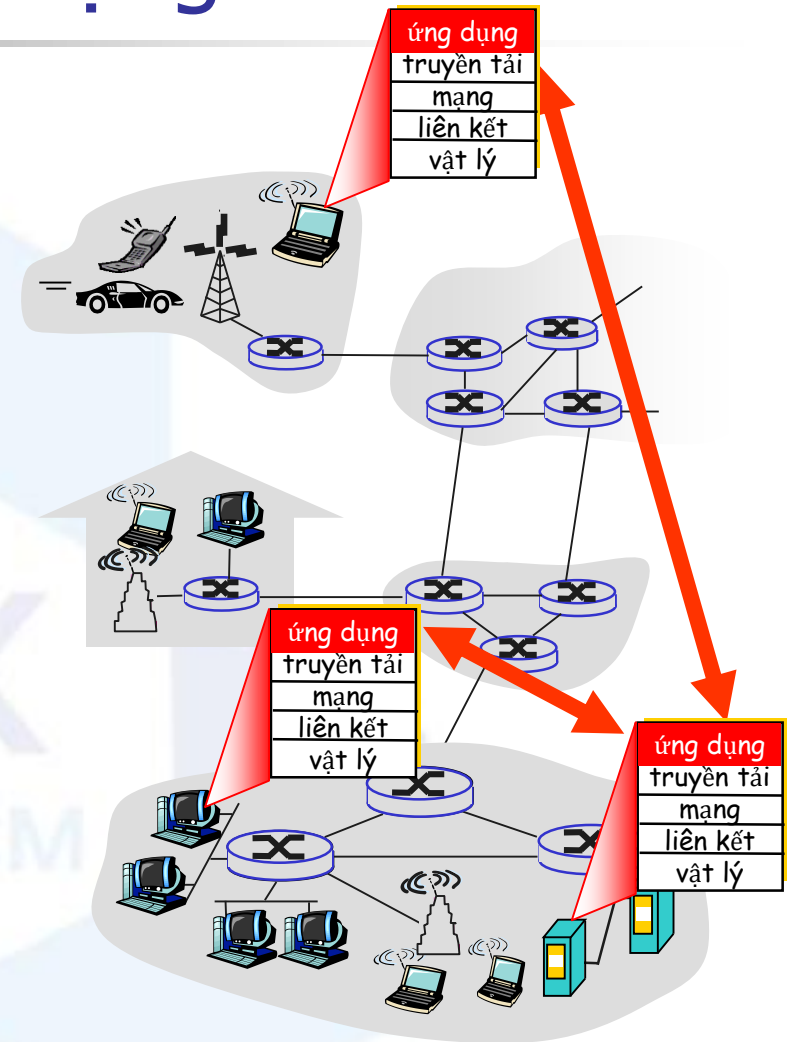
# Tạo ra một ứng dụng mạng

## Viết chương trình

- chạy trên những hệ thống khác nhau
- liên lạc thông qua mạng
- vd: phần mềm máy chủ web giao tiếp với phần mềm trình duyệt web

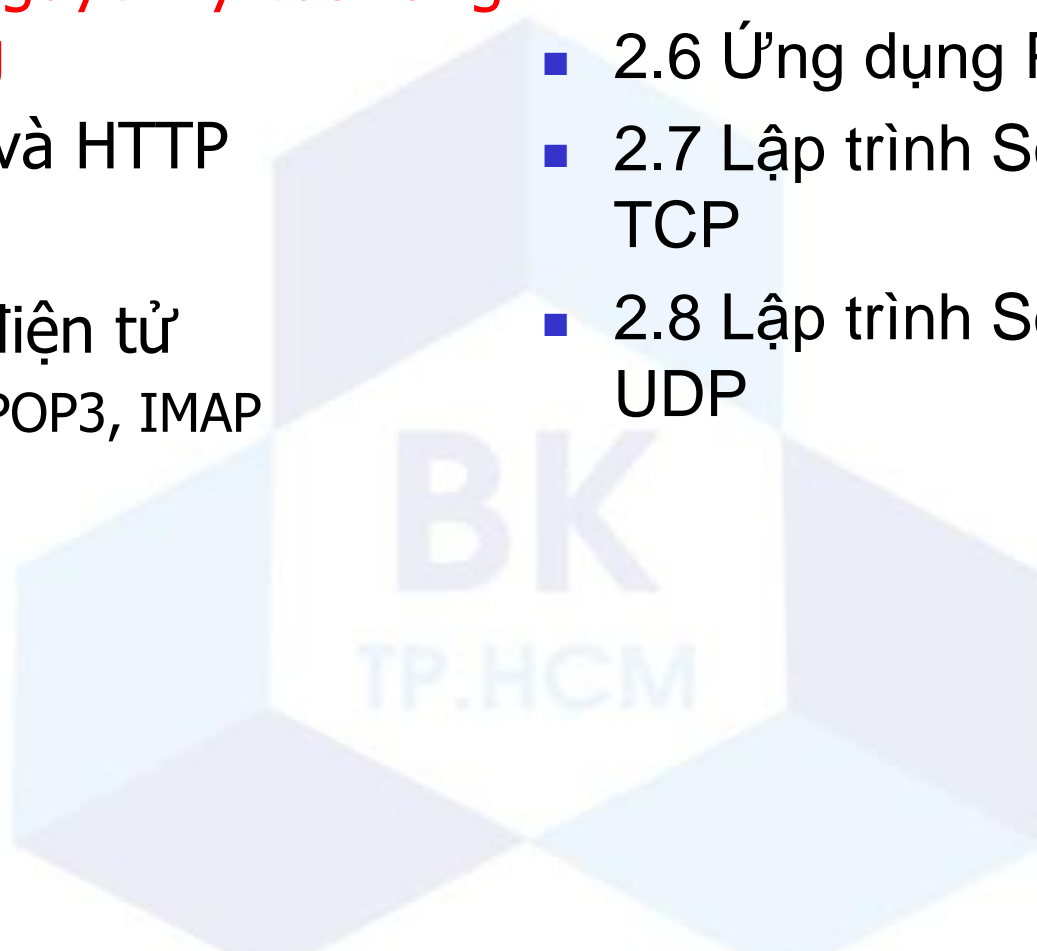
## Không cần thiết viết ph/mềm cho các thiết bị mạng (TBM)

- TBM không chạy những ứng dụng dành cho người dùng
- Phần mềm trên các TBM được chuyên biệt hóa cao



# Chương 2: Tầng ứng dụng

- 2.1 Các nguyên lý của tầng ứng dụng
- 2.2 Web và HTTP
- 2.3 FTP
- 2.4 Thư điện tử
  - SMTP, POP3, IMAP
- 2.5 DNS
- 2.6 Ứng dụng P2P
- 2.7 Lập trình Socket với TCP
- 2.8 Lập trình Socket với UDP



# Kiến trúc ứng dụng mạng

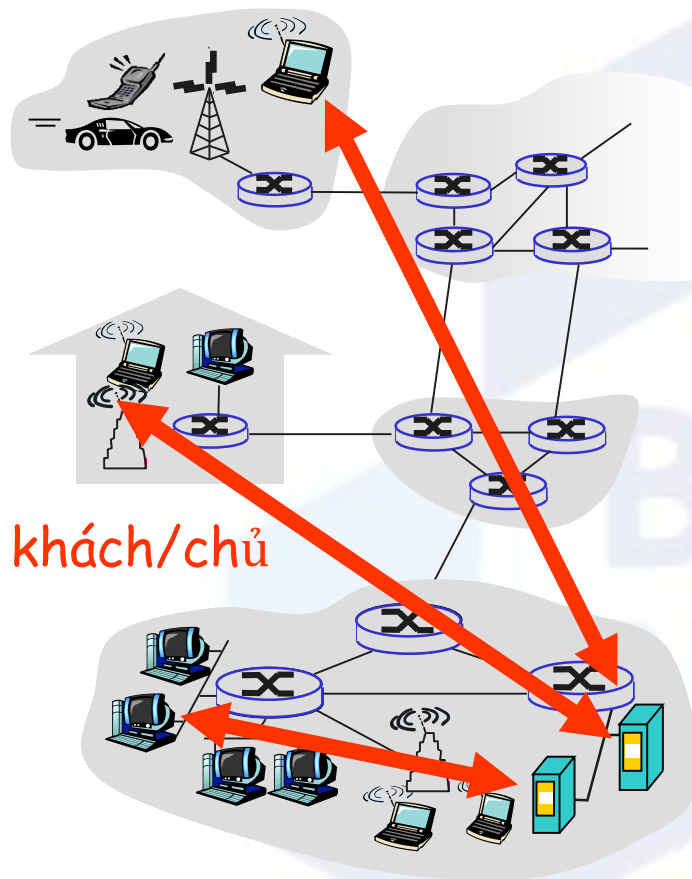
---

- khách-chủ
- thành viên-tới-thành viên (P2P)
- lai giữa khách-chủ và P2P





# Kiến trúc khách-chủ



## máy chủ:

- luôn luôn bật
- địa chỉ IP cố định
- hệ thống máy chủ cho việc mở rộng

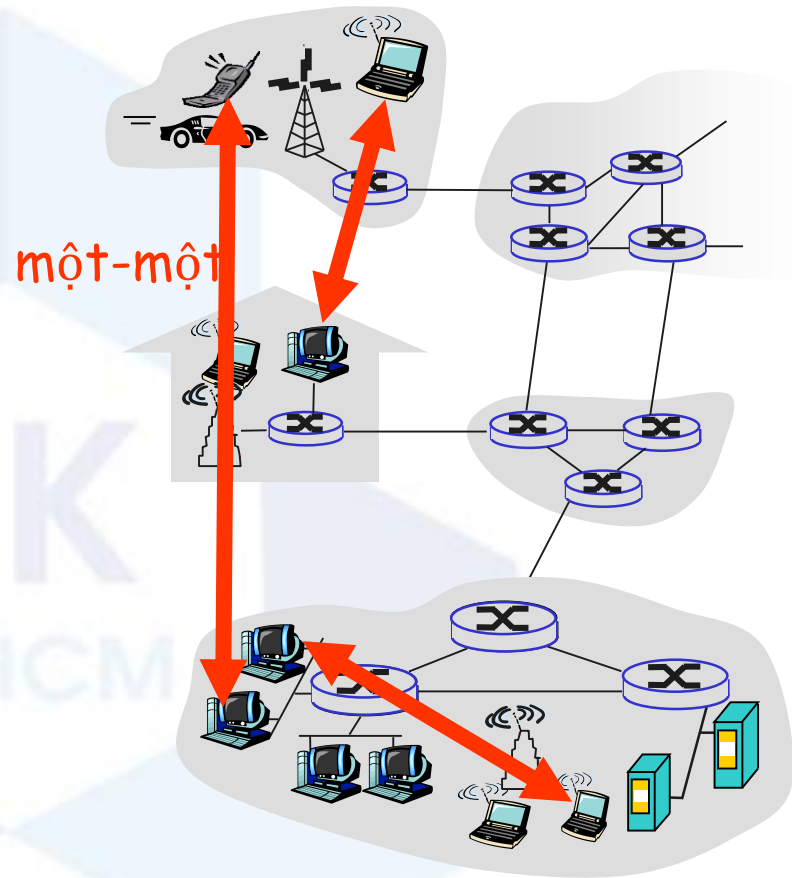
## máy khách:

- giao tiếp với máy chủ
- có thể kết nối không liên tục
- có thể có địa chỉ IP động
- không kết nối trực tiếp với người dùng khác

# Kiến trúc P2P

- không có máy chủ tập trung
- các máy đầu cuối liên lạc trực tiếp với nhau
- các bên thực hiện kết nối không liên tục và có thể sử dụng địa chỉ IP động

Khả năng co giãn (mở rộng)  
cao nhưng khó quản lý



# Mô hình lai của khách-chủ và P2P

## Skype

- điện thoại-IP: ứng dụng P2P
- máy chủ tập trung: quản lý địa chỉ của các bên tham gia
- kết nối khách-khách: trực tiếp (không qua máy chủ)

## Nhắn tin trực tiếp

- Hai người dùng chat với nhau bằng P2P
- Máy chủ đảm bảo: quản lý người dùng
  - người dùng đăng kí đ/c IP với máy chủ khi mà họ đăng nhập
  - máy chủ cung cấp danh sách bạn bè đang trên mạng
  - người dùng hỏi máy chủ để có đ/c IP của bạn

# Liên lạc giữa các tiến trình

**Tiến trình:** chương trình chạy trong một máy

- với cùng một máy, hai tiến trình liên lạc với nhau bằng cách sử dụng **giao tiếp tiến trình (inter-process communication)**.
- các tiến trình trong các máy khác nhau liên lạc thông qua trao đổi **thông điệp**

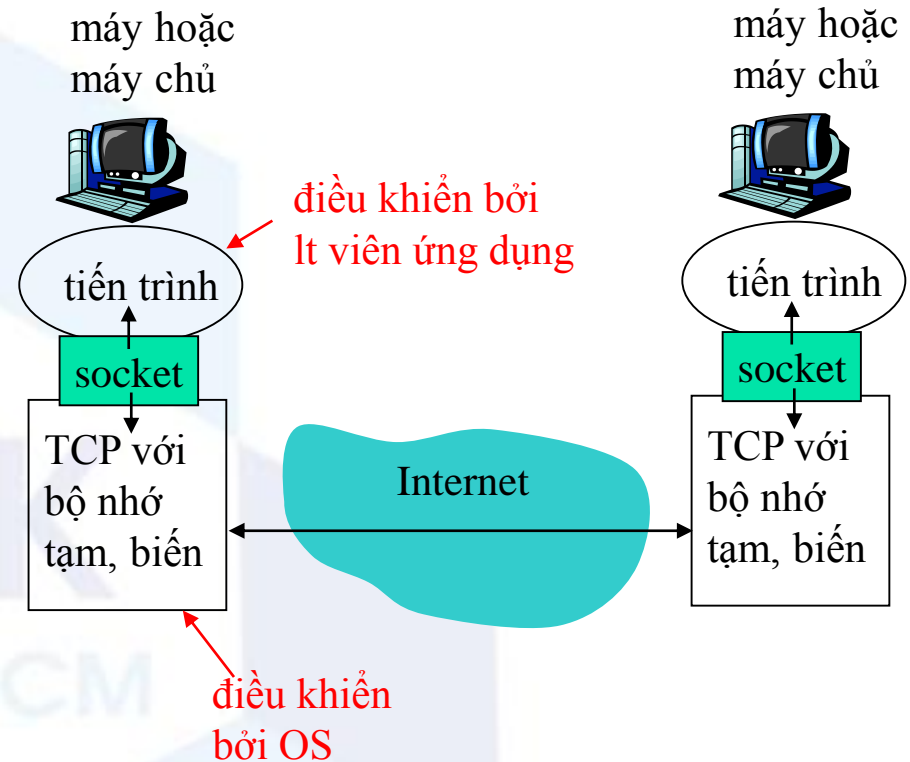
**Tiến trình khách:** bắt đầu quá trình kết nối

**Tiến trình chủ:** chờ khách kết nối tới

Lưu ý: các ứng dụng P2P có cả tiến trình khách lẫn tiến trình chủ

# Hỗ kết nối - Sockets

- tiến trình gửi/nhận thông điệp tới/từ **hỗ kết nối**
- hỗ kết nối tương tự như là cửa
  - tiến trình gửi đẩy thông điệp ra ngoài cửa
  - tiến trình gửi phụ thuộc vào cơ sở hạ tầng truyền tải ở bên phía kia cửa để mang thông điệp tới hỗ kết nối của tiến trình nhận
- API: (1) lựa chọn giao thức truyền tải; (2) khả năng một vài tham số



# Quá trình đánh địa chỉ

- để nhận được thông điệp tiến trình phải có *kí hiệu nhận dạng*
- mỗi thiết bị (máy tính) có một địa chỉ IP 32-bit độc nhất
- Hỏi: địa chỉ IP có thể dùng để đánh cho cả các tiến trình không?
  - Đáp: Không, *nhiều* tiến trình có thể cùng chạy trên một máy
- *kí hiệu nhận dạng* bao gồm **địa chỉ IP** và **số cổng** tương ứng với mỗi tiến trình trên máy
- ví dụ về số cổng:
  - máy chủ HTTP: 80
  - máy chủ Mail: 25
- để gửi thông điệp HTTP tới trang web gaia.cs.umass.edu:
  - **địa chỉ IP**: 128.119.245.12
  - **cổng**: 80

# Các giao thức tầng ứng dụng qui định...

- Loại thông điệp được trao đổi,
  - vd: truy vấn, phản hồi
- Cú pháp của thông điệp:
  - có những trường nào trong thông điệp & chúng được trình bày ntn
- Ngữ nghĩa của thông điệp
  - ý nghĩa của thông tin trong các trường
- Qui tắc về thời điểm và cách thức các tiến trình gửi và phản hồi các thông điệp

## Các giao thức công khai:

- định nghĩa trong RFC
- cho phép tương tác qua lại
- vd: HTTP, SMTP

## Các giao thức sở hữu:

- vd: Skype

# Các ứng dụng yêu cầu gì ở tầng truyền tải?

## Mất dữ liệu

- một số ứng dụng (voice-chat) có thể chấp nhận mất mát dữ liệu vừa phải
- những Ứ/D khác (vd: tải tệp tin, telnet) yêu cầu truyền tải toàn bộ 100% dữ liệu

## Độ trễ

- các Ứ/D ,vd: điện thoại Internet, trò chơi tương tác, yêu cầu độ trễ thấp

## Thông lượng

- một số Ứ/D (vd: đa phương tiện) yêu cầu một thông lượng tối thiểu để làm việc "hiệu quả"
- những Ứ/D khác (bittorrent) tận dụng mọi thông lượng mà nó có thể sử dụng

## Bảo mật

- Mã hóa, vệt toàn dữ liệu, ...



# Những yêu cầu lên tầng vận chuyển của một số ứng dụng phổ biến

Ứng dụng	Mất mát dl	Thông lượng	Độ trễ thấp
truyền tải tệp	không	linh hoạt	không
e-mail	không	linh hoạt	không
dịch vụ web	không	linh hoạt	không
phim/nhạc thời gian thực	chấp nhận	nhạc: 5kbps-1Mbps phim: 10kbps-5Mbps	có, 100's msec
phim/nhạc lưu trữ	chấp nhận	như trên	có, vài giây
trò chơi tương tác	chấp nhận	vài kbps trở lên	có, 100's msec
nhắn tin trực tiếp	không	linh hoạt	vừa phải

# Các dịch vụ tăng truyền tải Internet

## Dịch vụ TCP:

- *hướng kết nối:* yêu cầu tạo kết nối giữa khách và chủ
- *truyền tải tin cậy:* đảm bảo truyền tải đúng và đủ dữ liệu
- *kiểm soát lưu lượng:* người gửi không làm ngập người nhận
- *kiểm soát tắc nghẽn:* hãm tốc độ gửi khi mạng bị quá tải
- *không cung cấp/đảm bảo:* độ trễ, thông lượng tối thiểu, bảo mật

## Dịch vụ UDP:

- truyền tải dữ liệu không tin cậy (có thể mất dữ liệu)
- không cung cấp: thiết lập kết nối, kiểm soát lưu lượng, kiểm soát tắc nghẽn, độ trễ, thông lượng tối thiểu, hoặc bảo mật

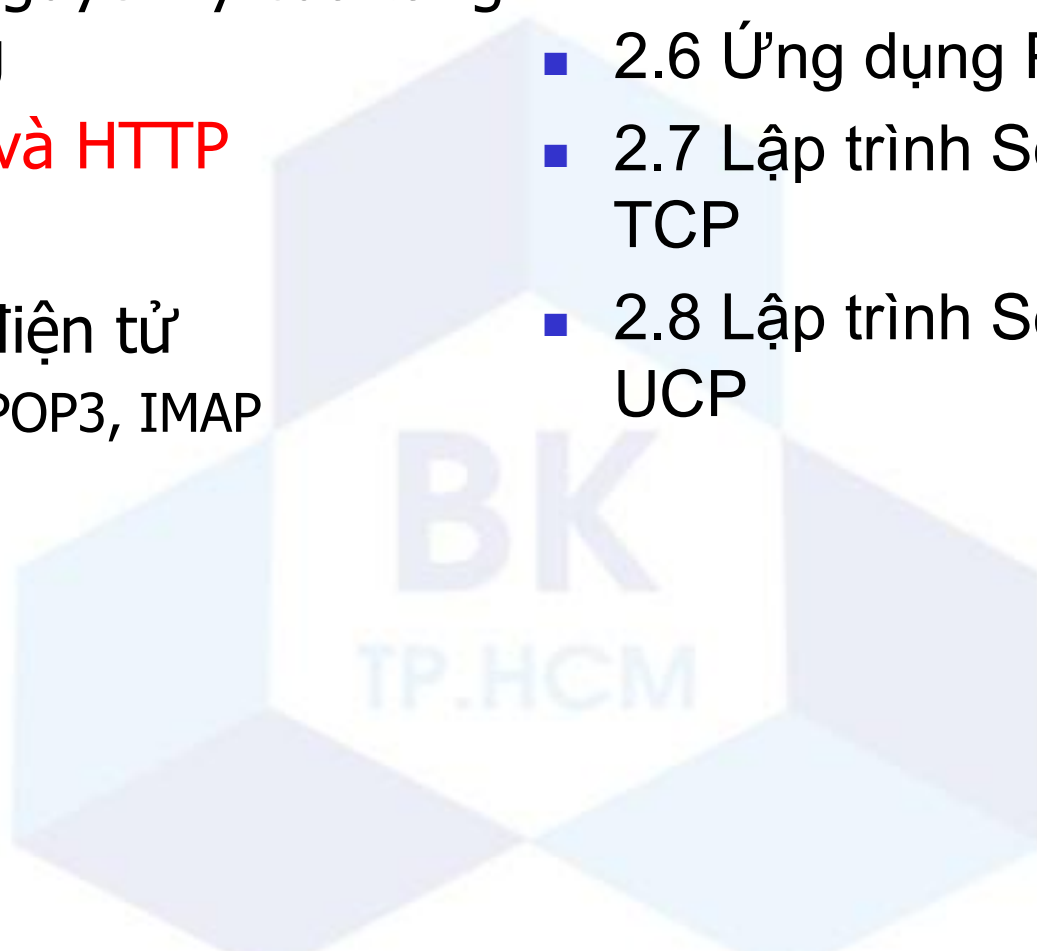
Hỏi: Tại sao lại cần có UDP?

# Ứ/d Internet và các giao thức tầng truyền tải

Ứng dụng	Giao thức tầng ứng dụng	Giao thức tầng truyền tải t/ứng
e-mail	SMTP [RFC 2821]	TCP
truy cập từ xa	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
truyền tải tệp	FTP [RFC 959]	TCP
xem phim trực tuyến	HTTP (vd: Youtube), RTP [RFC 1889]	TCP or UDP
điện thoại Internet	SIP, RTP, sở hữu (vd: Skype)	chủ yếu UDP

# Chương 2: Tầng ứng dụng

- 2.1 Các nguyên lý của tầng ứng dụng
- 2.2 Web và HTTP
- 2.3 FTP
- 2.4 Thư điện tử
  - SMTP, POP3, IMAP
- 2.5 DNS
- 2.6 Ứng dụng P2P
- 2.7 Lập trình Socket với TCP
- 2.8 Lập trình Socket với UCP



# Web và HTTP

## Vài khái niệm căn bản

- Trang Web chứa nhiều đối tượng
- Đối tượng có thể là tệp HTML, tệp ảnh JPEG, vi mã Java, tệp âm thanh,...
- Trang web chứa tệp HTML chính, tệp này sẽ chứa liên kết tới các đối tượng khác
- Mỗi đối tượng được dẫn đường bởi một URL – định vị tài nguyên đồng dạng
- ví dụ URL:

`www.someschool.edu/someDept/pic.gif`

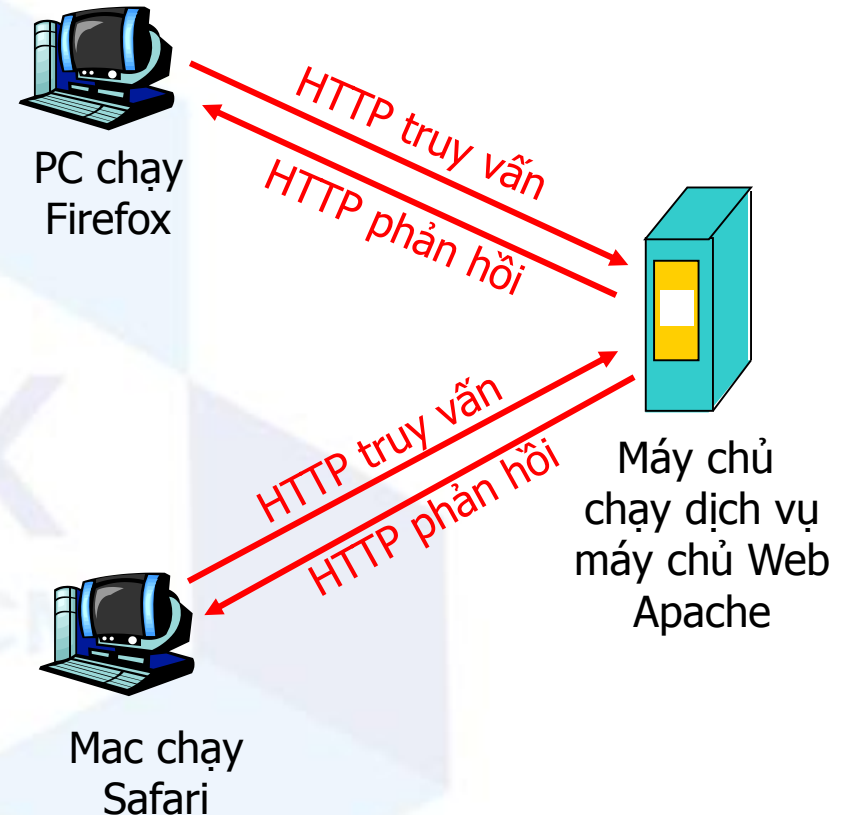
tên miền

đường dẫn

# Tóm lược HTTP

## Giao thức truyền tải siêu văn bản – HTTP (hypertext transfer protocol)

- giao thức ứng dụng web
- mô hình khách/chủ
  - **khách**: trình duyệt web yêu cầu, nhận và trình bày các đối tượng web
  - **chủ**: máy chủ Web gửi trả các đối tượng theo yêu cầu của khách



# Tóm lược HTTP (tt)

## Sử dụng TCP:

- khách khởi tạo kết nối TCP (tạo hốc kết nối) tới máy chủ tại cổng 80
- máy chủ chấp nhận kết nối TCP từ khách
- thông điệp HTTP (thông điệp giao thức tầng ứng dụng) được trao đổi giữa trình duyệt web (khách HTTP) và máy chủ Web (chủ HTTP)
- đóng kết nối TCP

## HTTP là giao thức không lưu giữ trạng thái

- máy chủ không lưu bất kỳ thông tin nào về ng/d

## Những giao thức lưu trạng thái tương đối phức tạp hơn nữa

- ☐ lịch sử duyệt web phải được lưu giữ
- ☐ nếu máy chủ/khách bị hỏng thì thông tin về trạng thái có thể không thống nhất, cần phải cân chỉnh lại

# Kết nối HTTP

## HTTP ngắn hạn (Non-Persistent)

- Gần như một đối tượng được gửi qua MỘT kết nối TCP

## HTTP dài hạn (Persistent)

- Nhiều đối tượng có thể được gửi qua một kết nối TCP duy nhất giữa khách và chủ





# HTTP không ổn định

người dùng đưa vào URL sau

`www.someSchool.edu/someDepartment/home.index`

(chứa văn bản,  
liên kết tới 10  
hình ảnh jpeg)

1a. khách HTTP khởi tạo liên kết TCP tới máy chủ HTTP (tiến trình) tại `www.someSchool.edu` ở cổng 80

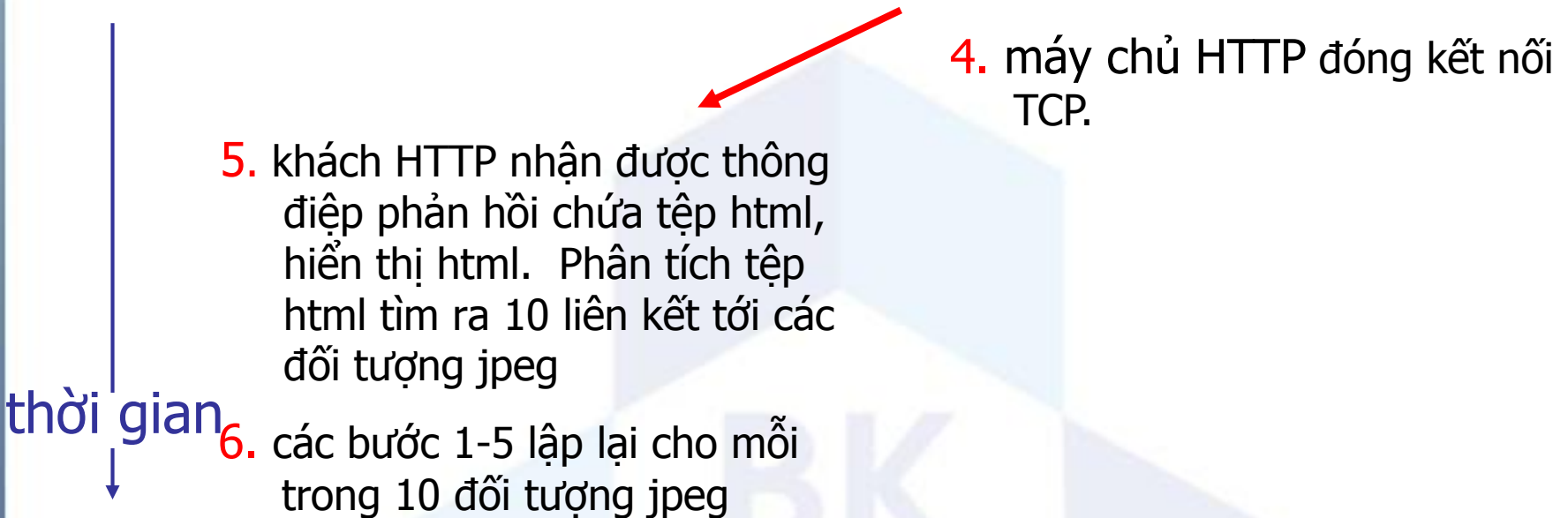
1b. máy chủ HTTP tại `www.someSchool.edu` chờ kết nối TCP tại cổng 80. “chấp nhận” kết nối từ khách, thông báo lại cho khách

2. khách HTTP gửi *thông điệp truy vấn* HTTP (chứa URL) vào hốc kết nối TCP. Thông điệp chỉ ra là khách muốn có đối tượng “`someDepartment/home.index`”

3. máy chủ HTTP server nhận được thông điệp truy vấn, tạo ra một *thông điệp phản hồi* chứa đối tượng được yêu cầu, và gửi thông điệp vào hốc kết nối

thời gian

# HTTP không ổn định(tt)



# HTTP ngắn hạn (non-persistent): thời gian phản hồi

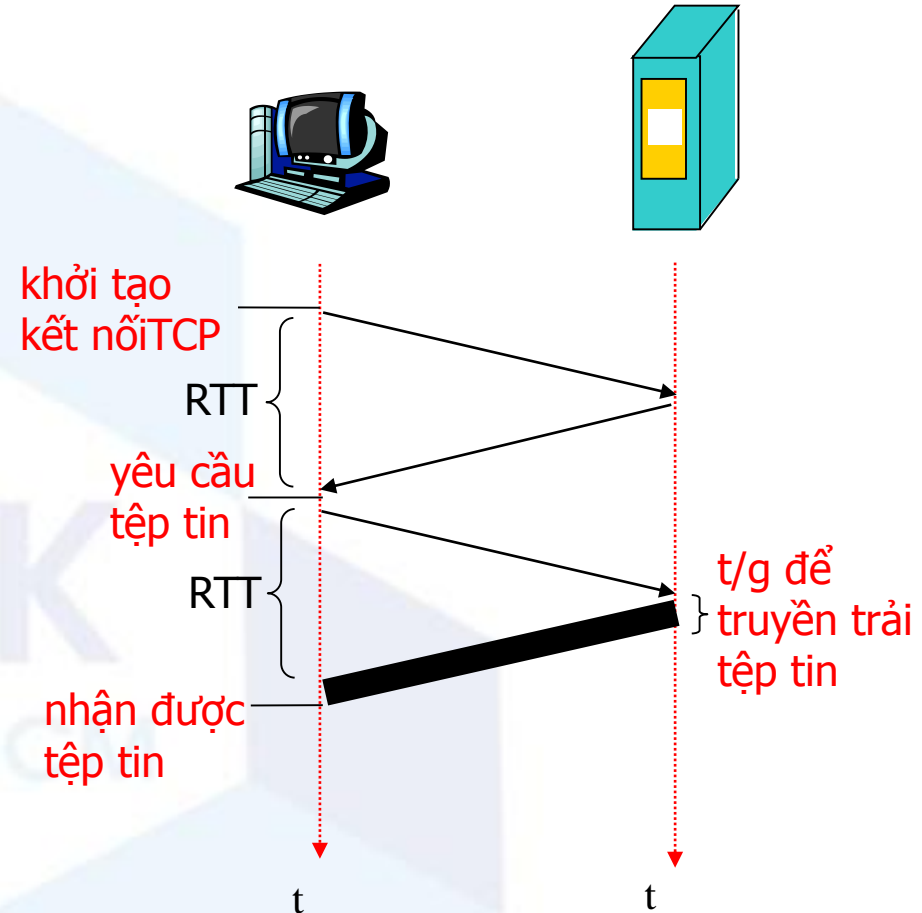
## Thời gian xoay vòng – RTT:

là t/g cần để một gói tin nhỏ di chuyển từ khách tới chủ và quay trở lại

## Thời gian phản hồi:

- một RTT để khởi tạo kết nối TCP
- một RTT để gửi truy vấn HTTP và vài byte đầu của phản hồi HTTP quay trở lại
- thời gian truyền tải tệp

**tổng =  $2RTT + \text{t/g truyền tải}$**



# HTTP dài hạn (Persistent)

## Những vấn đề của HTTP không ổn định:

- yêu cầu 2 RTT cho mỗi đối tượng
- hệ điều hành (HĐH – OS) tăng tải cho mỗi kết nối TCP
- trình duyệt thường mở nhiều kết nối TCP song song để tải những đối tượng liên kết

## HTTP ổn định

- máy chủ giữ lại liên kết sau khi gửi phản hồi, gọi là liên kết mở
- những thông điệp HTTP tiếp theo giữa cặp khách/chủ đó sẽ được gửi qua liên kết mở ở trên
- khách gửi truy vấn ngay khi nó gặp phải một liên kết tới đối tượng
- chỉ cần 1 RTT cho tất cả các đối tượng được liên kết tới

# Thông điệp truy vấn HTTP

- hai loại thông điệp HTTP: **truy vấn (*request*)**, **phản hồi (*response*)**
- **thông điệp truy vấn HTTP:**
  - ASCII (định dạng con người đọc được)

dòng truy vấn  
(các câu lệnh GET,  
POST, HEAD)

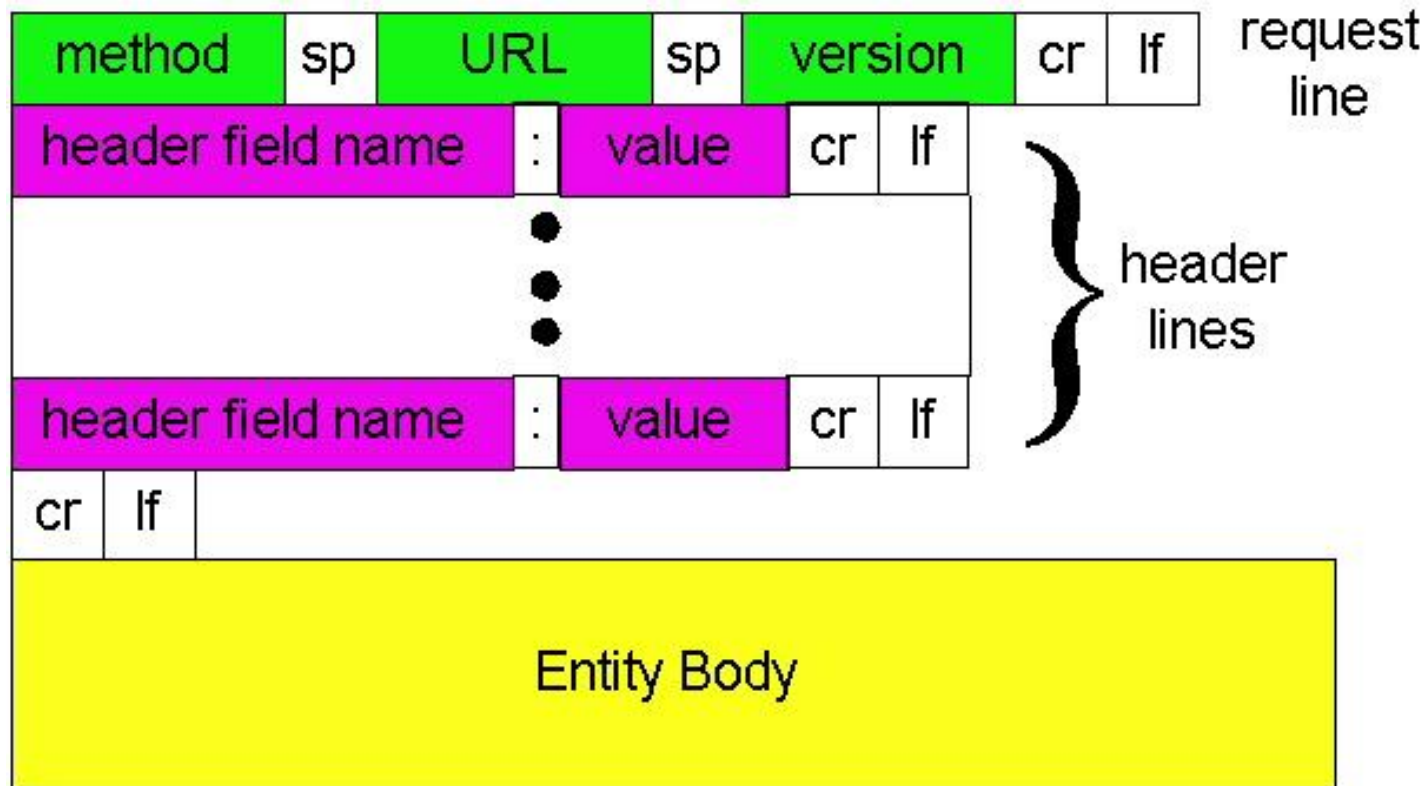
các dòng  
mào đầu

Xuống dòng  
báo hiệu kết thúc  
thông điệp

(dấu xuống dòng phụ)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

# Thông điệp truy vấn HTTP: định dạng chung



# Khung nhập dữ liệu tải lên

## Phương pháp POST:

- Trang web thường có những khung điền dữ liệu
- Dữ liệu vào được tải lên máy chủ trong phần **thân (entity body)** của thông điệp truy vấn

## Phương pháp GET (URL):

- sử dụng phương pháp GET
- Dữ liệu vào được tải lên trực tiếp trong các trường của URL của dòng truy vấn

`www.somesite.com/animalsearch?monkeys&banana`

# Các loại phương pháp

## HTTP/1.0

- GET
- POST
- HEAD
  - đòi hỏi máy chủ đặt đối tượng được yêu cầu nằm ngoài thông điệp phản hồi

## HTTP/1.1

- GET, POST, HEAD
- PUT
  - tải tệp tin trong phần thân lên đường dẫn được nêu trong URL
- DELETE
  - xóa tệp được nêu trong URL





# Thông điệp phản hồi HTTP

dòng trạng thái  
(giao thức  
mã trạng thái  
câu trạng thái)

các dòng  
mào đầu

dữ liệu, vd:  
tệp HTML  
được yêu cầu

```
HTTP/1.1 200 OK
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

dữ liệu ... dữ liệu ...
```

# Mã trạng thái của thông điệp phản hồi HTTP

Nằm trong dòng đầu tin của thông điệp phản hồi từ chủ->khách  
Một vài mã ví dụ:

**200 OK**

- truy vấn thành công, đối tượng theo sau trong thông điệp này

**301 Được di chuyển vĩnh viễn - Moved permanently**

- đối tượng truy vấn đã được di chuyển, vị trí mới kèm theo sau trong thông điệp này (Location:)

**400 Truy vấn không hợp lệ - Bad Request**

- máy chủ không hiểu thông điệp truy vấn

**404 Không tìm thấy - Not Found**

- tài liệu yêu cầu không tìm thấy trên máy chủ này

**505 Phiên bản HTTP không được hỗ trợ -HTTP Version Not Supported**

# Vọc HTTP (với vai khách)

## 1. Telnet tới trang web yêu thích:

```
telnet cis.poly.edu 80
```

Mở kết nối TCP tới cổng 80 (cổng mặc định của máy chủ Web) tại cis.poly.edu.  
Tất cả những gì ta gõ vào đều được gửi tới cổng 80 tại cis.poly.edu

## 2. Gõ vào một truy vấn HTTP "GET" :

```
GET /~ross/ HTTP/1.1  
Host: cis.poly.edu
```

Bằng cách này (nhấn vào nút Enter 2 lần), bạn gửi một truy vấn GET tối thiểu (nhưng đầy đủ) tới máy chủ HTTP

## 3. Xem xét thông điệp phản hồi mà máy chủ HTTP gửi trả lại cho bạn

# Trạng thái người dùng/máy chủ: cookies

Nhiều trang web lớn sử dụng cookies

## Bốn thành phần:

- 1) dòng mào đầu cookie của thông điệp phản hồi HTTP
- 2) dòng mào đầu cookie của thông điệp truy vấn HTTP
- 3) tệp tin cookie được lưu trên máy người dùng và quản lý bởi trình duyệt
- 4) thông tin về phiên làm việc của người dùng đồng thời được lưu tại CSDL của trang web

## Ví dụ:

- Người dùng luôn luôn truy cập Internet từ máy tính cá nhân
- đầu tiên, lướt một trang thương mại điện tử
- khi truy vấn HTTP đầu tiên đến trang web, trang web sẽ tạo ra:
  - số định danh (ID) độc nhất
  - một thẻ ghi trong CSDL cho ID đó

# Cookies: duy trì “trạng thái” (tt)

khách

chủ



tệp cookie



1 tuần sau:



truy vấn http

phản hồi http  
**Set-cookie: 1678**

truy vấn http  
**cookie: 1678**

phản hồi http

truy vấn http  
**cookie: 1678**

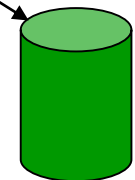
phản hồi http

máy chủ Amazon  
tạo ra ID  
1678 cho ng/d

hành vi khi  
có cookie

hành vi khi  
có cookie

tạo  
mục



CSDL  
máy chủ

truy cập

truy cập

# Cookies (tt)

## Cookies có thể chứa gì? :

- thông tin ủy quyền
- giỏ mua sắm trực tuyến
- trạng thái phiên làm việc người dùng (Web e-mail)

## Cookies và sự riêng tư: hiệu ứng

- cookies cho phép trang web tìm hiểu nhiều thông tin về bạn
- bạn có thể cung cấp tên và email cho các web

## Làm sao để duy trì “trạng thái”:

- các đầu cuối: lưu trạng thái ở người gửi, người nhận qua những giao dịch
- cookies: các thông điệp HTTP mang các thông tin trạng thái