TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA CÔNG NGHỆ THÔNG TIN

MẠNG MÁY TÍNH

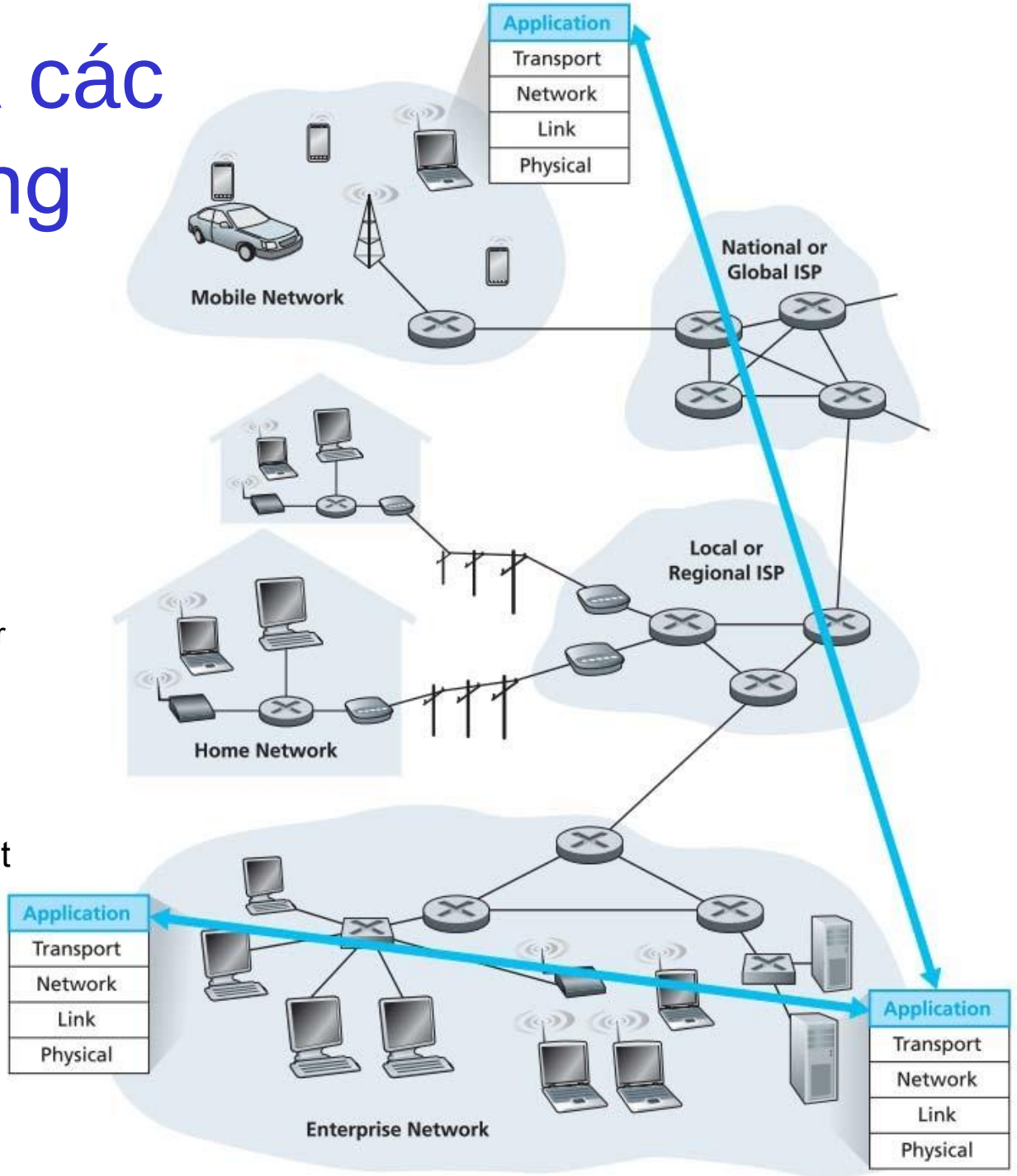
ThS. Đoàn Thị Quế
Email: dtque@tlu.edu.vn

Chương 2: Tầng ứng dụng

- Đặc điểm của ứng dụng mạng
- Web và HTTP
- Truyền tập tin: FTP
- Thư điện tử
- Hệ thống tên miền: DNS
- Ứng dụng ngang hàng

Giao tiếp giữa các ứng dụng mạng

- ứng dụng mạng (network application): chương trình chạy trên end system và giao tiếp với end system khác qua mạng
- ứng dụng Web
  - hai chương trình
    - trinhd duyệt chạy trên host của người dùng
    - chương trình Web server chạy trên Web server host
- hệ thống chia sẻ tập tin ngang hàng (P2P file-sharing system):
  - chương trình trên mỗi host tham gia vào hệ thống chia sẻ tập tin

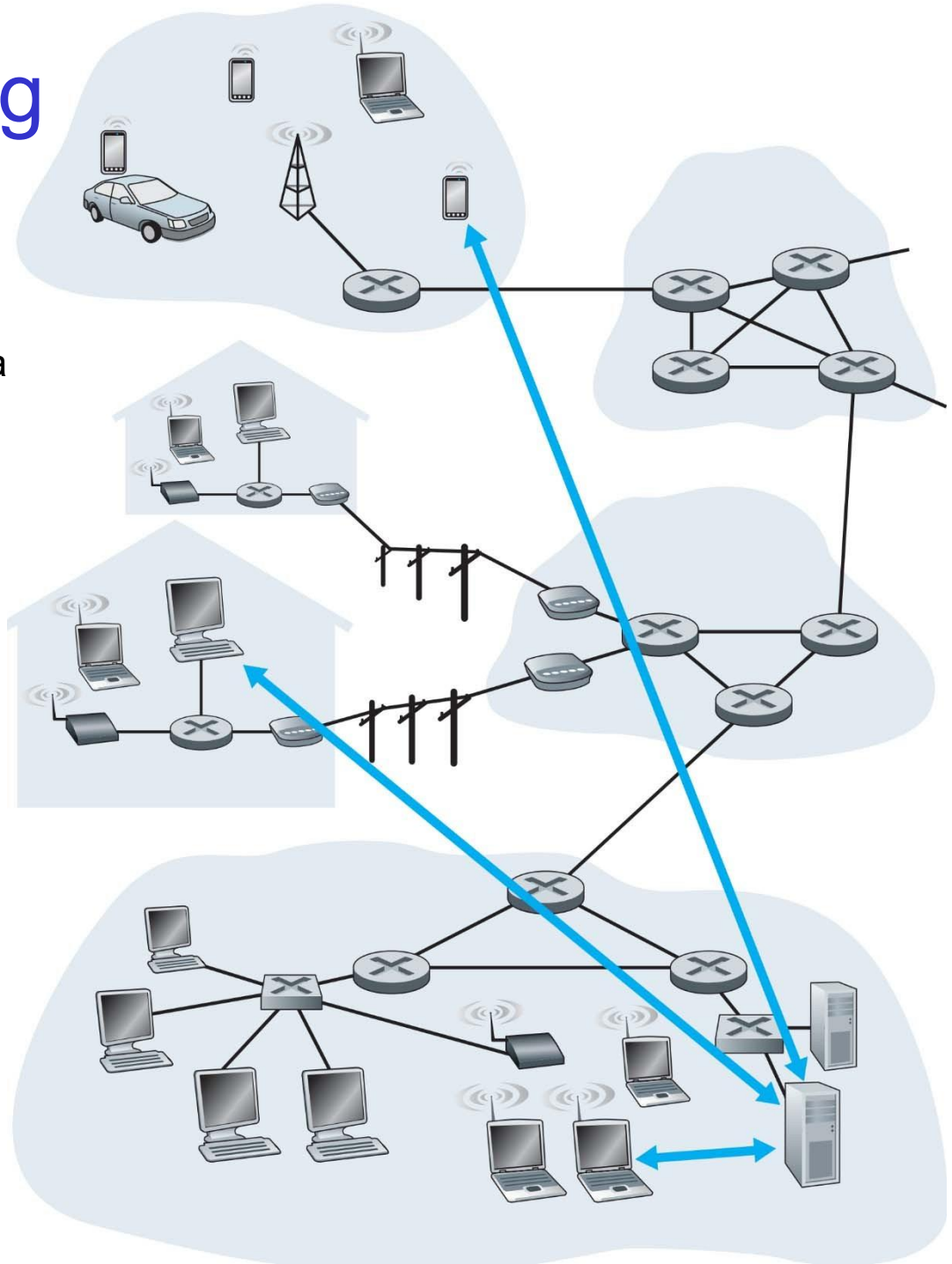


Kiến trúc ứng dụng mạng

- Kiến trúc ứng dụng và kiến trúc mạng
- Hai kiểu kiến trúc dùng trong ứng dụng mạng:
  - kiến trúc khách chủ (client-server)
  - kiến trúc ngang hàng (peer-to-peer, P2P)

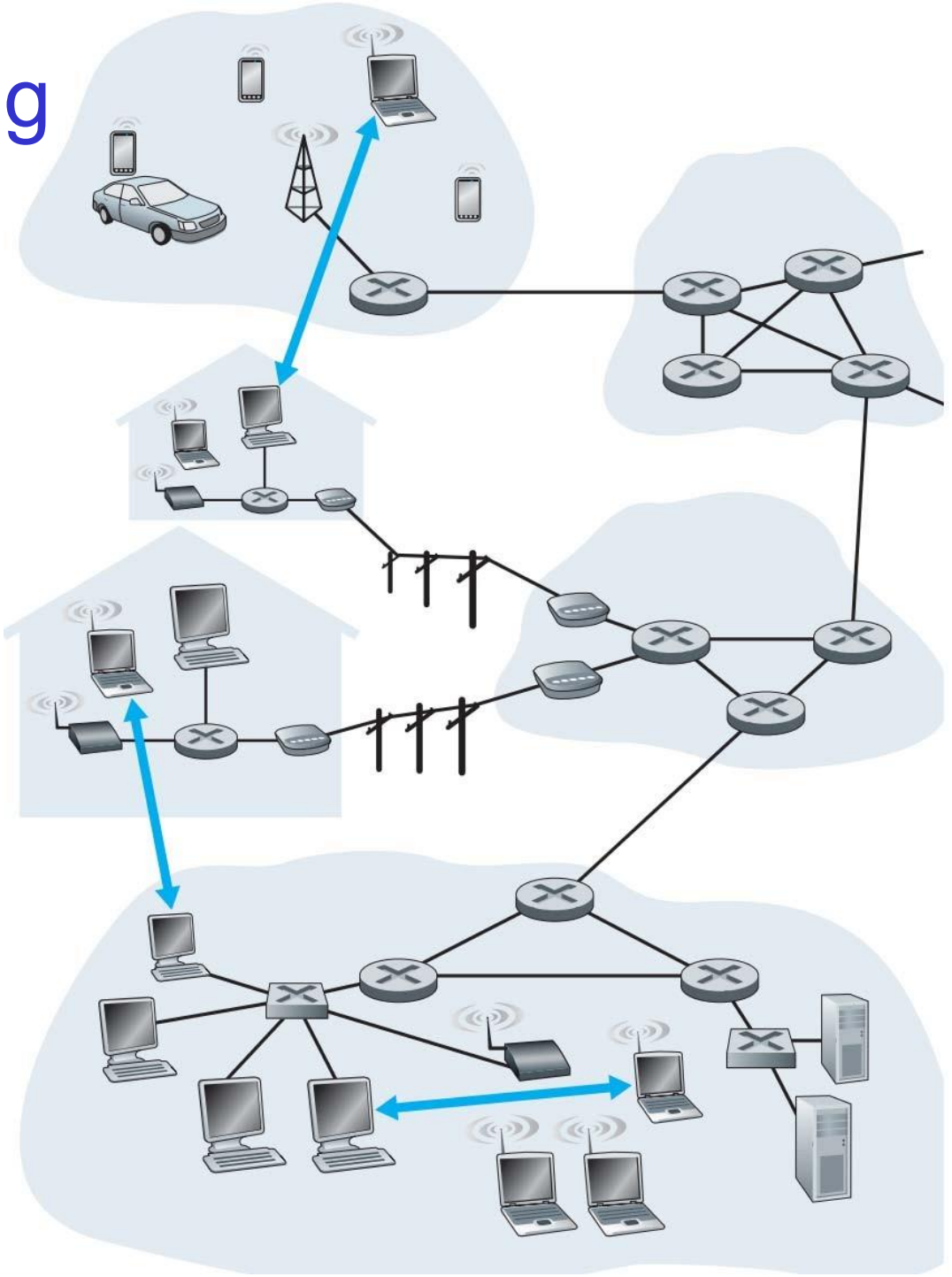
Kiến trúc ứng dụng mạng

- kiến trúc client-server
  - có 1 chương trình luôn chạy gọi là server, phục vụ nhiều chương trình chạy trên các host khác, gọi là client
  - các client không trực tiếp giao tiếp với nhau
  - server có địa chỉ cố định và các client biết địa chỉ này
- trong ứng dụng client-server, server chạy trên 1 host duy nhất thường không đáp ứng được tất cả các yêu cầu từ các client
  - data center, chứa một số lượng lớn host, thường sử dụng để tạo ra 1 virtual server mạnh
- một só ứng dụng client-server: Web, FTP, Telnet, e-mail



Kiến trúc ứng dụng mạng

- Kiến trúc P2P
  - phụ thuộc ít hoặc không phụ thuộc vào server dành riêng trong data center
  - ứng dụng khai thác trao đổi trực tiếp giữa cặp host khi có kết nối, gọi là peer
  - Peer không do có nhà cung cấp dịch vụ làm chủ mà hầu hết là do người sử dụng làm chủ
- Một số ứng dụng dùng kiến trúc: chia sẻ tập tin (như BitTorrent), Internet Telephony (như Skype), IPTV (như Kankan hay PPstream)



Kiến trúc ứng dụng mạng

- Kiến trúc lai
  - kết hợp cả client-sever và P2P
  - ví dụ các ứng dụng nhắn tin, voip (như skype)

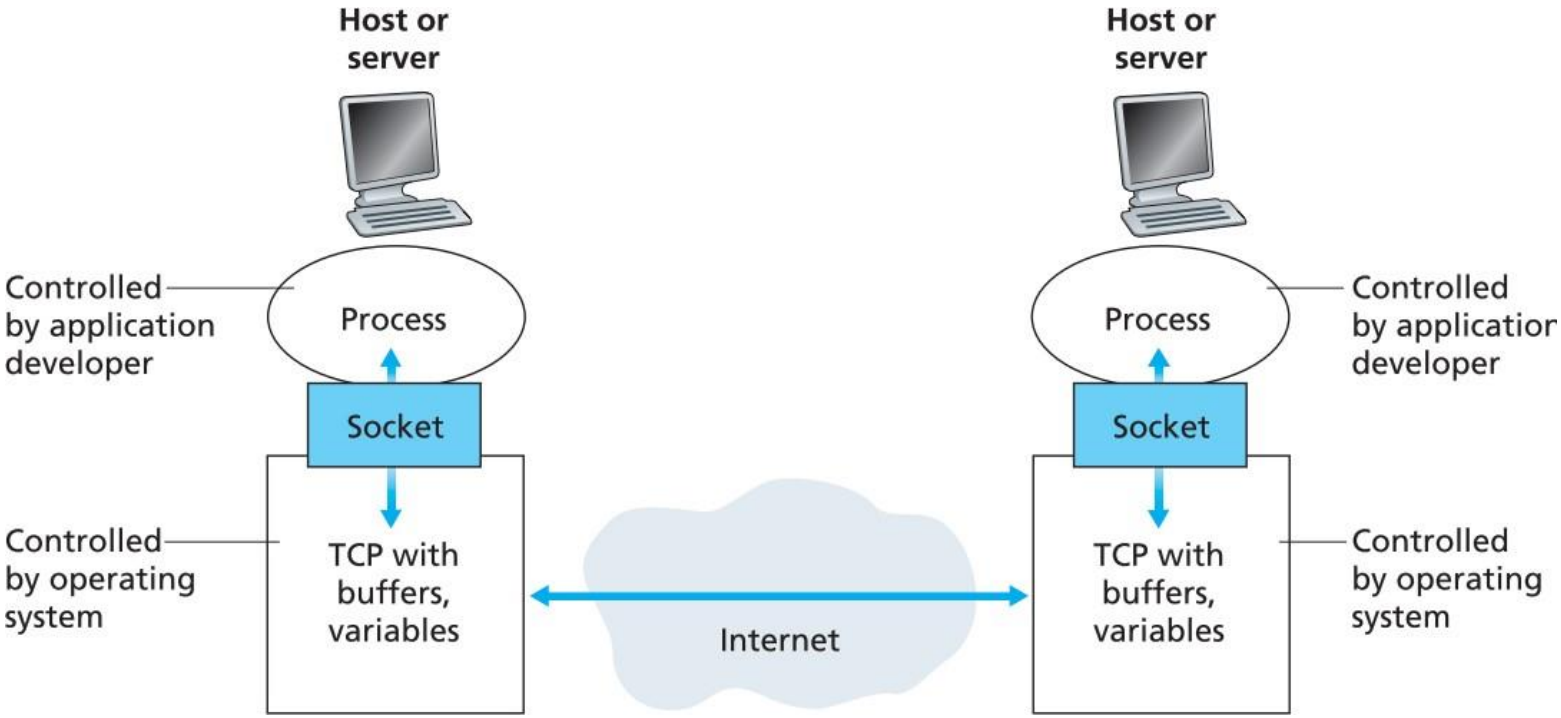
Kiến trúc ứng dụng mạng

- Ưu điểm và thách thức của kiến trúc P2P
  - ưu điểm
    - khả năng tự tùy biến về quy mô
    - hiệu quả chi phí
  - thách thức
    - ISP Friendly: Băng thông được ISP thiết lập sẵn cho Upload /Download
    - Bảo mật: Sự tham gia của nhiều người
    - Khích lệ: Người dùng sẵn sàng chia sẻ tài nguyên

Giao tiếp tiến trình

- Theo hệ điều hành, tiến trình thực hiện các giao tiếp chứ không phải chương trình
- Tiến trình client và Server:
  - tiến trình khởi tạo giao tiếp (liên lạc với tiến trình khác vào bắt đầu một phiên) gọi là client
  - tiến trình chờ để tiến trình khác liên lạc thì gọi là server
  - Ví dụ: Trong Web: client? server? Trong ứng dụng chia sẻ tập tin P2P, khi Peer A liên lạc với Peer B để gửi một tập tin: client? server?

Giao tiếp tiến trình



- Giao diện giữa tiến trình và mạng máy tính
  - Tiến trình gửi bản tin và nhận bản tin từ mạng qua một giao diện phần mềm gọi là socket
  - Socket là giao diện giữa tầng ứng dụng và tầng giao vận trong một host

Giao tiếp tiến trình

- Đánh địa chỉ tiến trình
  - để một tiến trình chạy trên một host gửi gói tin tới một tiến trình trên một host khác, tiến trình nhận cần có địa chỉ
    - (1) địa chỉ của host
    - (2) định danh xác định tiến trình nhận trong host đích
  - trong Internet
    - host được xác định bởi **IP address**
    - tiến trình nhận (hay socket nhận) chạy trên host đích được xác định bởi **port number**. Tại sao lại cần thông tin này?

Dịch vụ giao vận cung cấp cho ứng dụng

- Nhiều mạng, bao gồm Internet, cung cấp nhiều hơn một giao thức tầng giao vận. Khi phát triển ứng dụng, người phát triển phải chọn một trong các giao thức của tầng giao vận. Làm sao để chọn giao thức tầng giao vận?
- Giao thức tầng giao vận có thể cung cấp các dịch vụ nào cho ứng dụng?
  - truyền dữ liệu tin cậy
  - thông lượng
  - thời gian
  - an toàn bảo mật

Dịch vụ giao vận cung cấp 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
  - Khi giao thức tầng giao vận không cung cấp dịch vụ truyền tin cậy, dữ liệu gửi bởi tiến trình gửi có thể không tới tiến trình nhận. Điều này có thể chấp nhận được đối với các ứng dụng chấp nhận mất tin (**loss-tolerant application**). Ví dụ loss-tolerant application?

Dịch vụ giao vận cung cấp cho ứng dụng

- Thông lượng
  - Trong một phiên truyền thông giữa hai tiến trình dọc một đường đi trong mạng, thông lượng là tốc độ mà tiến trình gửi có thể chuyển các bít 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 bít/giây và tầng giao vật sẽ đảm bảo rằng thông lượng khả dụng luôn thấp nhất là r bít/giây
  - Ứng dụng cần đảm bảo yêu cầu thông lượng gọi là ứng dụng tác động bởi 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?

Dịch vụ giao vận cung cấp cho ứng dụng

- Thời gian
  - Đảm bảo thời gian: ví dụ mỗi bít 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

Dịch vụ giao vận cung cấp 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

Dịch vụ tầng giao vận của Internet

- Internet có hai giao thức tầng giao vận là UDP và TCP
- Ứng dụng phải lựa chọn việc sử dụng UDP hay TCP

Dịch vụ tầng giao vận của Internet

- Yêu cầu dịch vụ của một số ứng dụng

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–1Mbps 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

Dịch vụ tầng giao vận của Internet

- 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. Có phải các dịch vụ bị ảnh hưởng bởi thời gian như Internet telephony không thể chạy trong Internet?
  - Internet ngày nay có thể cung cấp các dịch vụ chấp nhận được cho các ứng dụng bị ảnh hưởng bởi 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

Giao thức tầng giao vận

- Các bản tin của tầng giao vận (message) được cấu trúc như thế nào? Vai trò của các trường trong bản tin? Khi nào các tiến trình gửi bản tin?
- Giao thức tầng giao vận định nghĩa:
  - 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 (the HyperText Transfer Protocol [RFC 2616])
- Nhiều giao thức tầng ứng dụng là có bản quyền sở hữu
  - Giao thức tầng ứng dụng của Skype
- Phân biệt giữa ứng dụng mạng và giao thức tầng ứng dụng

Chương 2: Tầng ứng dụng

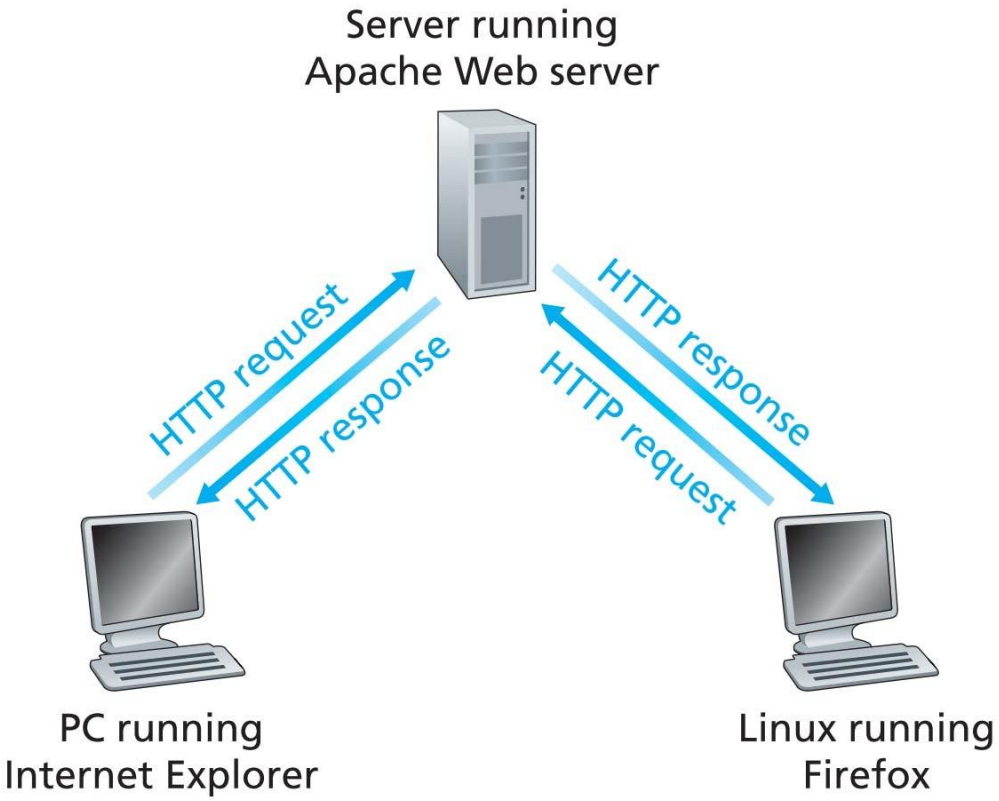
- Đặc điểm của ứng dụng mạng
- Web và HTTP
- Truyền tập tin: FTP
- Thư điện tử
- Hệ thống tên miền: DNS
- Ứng dụng ngang hàng

Web

- World Wide Web: Berners-Lee 1994
- 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 phi rất thấp

Giới thiệu về HTTP

- HyperText Transfer Protocol (HTTP): 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
- HTTP là giao thức không trạng thái **(stateless protocol)**



Kết nối Non-Persistent và Persistent

Kiểu không giữ kết nối (non-persistent HTTP)

- Một 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)

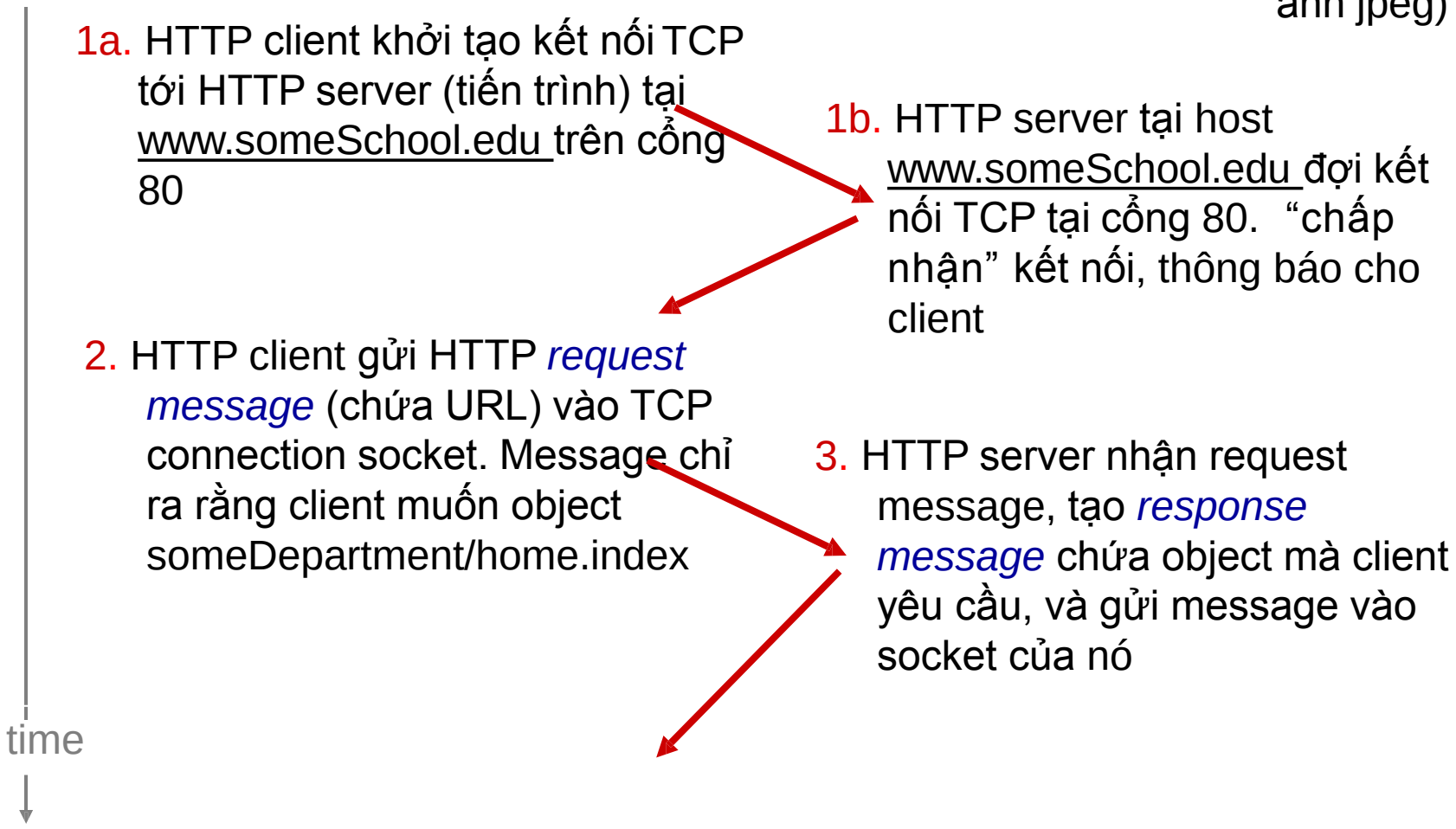
- Nhiều objects có thể gửi qua một kết nối TCP giữa client và server

Kết nối Non-Persistent và Persistent

- Non-Persistent HTTP

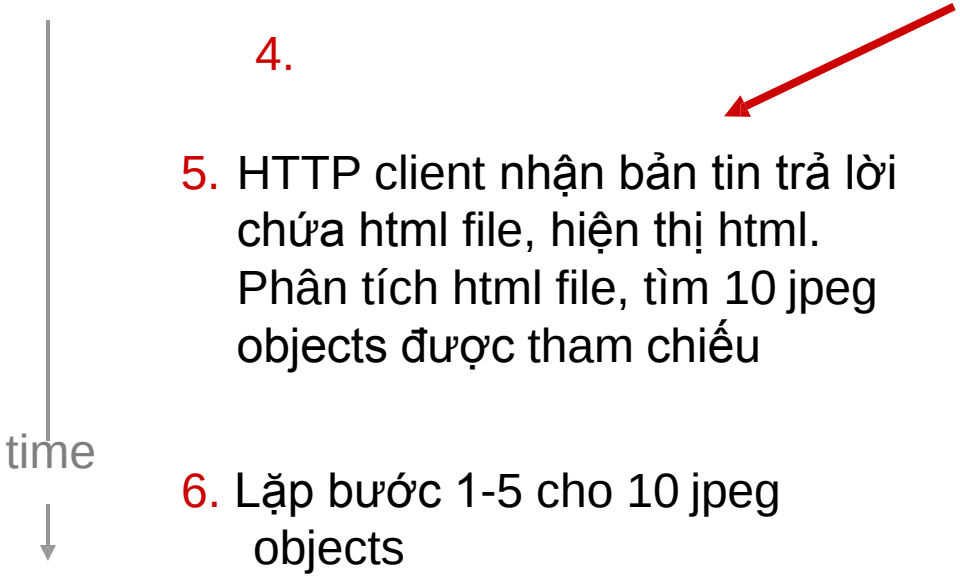
Người dùng gõ URL:
www.someSchool.edu/someDepartment/home.index

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



Kết nối Non-Persistent và Persistent

- Non-Persistent HTTP



Kết nối Non-Persistent và Persistent

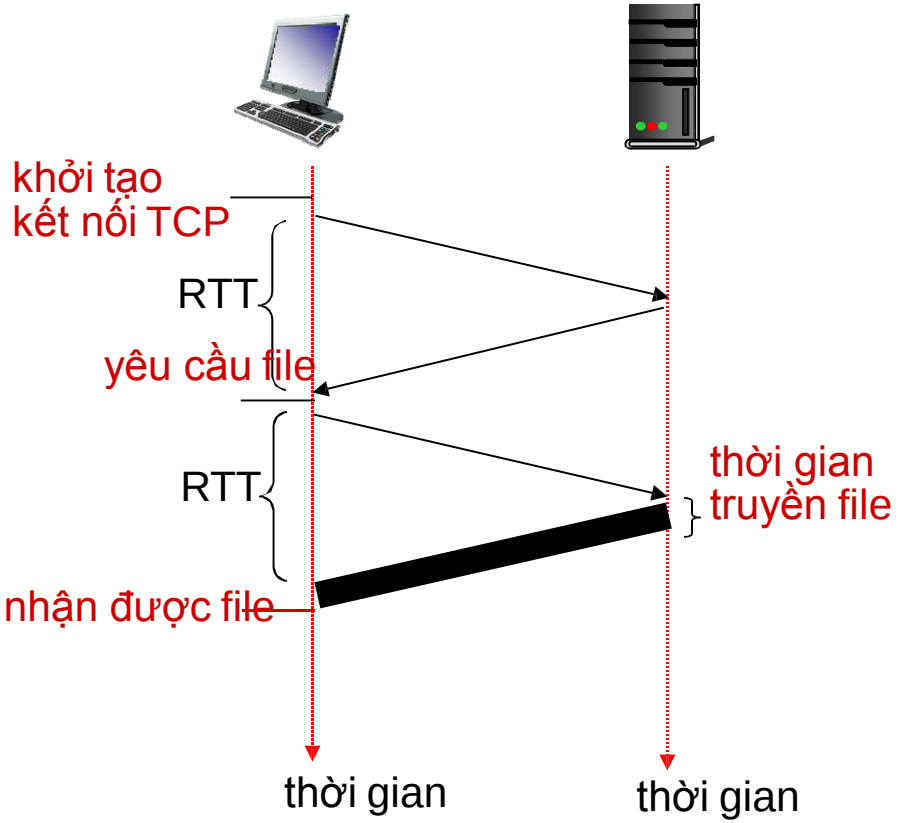
- Non-persistent HTTP: thời gian đáp ứng (HTTP response time)

RTT: thời gian để một gói tin gửi từ client tới sever rồi từ server tới client

HTTP response time:

- 1 RTT để tạo kết nối TCP
- 1 RTT cho HTTP request và một số byte đầu tiên của HTTP response tới client
- Thời gian truyền file
- Non-persistent HTTP response time =

  2RTT+ thời gian truyền file



So sánh kết nối Non-Persistent và Persistent

non-persistent HTTP:

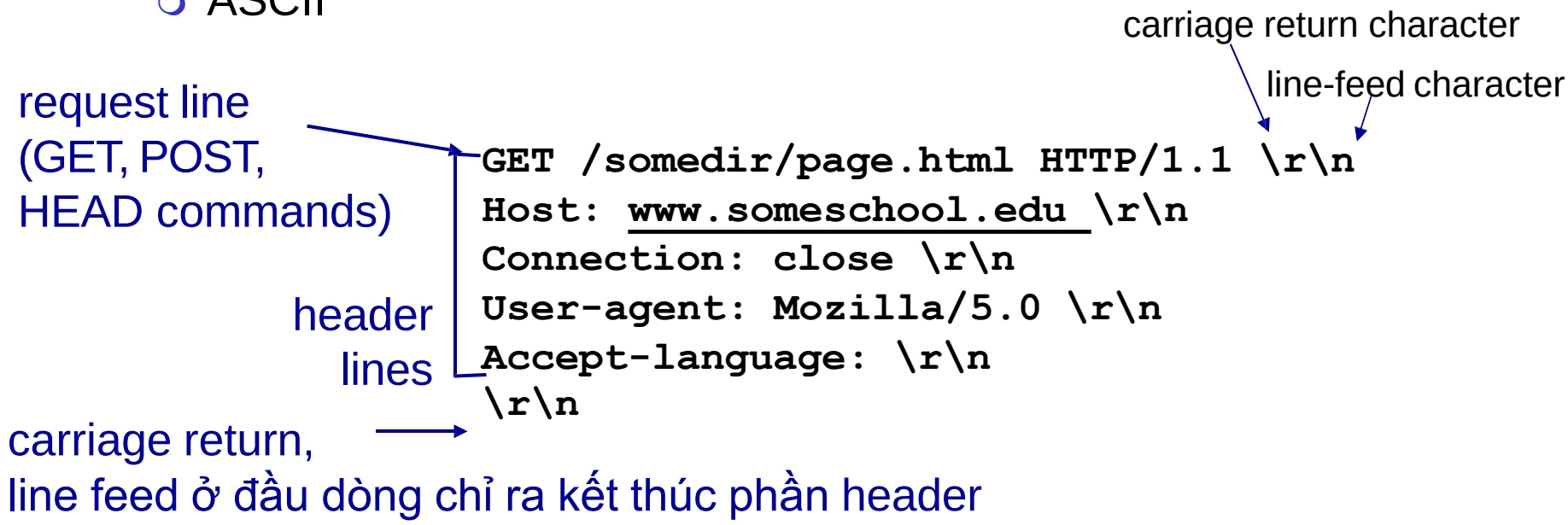
- Một kết nối TCP mới phải thiết lập cho mỗi object
- Cần 2 RTT cho 1 object
- Dư thừa dữ liệu ở mỗi kết nối TCP
- Trình duyệt thường mở đồng thời nhiều kết nối TCP song song để lấy về các object được tham chiếu

persistent HTTP:

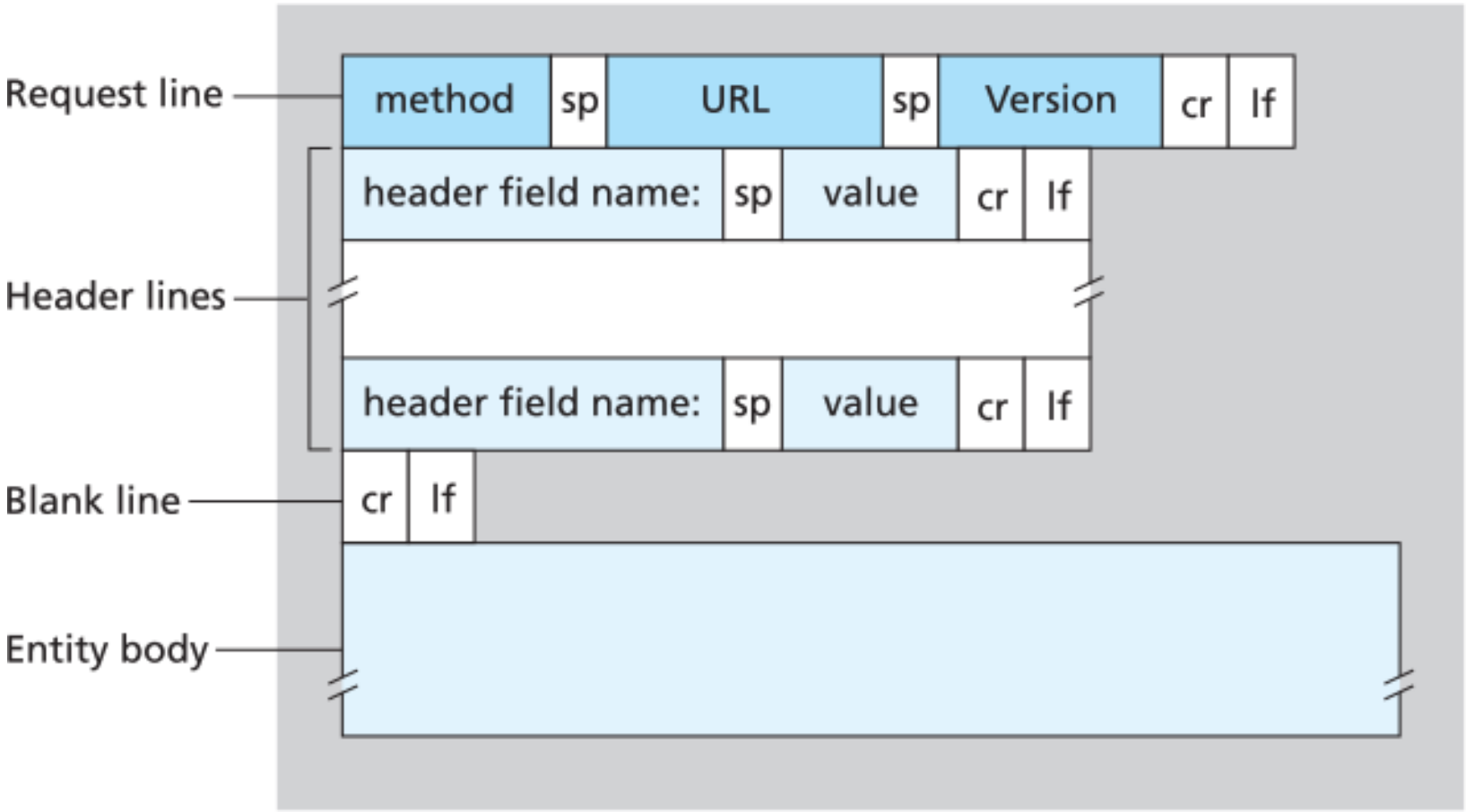
- Server giữ kết nối TCP sau khi gửi response
- Các HTTP message sau đó giữa client/server được gửi qua kết nối này
- Client gửi request ngay khi thấy một object được tham chiếu
- 1 RTT cho tất cả object được tham chiếu

Cấu trúc HTTP Message: HTTP request message

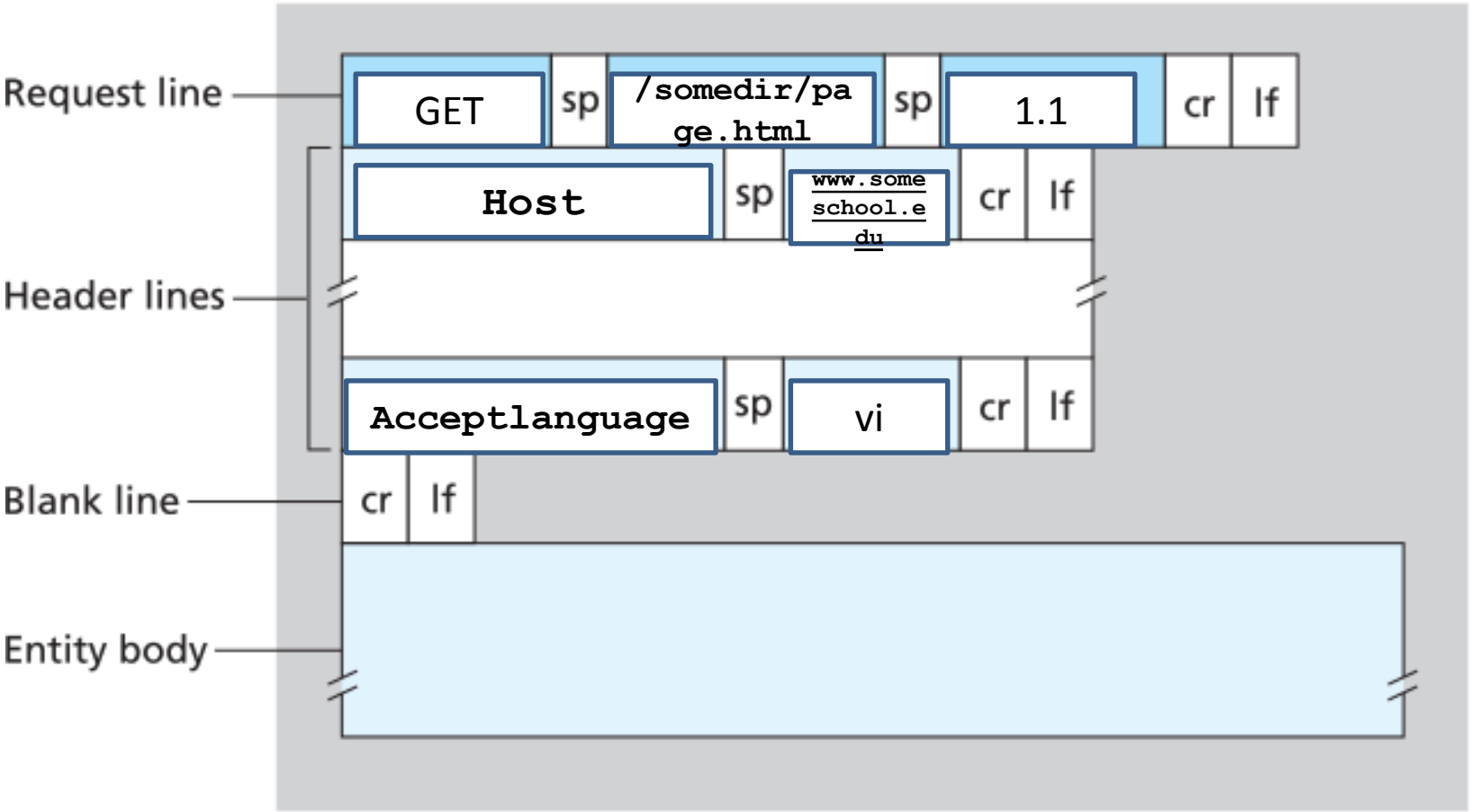
- 2 kiểu HTTP message: *request*, *response*
- HTTP request message:
  - ASCII



Cấu trúc HTTP Message: HTTP request message



Cấu trúc HTTP Message: HTTP request message



Cấu trúc HTTP Message: HTTP response message

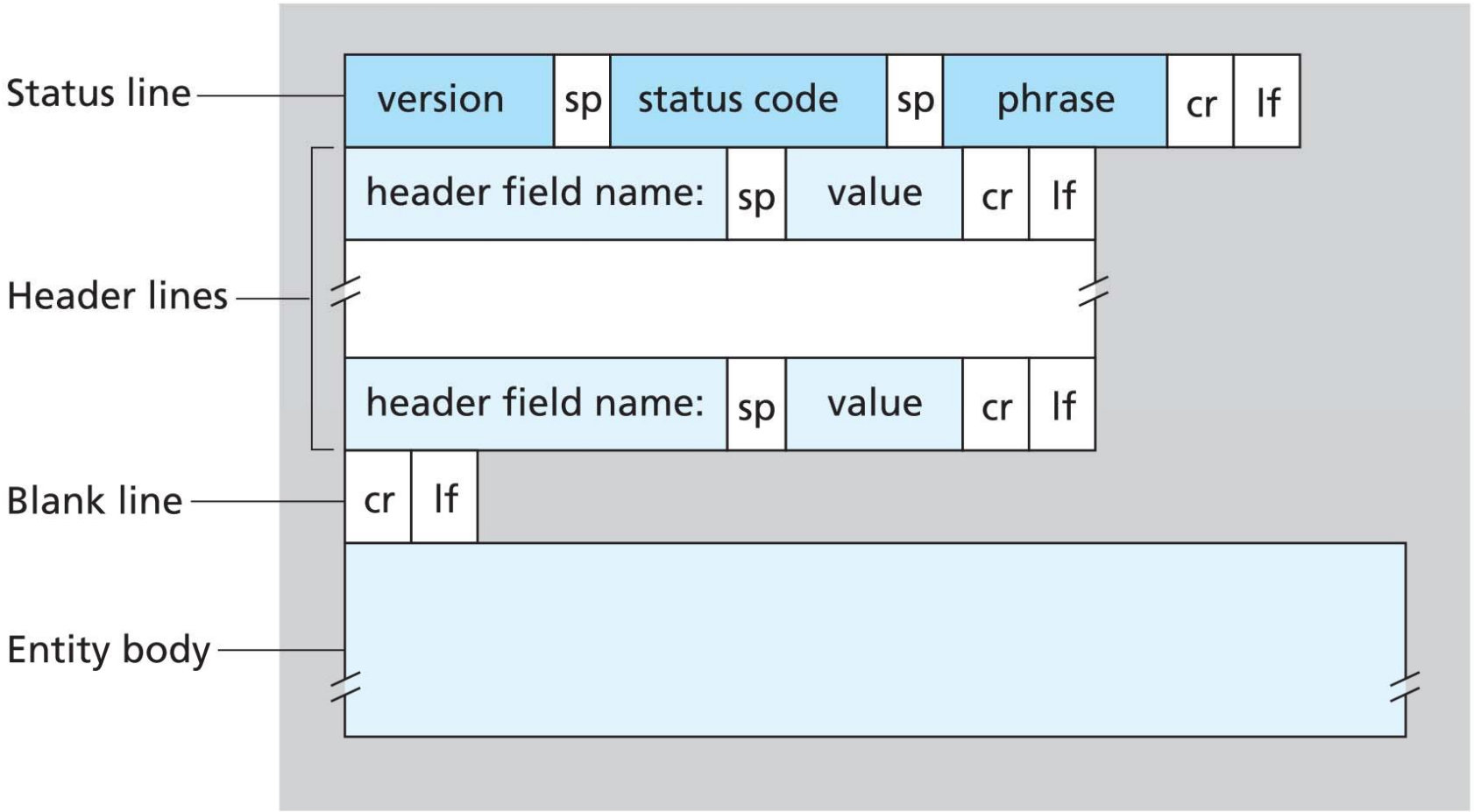
status line (protocol status code status phrase)

header lines

data, e.g., requested HTML file

```
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 ...
```

Cấu trúc HTTP Message: HTTP response message



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: netlab.tlu.edu.vn
3. Dừng việc ghi dữ liệu của phần mềm Wireshark
4. Gõ Filter: http.host == "netlab.tlu.edu.vn"
5. Quan sát HTTP message

Quan sát HTTP message: Trực tiếp tạo HTTP request

Chạy Wireshark để ghi lại dữ liệu

1. Telnet tới Web server:

```
telnet
set localecho
open netlab.tlu.edu.vn 80
```

mở kết nối TCP connection tới cổng 80 (cổng mặc định của HTTP server) tại netlab.tlu.edu.vn
nội dung gõ sẽ gửi tới cổng 80 tại netlab.tlu.edu.vn

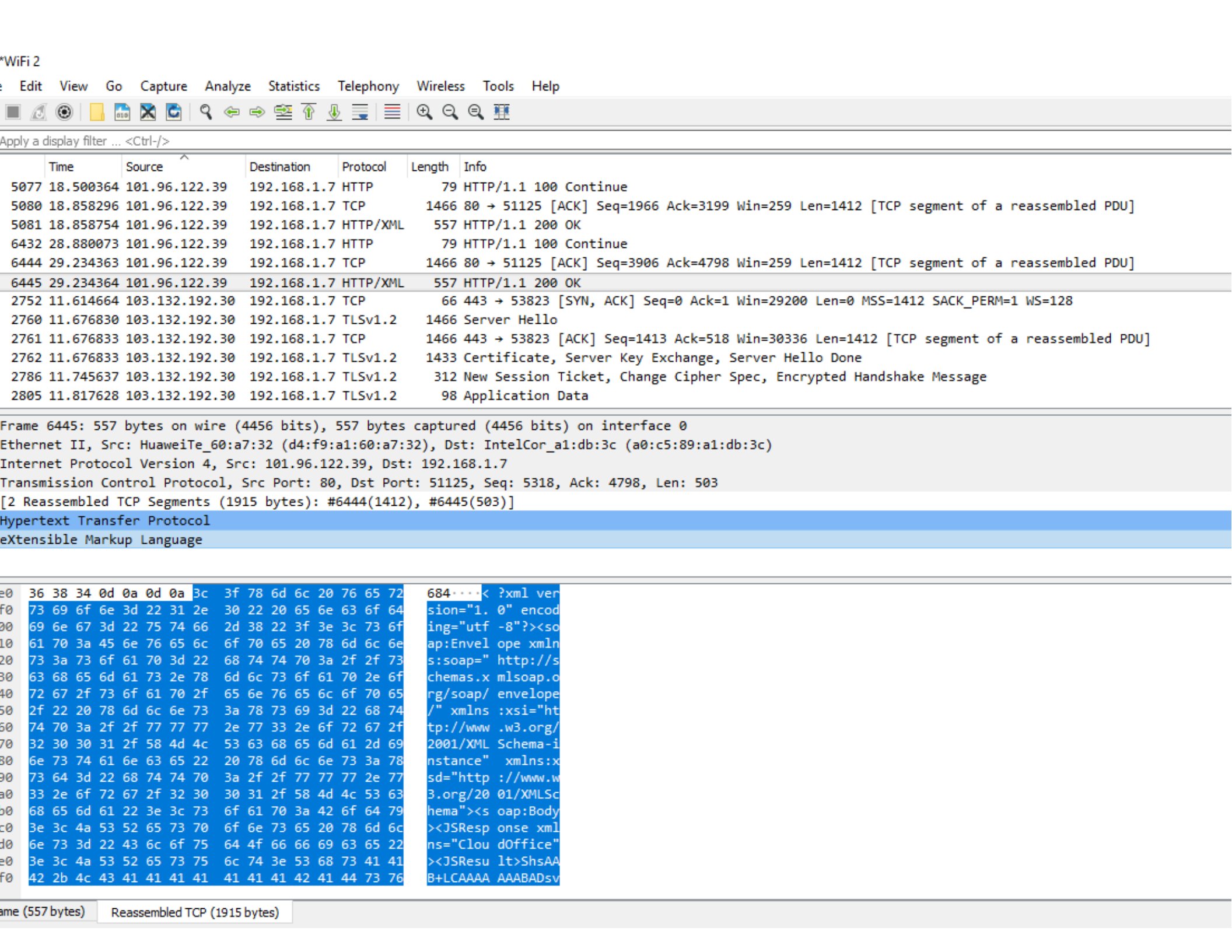
2. Gõ GET HTTP request:

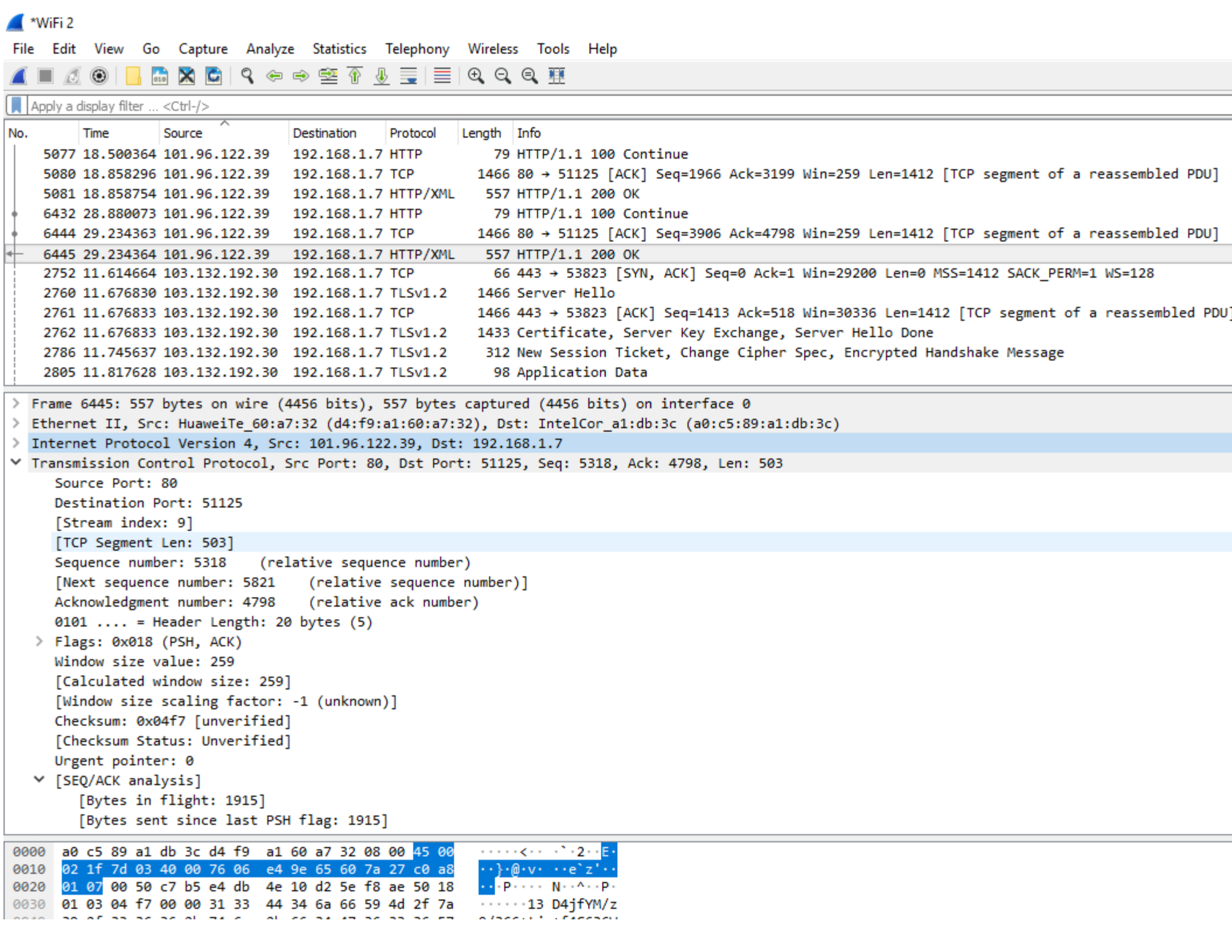
```
GET /~minhpt/ HTTP/1.1
Host: netlab.tlu.edu.vn
```

gõ (bấm xuống dòng 2 lần), bạn đã gửi GET request tới HTTP server

3. Quan sát response message do HTTP server gửi lại

(hoặc sử dụng Wireshark để xem HTTP request/response)

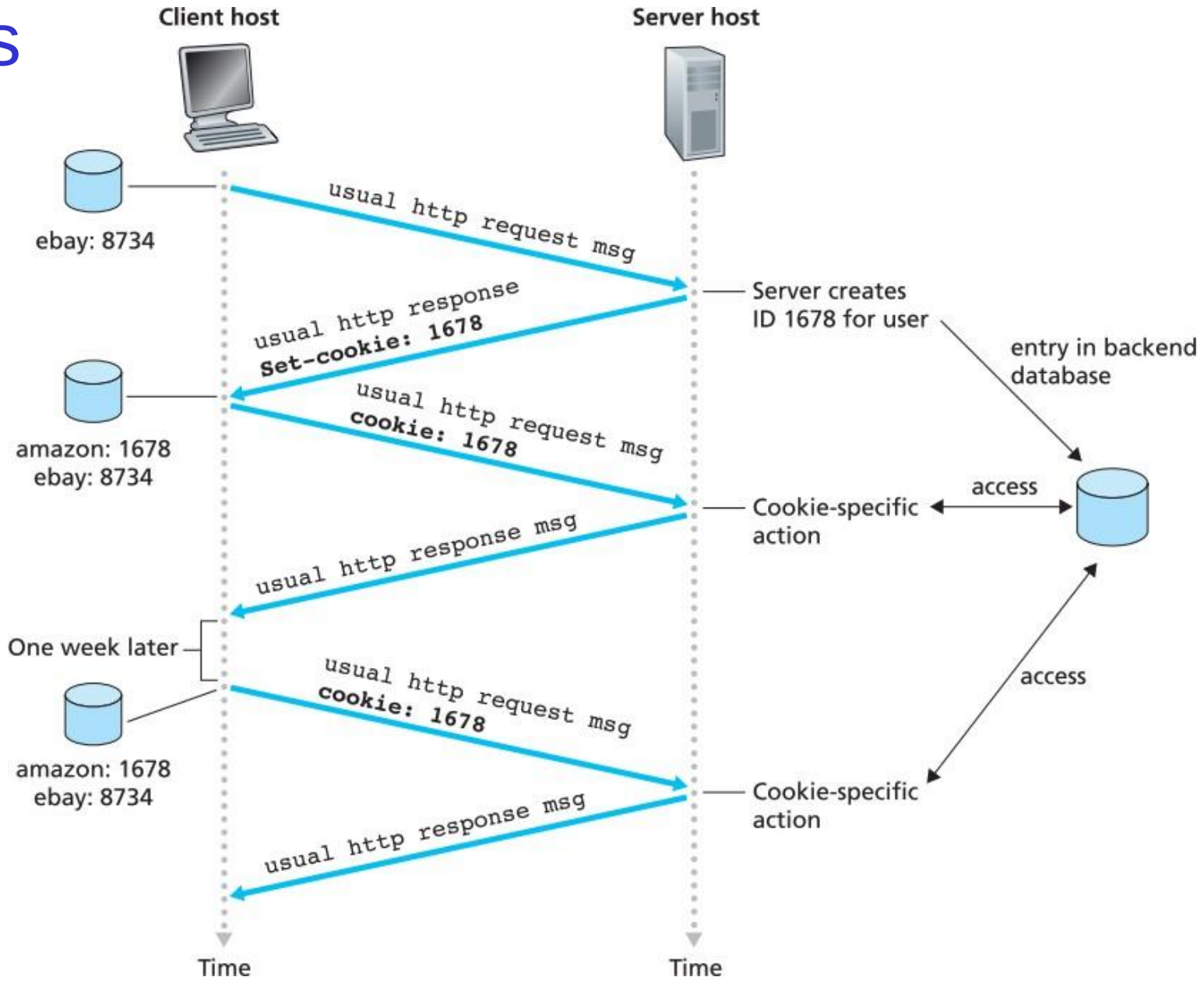

*WiFi 2
Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Time Source Destination Protocol Length Info
5077 18.500364 101.96.122.39 192.168.1.7 HTTP 79 HTTP/1.1 100 Continue
5080 18.858296 101.96.122.39 192.168.1.7 TCP 1466 80 → 51125 [ACK] Seq=1966 Ack=3199 Win=259 Len=1412 [TCP segment of a reassembled PDU]
5081 18.858754 101.96.122.39 192.168.1.7 HTTP/XML 557 HTTP/1.1 200 OK
6432 28.880073 101.96.122.39 192.168.1.7 HTTP 79 HTTP/1.1 100 Continue
6444 29.234363 101.96.122.39 192.168.1.7 TCP 1466 80 → 51125 [ACK] Seq=3906 Ack=4798 Win=259 Len=1412 [TCP segment of a reassembled PDU]
6445 29.234364 101.96.122.39 192.168.1.7 HTTP/XML 557 HTTP/1.1 200 OK
2752 11.614664 103.132.192.30 192.168.1.7 TCP 66 443 → 53823 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1412 SACK_PERM=1 WS=128
2760 11.676830 103.132.192.30 192.168.1.7 TLSv1.2 1466 Server Hello
2761 11.676833 103.132.192.30 192.168.1.7 TCP 1466 443 → 53823 [ACK] Seq=1413 Ack=518 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2762 11.676833 103.132.192.30 192.168.1.7 TLSv1.2 1433 Certificate, Server Key Exchange, Server Hello Done
2786 11.745637 103.132.192.30 192.168.1.7 TLSv1.2 312 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
2805 11.817628 103.132.192.30 192.168.1.7 TLSv1.2 98 Application Data
Frame 6445: 557 bytes on wire (4456 bits), 557 bytes captured (4456 bits) on interface 0
Ethernet II, Src: HuaweiTe_60:a7:32 (d4:f9:a1:60:a7:32), Dst: IntelCor_a1:db:3c (a0:c5:89:a1:db:3c)
Internet Protocol Version 4, Src: 101.96.122.39, Dst: 192.168.1.7
Transmission Control Protocol, Src Port: 80, Dst Port: 51125, Seq: 5318, Ack: 4798, Len: 503
[2 Reassembled TCP Segments (1915 bytes): #6444(1412), #6445(503)]
Hypertext Transfer Protocol
eXtensible Markup Language
e0 36 38 34 0d 0a 0d 0a 3c 3f 78 6d 6c 20 76 65 72 684···· < ?xml ver
f0 73 69 6f 6e 3d 22 31 2e 30 22 20 65 6e 63 6f 64 sion="1. 0" encod
00 69 6e 67 3d 22 75 74 66 2d 38 22 3f 3e 3c 73 6f ing="utf -8"?><so
10 61 70 3a 45 6e 76 65 6c 6f 70 65 20 78 6d 6c 6e ap:Envel ope xmln
20 73 3a 73 6f 61 70 3d 22 68 74 74 70 3a 2f 2f 73 s:soap=" http://s
30 63 68 65 6d 61 73 2e 78 6d 6c 73 6f 61 70 2e 6f chemas.x mlsoap.o
40 72 67 2f 73 6f 61 70 2f 65 6e 76 65 6c 6f 70 65 rg/soap/ envelope
50 2f 22 20 78 6d 6c 6e 73 3a 78 73 69 3d 22 68 74 /" xmlns :xsi="ht
60 74 70 3a 2f 2f 77 77 77 2e 77 33 2e 6f 72 67 2f tp://www .w3.org/
70 32 30 30 31 2f 58 4d 4c 53 63 68 65 6d 61 2d 69 2001/XML Schema-i
80 6e 73 74 61 6e 63 65 22 20 78 6d 6c 6e 73 3a 78 nstance" xmlns:x
90 73 64 3d 22 68 74 74 70 3a 2f 2f 77 77 77 2e 77 sd="http ://www.w
a0 33 2e 6f 72 67 2f 32 30 30 31 2f 58 4d 4c 53 63 3.org/20 01/XMLSc
b0 68 65 6d 61 22 3e 3c 73 6f 61 70 3a 42 6f 64 79 hema"><s oap:Body
c0 3e 3c 4a 53 52 65 73 70 6f 6e 73 65 20 78 6d 6c ><JSResp onse xml
d0 6e 73 3d 22 43 6c 6f 75 64 4f 66 66 69 63 65 22 ns="Clou dOffice"
e0 3e 3c 4a 53 52 65 73 75 6c 74 3e 53 68 73 41 41 ><JSResu lt>ShsAA
f0 42 2b 4c 43 41 41 41 41 41 41 41 42 41 44 73 76 B+LCAAAA AAABADsv
ame (557 bytes)
Reassembled TCP (1915 bytes)


*WiFi 2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
5077 18.500364 101.96.122.39 192.168.1.7 HTTP 79 HTTP/1.1 100 Continue
5080 18.858296 101.96.122.39 192.168.1.7 TCP 1466 80 → 51125 [ACK] Seq=1966 Ack=3199 Win=259 Len=1412 [TCP segment of a reassembled PDU]
5081 18.858754 101.96.122.39 192.168.1.7 HTTP/XML 557 HTTP/1.1 200 OK
6432 28.880073 101.96.122.39 192.168.1.7 HTTP 79 HTTP/1.1 100 Continue
6444 29.234363 101.96.122.39 192.168.1.7 TCP 1466 80 → 51125 [ACK] Seq=3906 Ack=4798 Win=259 Len=1412 [TCP segment of a reassembled PDU]
6445 29.234364 101.96.122.39 192.168.1.7 HTTP/XML 557 HTTP/1.1 200 OK
2752 11.614664 103.132.192.30 192.168.1.7 TCP 66 443 → 53823 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1412 SACK_PERM=1 WS=128
2760 11.676830 103.132.192.30 192.168.1.7 TLSv1.2 1466 Server Hello
2761 11.676833 103.132.192.30 192.168.1.7 TCP 1466 443 → 53823 [ACK] Seq=1413 Ack=518 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2762 11.676833 103.132.192.30 192.168.1.7 TLSv1.2 1433 Certificate, Server Key Exchange, Server Hello Done
2786 11.745637 103.132.192.30 192.168.1.7 TLSv1.2 312 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
2805 11.817628 103.132.192.30 192.168.1.7 TLSv1.2 98 Application Data
Frame 6445: 557 bytes on wire (4456 bits), 557 bytes captured (4456 bits) on interface 0
Ethernet II, Src: HuaweiTe_60:a7:32 (d4:f9:a1:60:a7:32), Dst: IntelCor_a1:db:3c (a0:c5:89:a1:db:3c)
Internet Protocol Version 4, Src: 101.96.122.39, Dst: 192.168.1.7
Transmission Control Protocol, Src Port: 80, Dst Port: 51125, Seq: 5318, Ack: 4798, Len: 503
Source Port: 80
Destination Port: 51125
[Stream index: 9]
[TCP Segment Len: 503]
Sequence number: 5318 (relative sequence number)
[Next sequence number: 5821 (relative sequence number)]
Acknowledgment number: 4798 (relative ack number)
0101 .... = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
Window size value: 259
[Calculated window size: 259]
[Window size scaling factor: -1 (unknown)]
Checksum: 0x04f7 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
[SEQ/ACK analysis]
[Bytes in flight: 1915]
[Bytes sent since last PSH flag: 1915]
0000 a0 c5 89 a1 db 3c d4 f9 a1 60 a7 32 08 00 45 00
0010 02 1f 7d 03 40 00 76 06 e4 9e 65 60 7a 27 c0 a8
0020 01 07 00 50 c7 b5 e4 db 4e 10 d2 5e f8 ae 50 18
0030 01 03 04 f7 00 00 31 33 44 34 6a 66 59 4d 2f 7a

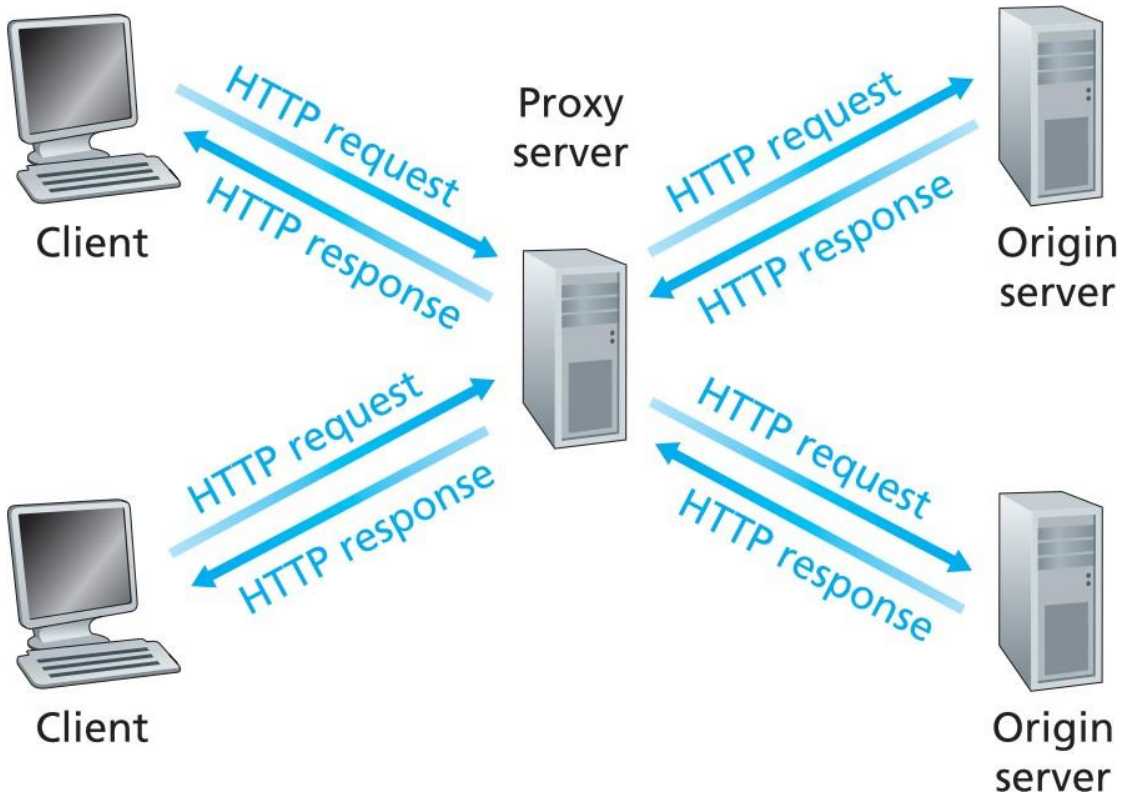
Một số mã trạng thái phổ biến

- 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 Vị trí: tiêu đề của thông báo phản hồi. Phần mềm máy khách sẽ tự động truy xuất URL mới.
- 400 Bad Request: Đây là mã lỗi chung cho biết rằng máy chủ không thể hiểu được yêu cầu đó.
- 404 Not Found: Tài liệu được yêu cầu không tồn tại trên máy chủ này.
- 505 HTTP Version Not Supported: Phiên bản giao thức HTTP được yêu cầu không được máy chủ hỗ trợ.

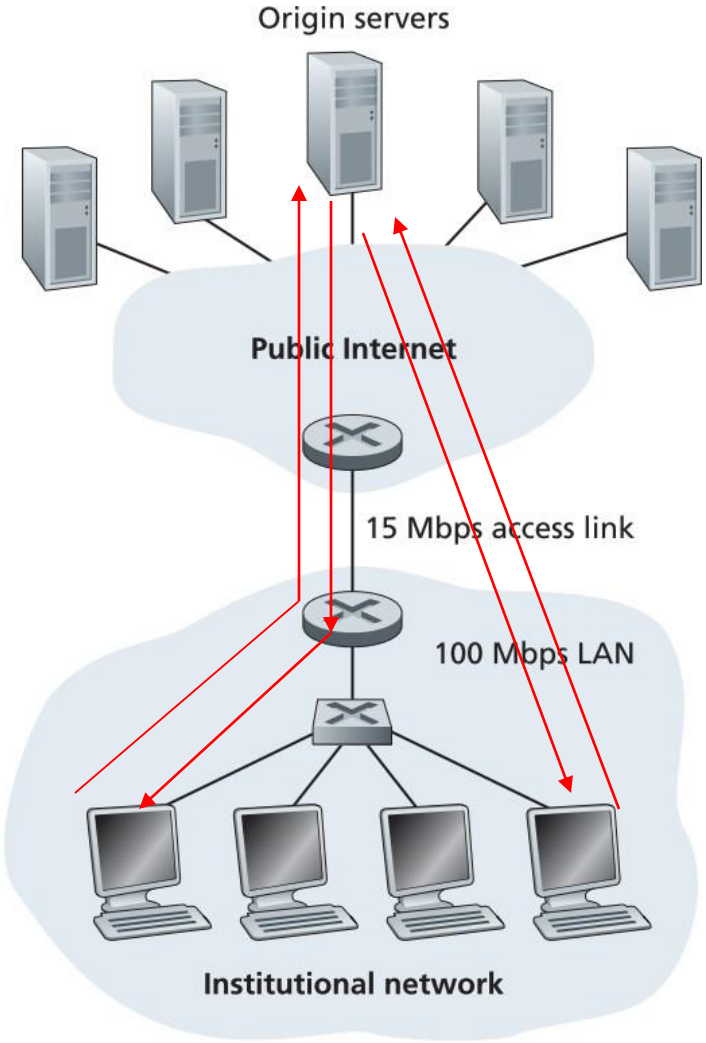
Tương tác User-Server: Cookies

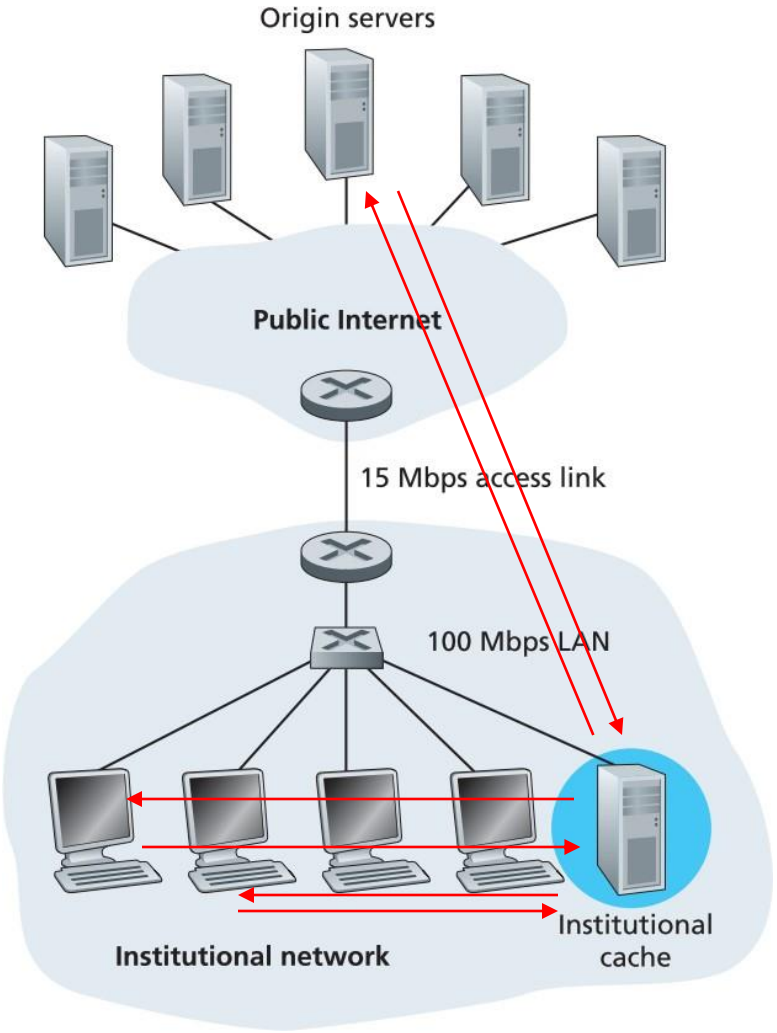


Web Caching


Client
HTTP request
HTTP response
Proxy server
HTTP request
HTTP response
Origin server
Client
HTTP request
HTTP response
HTTP request
HTTP response
Origin server

Web caching: Lợi ích

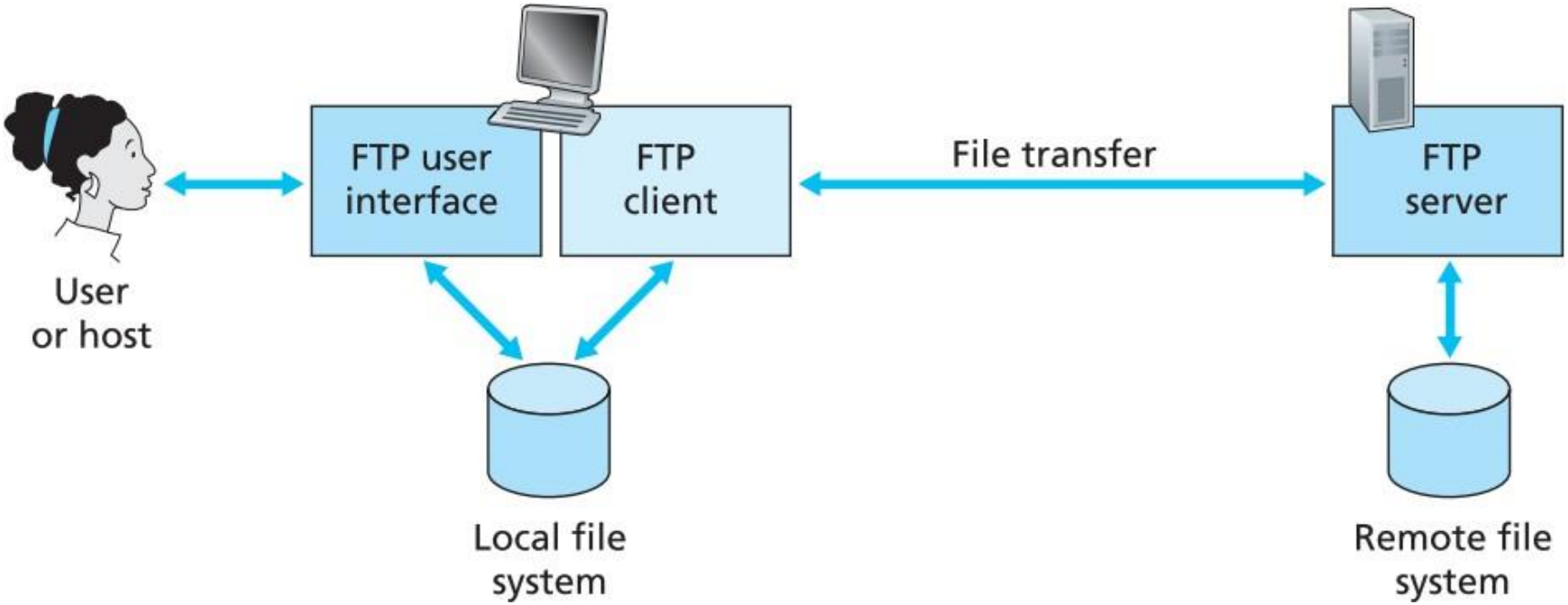




Chương 2: Tầng ứng dụng

- Đặc điểm của ứng dụng mạng
- Web và HTTP
- Truyền tập tin: FTP
- Thư điện tử
- Hệ thống tên miền: DNS
- Ứng dụng ngang hàng

Truyền tập tin: FTP



Truyền tập tin: FTP

- kết nối điều khiển (control connection): dùng riêng
  - HTTP connection: dùng chung
- FTP server duy trì trạng thái: thư mục hiện tại, xác thực
  - HTTP không duy trì trạng thái

FTP command, response

ví dụ command:

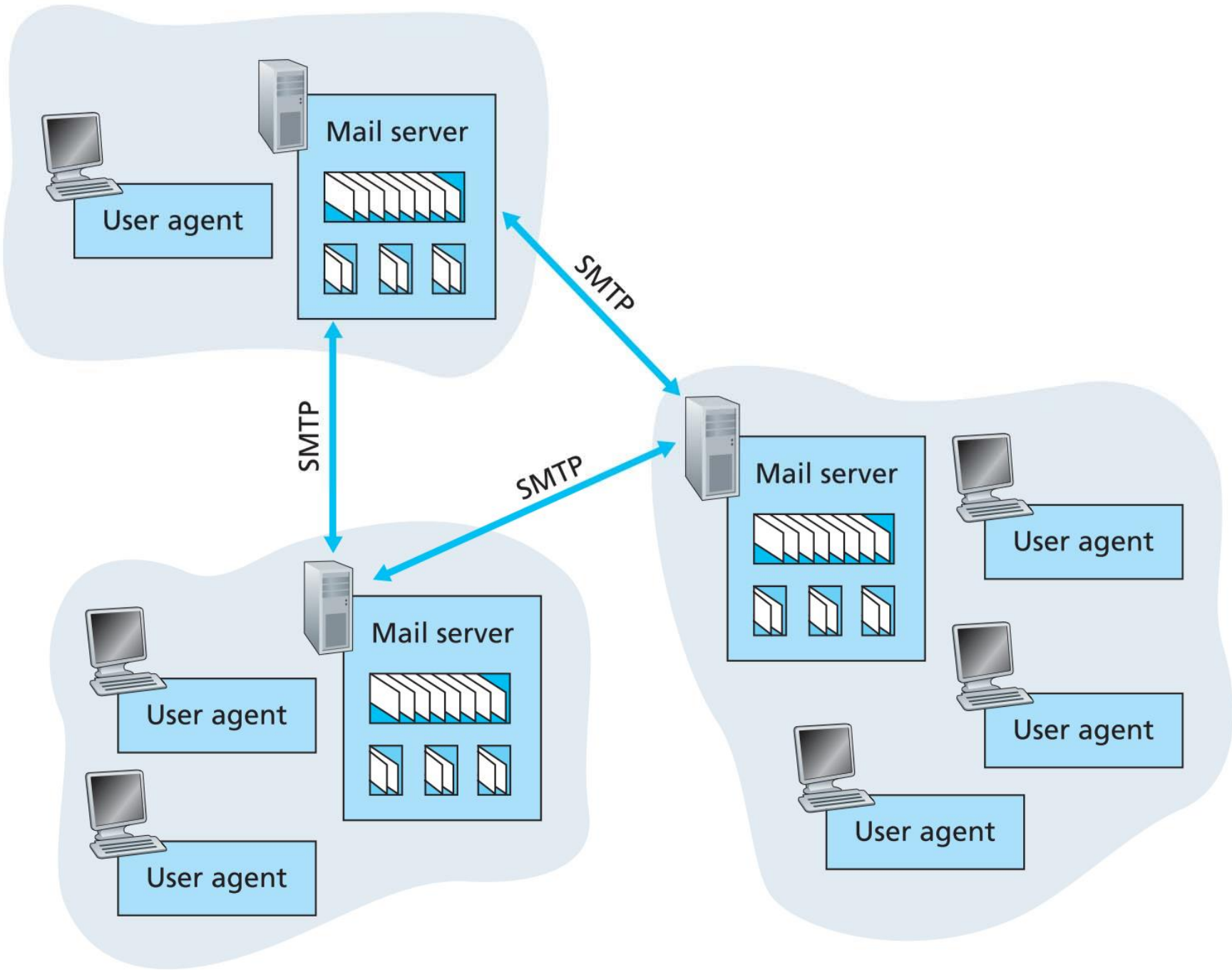
- gửi dạng ASCII text qua kết nối điều khiển
- **`USER`** ***`username`***
- **`PASS`** ***`password`***
- **`LIST`** trả về danh dách tập tin trong thư mục hiện tại
- **`RETR filename`** lấy (get) tập tin
- **`STOR filename`** lưu (put) tập tin lên remote host

ví dụ return code

- status code và phrase (như trong HTTP)
- **`331 Username OK, password required`**
- **`125 data connection already open; transfer starting`**
- **`425 Can't open data connection`**
- **`452 Error writing file`**

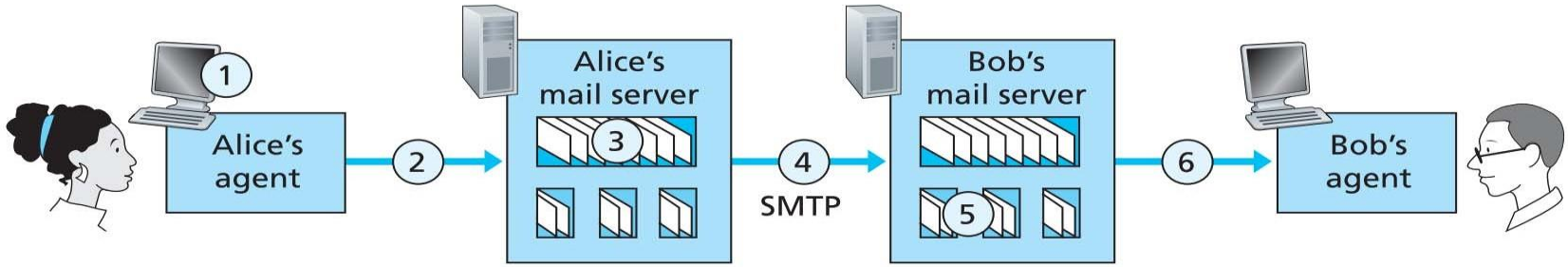
Chương 2: Tầng ứng dụng

- Đặc điểm của ứng dụng mạng
- Web và HTTP
- Truyền tập tin: FTP
- Thư điện tử
- Hệ thống tên miền: DNS
- Ứng dụng ngang hàng


Mail server
User agent
SMTP
SMTP
SMTP
Mail server
User agent
User agent
User agent
Mail server
User agent
User agent

SMTP

- RFC 2821
- sử dụng TCP để truyền tin cậy bản tin email từ client tới server, cổng 25
- gửi trực tiếp: server gửi tới server nhận
- giao tiếp kiểu command/response (giống HTTP, FTP)
  - commands: ASCII text
  - response: mã trạng thái (status code) và thông điệp (phrase)
- bản tin phải là 7-bit ASCI



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:

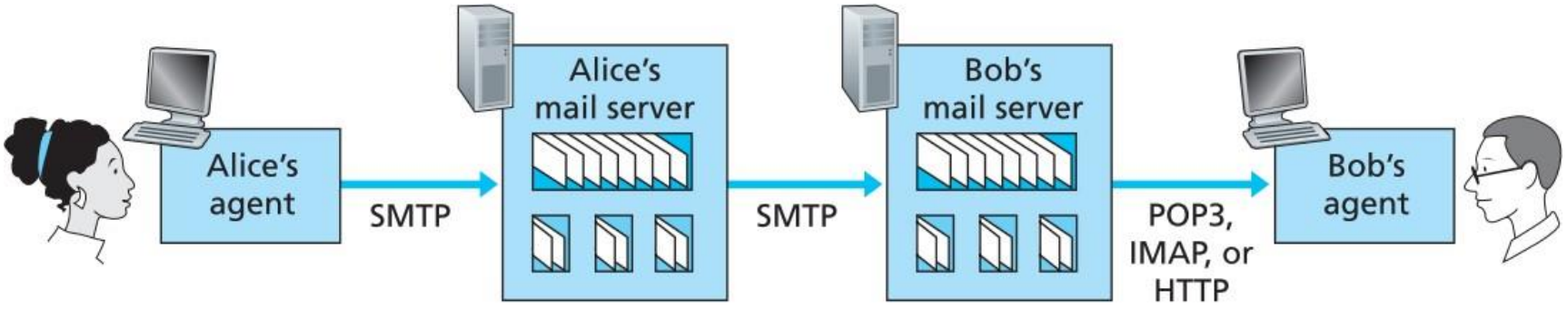
- HTTP: pull
- SMTP: push
- cùng giao tiếp dùng ASCII command/response, status code
- HTTP: mỗi object chứa trong bản tin riêng
- SMTP: nhiều object được gửi trong một bản tin có nhiều phần (multipart message)

Ví dụ giao tiếp SMTP

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

Note: S: Server, C: Client

Giao thức truy cập thư điện tử

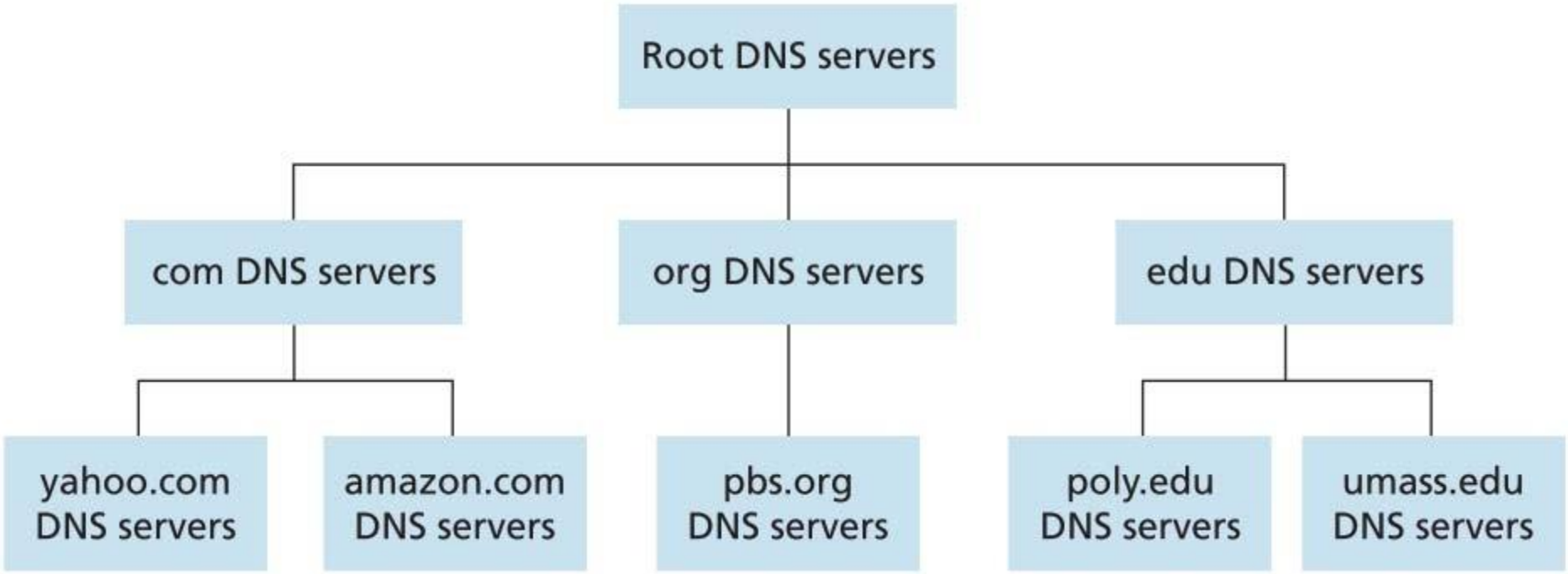


- SMTP: chuyển email tới server của người nhận
- Giao thức truy cập thư điện tử (mail access protocol): lấy email từ server
  - POP: Post Office Protocol [RFC 1939]: xác thực, tải về
  - IMAP: Internet Mail Access Protocol [RFC 1730]: có nhiều thao tác hơn, ví dụ thao tác với email trên server
  - HTTP: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

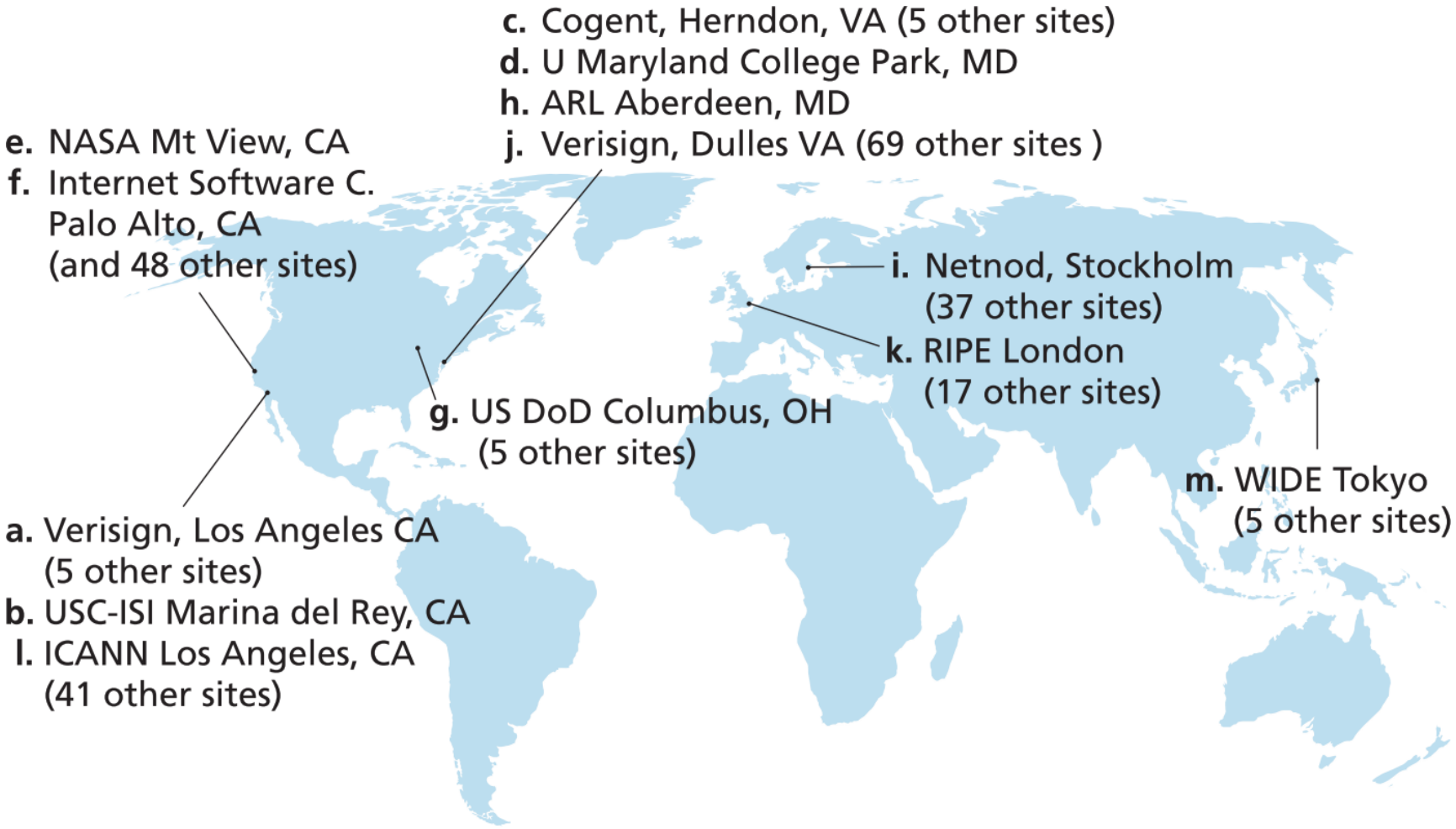
Chương 2: Tầng ứng dụng

- Đặc điểm của ứng dụng mạng
- Web và HTTP
- Truyền tập tin: FTP
- Thư điện tử
- Hệ thống tên miền: DNS
- Ứng dụng ngang hàng

Cơ sở dữ liệu phân cấp và phân tán

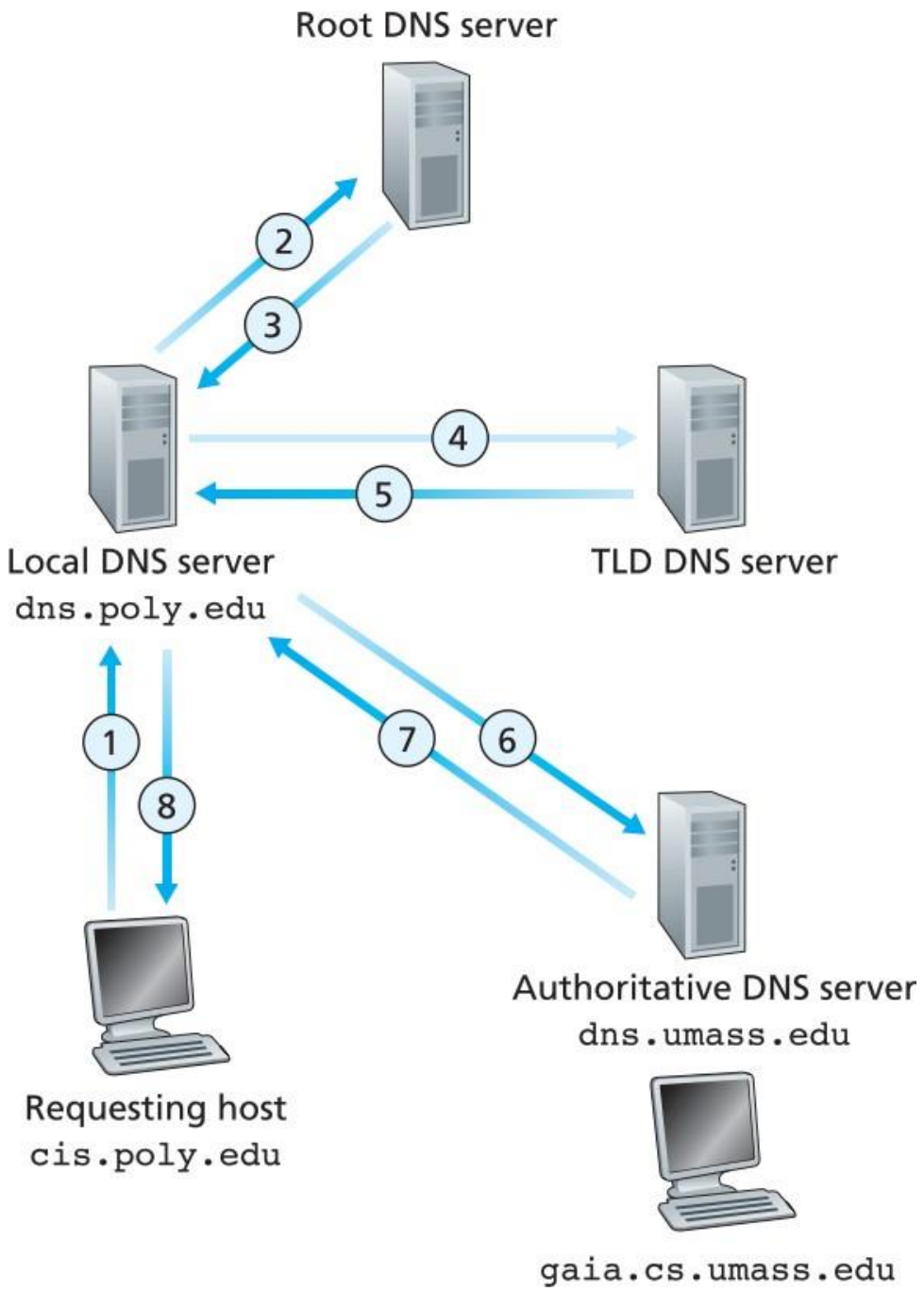


Cơ sở dữ liệu phân cấp và phân tán



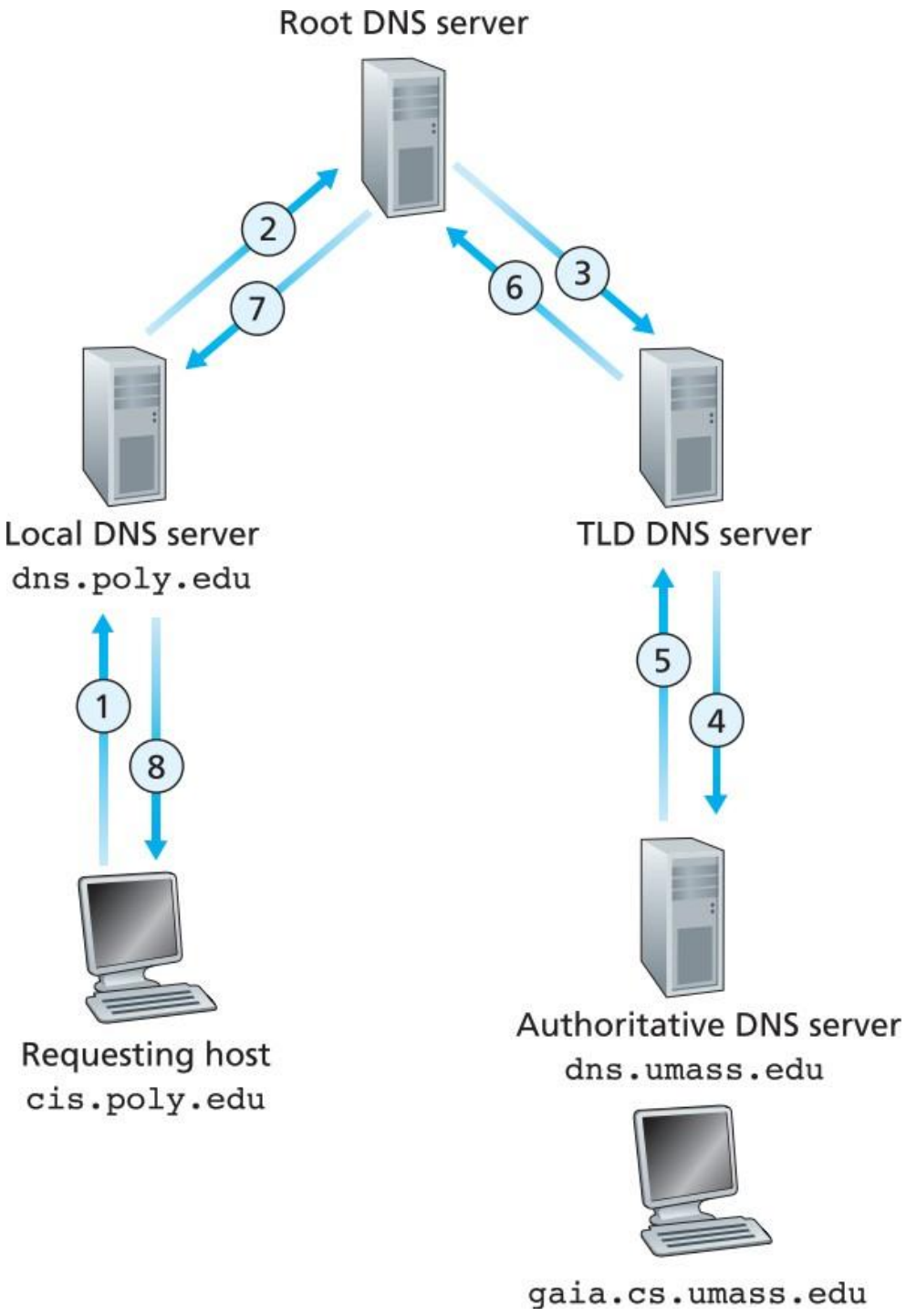
Giao tiếp giữa DNS server

- Yêu cầu lặp (iterated query): 2-3, 4-5, 6-7



Giao tiếp giữa DNS server

- Yêu cầu đệ quy (Recursive queries)



Hoạt động của DNS

- Khi name server nhận biết được mapping, name server sẽ lưu (*cache)* mapping
  - Thông tin lưu trong cache entry, và bị xóa sau một khoảng thời gian (TTL)
  - TLD server thường được lưu trong các local name server
    - Vì vậy root name server sẽ không thường xuyên được truy vấn
- Các cached entry có thể không được cập nhật
  - Nếu host name thay đổi địa chỉ IP của nó, địa chỉ IP này có thể không được cập nhật tới khi TTL quá hạn
- Cơ chế cập nhật: RFC 2136

Bản ghi DNS

DNS: Cơ sở dữ liệu phân tán chứa các bản ghi tài nguyên (resource records (RR)

RR format: **(name, value, type, ttl)**

type=A

- **name** là hostname
- **value** là IP address

type=NS

- **name** là domain (ví dụ: foo.com)
- **value** là hostname của authoritative name server của domain

type=CNAME

- **name** là alias name cho một số "canonical" (real) name
- **www.ibm.com** thực tế là **servereast.backup2.ibm.com**
- **value** là canonical name

type=MX

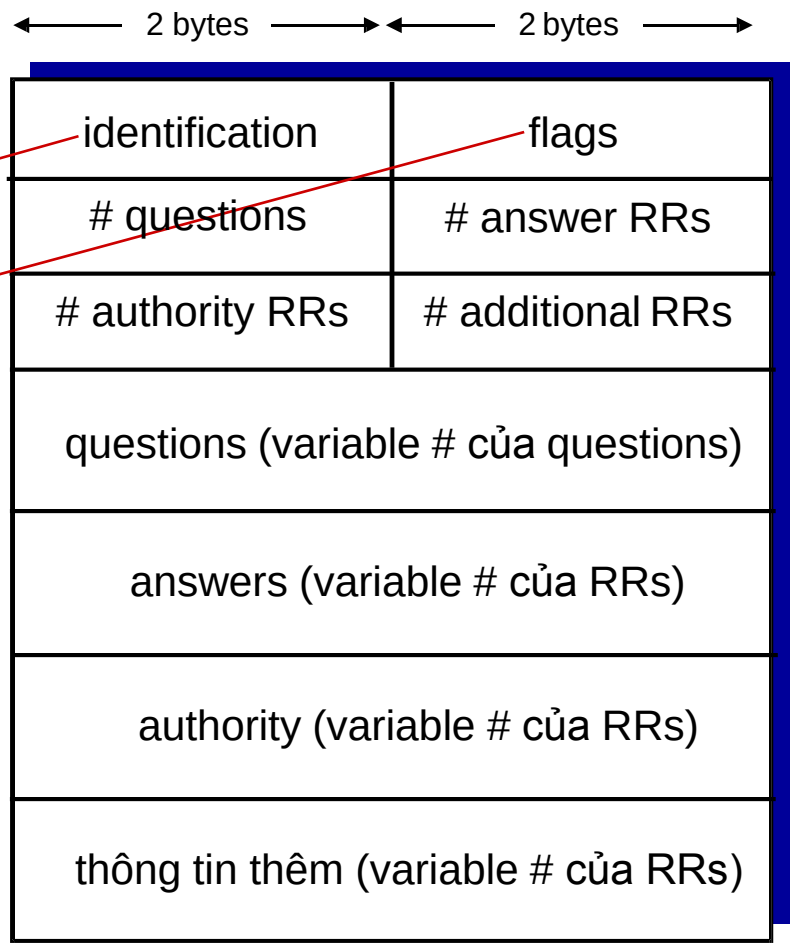
- **value** là tên của mailserver liên kết với **name**

Giao thức DNS protocol, bản tin DNS

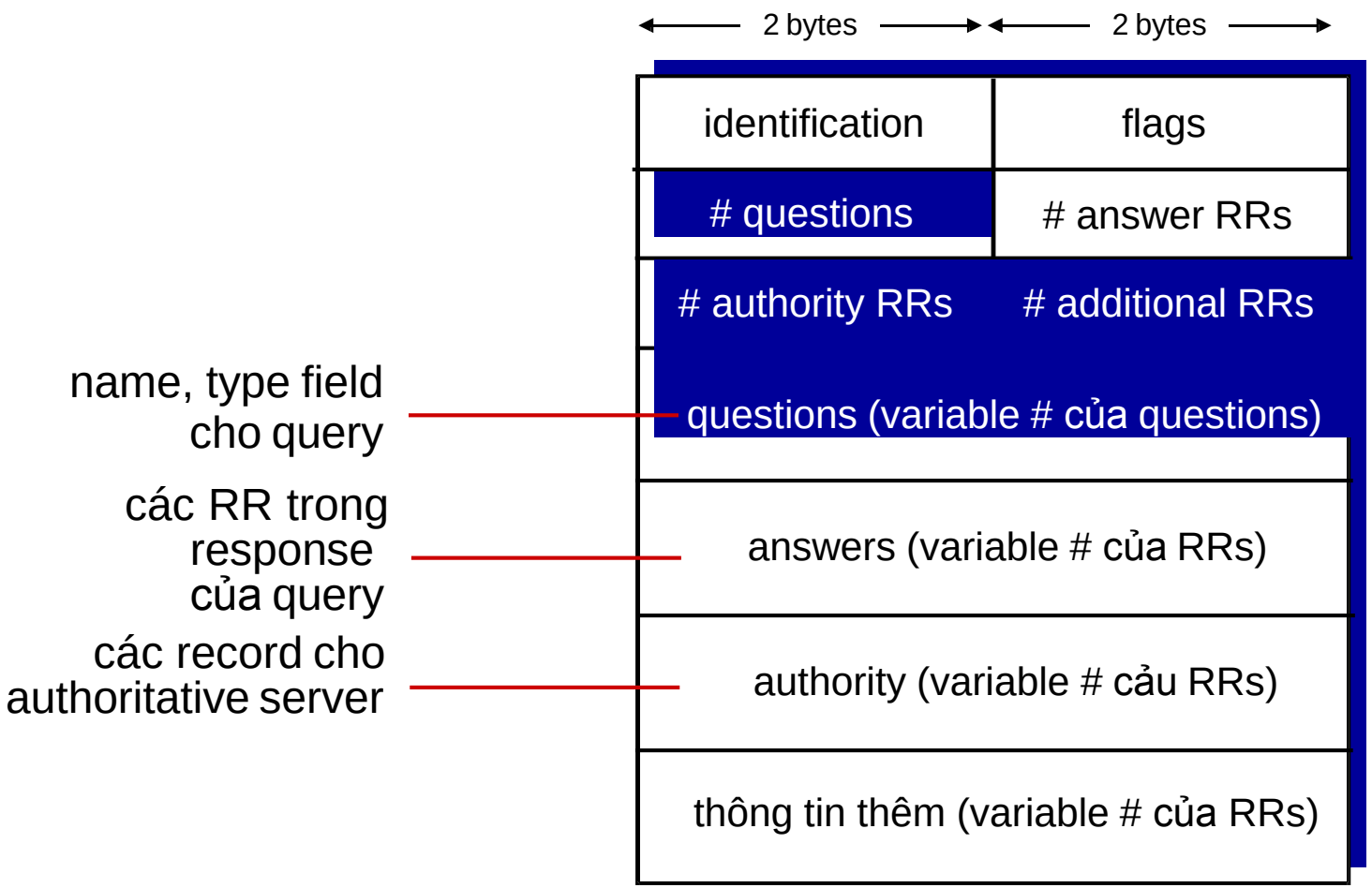
- *query* message và *reply* message, có cùng cấu trúc

Message header

- identification: 16 bit # cho query, reply của query có cùng giá trị id
- flag:
  - query hoặc reply
  - recursion desired
  - recursion available
  - authoritative reply



Giao thức DNS protocol, bản tin DNS



Chèn bản ghi vào DNS

- ví dụ: tổ chức mới “Network Utopia”
- đăng kí tên networkuptopia.com tại *DNS registrar* (ví dụ, Network Solutions)
  - cung cấp các name, các IP address của authoritative name server (primary và secondary)
  - registrar chèn 2 RR vào .com TLD server:
    (networkutopia.com, dns1.networkutopia.com, NS)
    (dns1.networkutopia.com, 212.212.212.1, A)
- tạo authoritative server type A record cho www.networkuptopia.com; type MX record cho networkutopia.com

Tấn công DNS

Tấn công DDoS

- làm root server quá tải bằng cách tăng lưu lượng
  - hiện tại đã bị ngăn chặn
  - Traffic Filtering
  - các Local DNS servers lưu các IP của các TLD server
- làm TLD server quá tải

Tấn công Redirect

- Man-in-middle
  - chặn query
- DNS poisoning
  - gửi rely giả tới DNS server, which caches

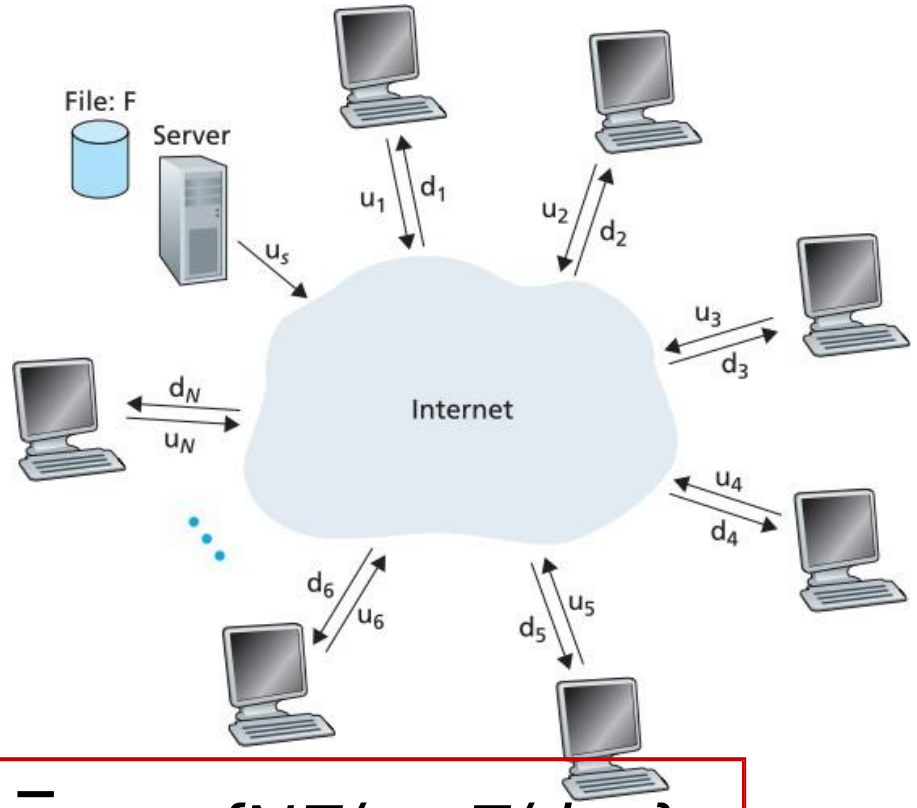
Chương 2: Tầng ứng dụng

- Đặc điểm của ứng dụng mạng
- Web và HTTP
- Truyền tập tin: FTP
- Thư điện tử
- Hệ thống tên miền: DNS
- Ứng dụng ngang hàng

Khả năng tùy biến quy mô của kiến trúc P2P

- *server transmission:* phải tuần tự gửi (upload) N bản sao của 1 file:
  - time để gửi 1 file: F/u_s
  - thời gian để gửi N bản sao: NF/u_s

- *client:* mỗi client phải tải bản sao của file copy
  - d_{min} = tốc độ tải nhỏ nhất của client
  - Thời gian download nhanh nhất có thể để N client để tải file F về: F/d_{min}



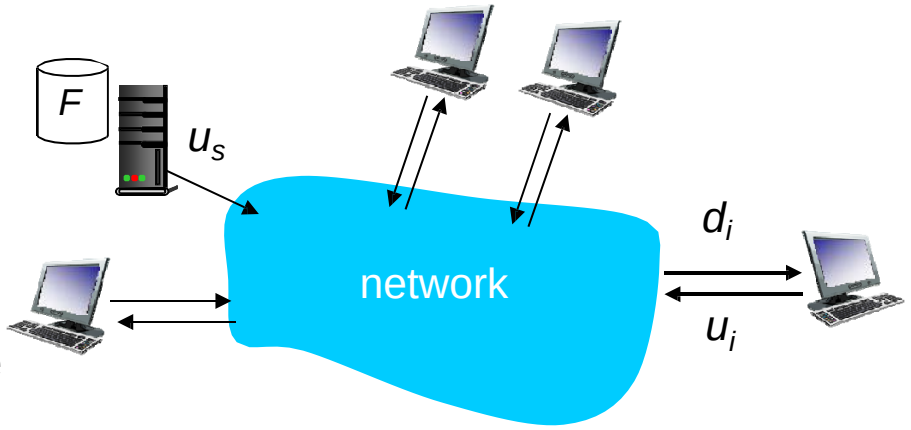
thời gian để gửi F tới N client dùng kiến trúc client-server

$$D_{c\text{-}s} \geq max\{NF/u_s, F/d_{min}\}$$

tăng tuyến tính theo N

Khả năng tùy biến quy mô của kiến trúc P2P

- *server transmission:* phải upload ít nhất 1 bản sao
  - thời gian để gửi 1 bản sao: F/u_s
- *client:* mỗi client phải tải bản sao của file
  - Thời gian download nhanh nhất có thể để N client để tải file F về: F/d_{min}



- *clients:* cùng download *NF* bit
  - max upload rate (giới hạn max download rate): $u_s + \Sigma u_i$

thời gian để chuyển F tới N client sử dụng kiến trúc P2P

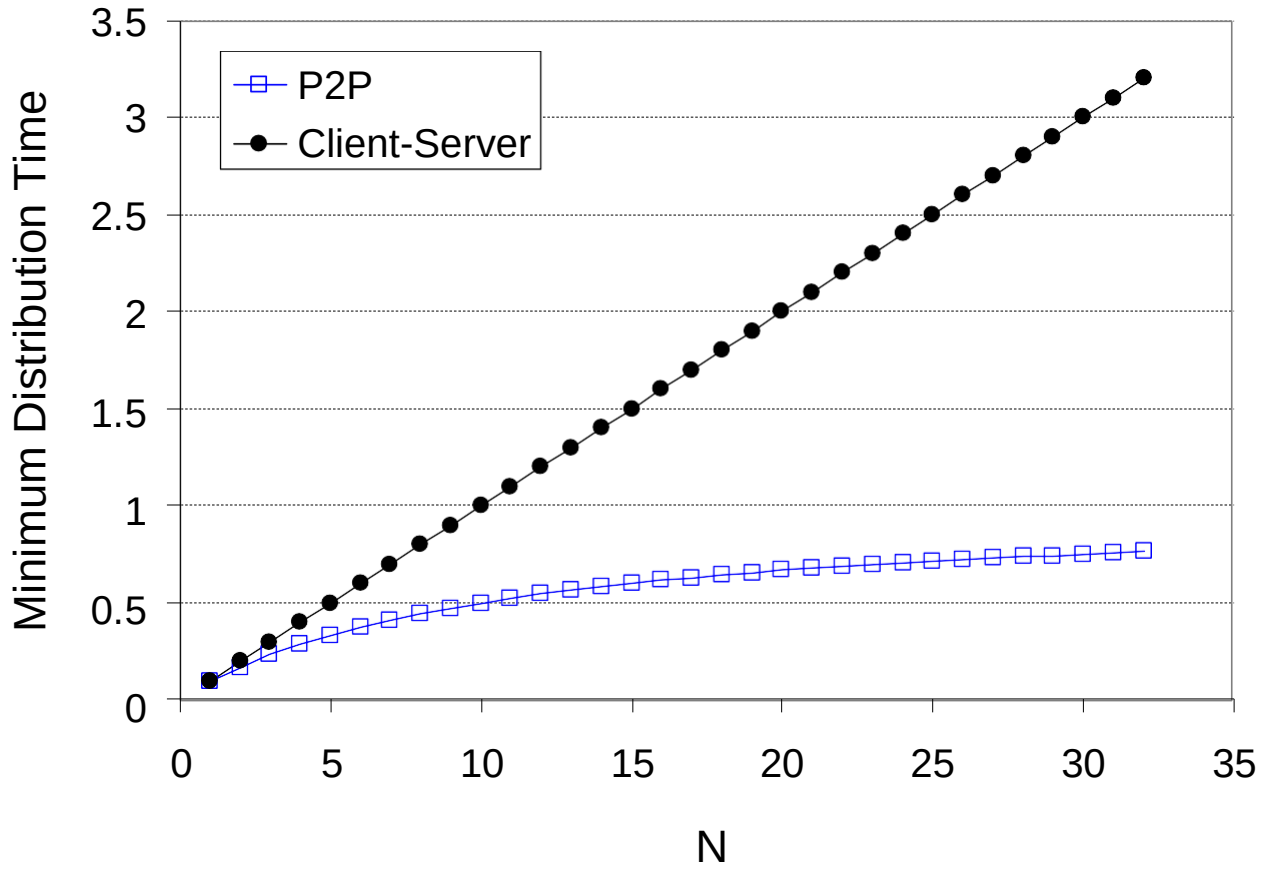
$$D_{P2P} > max\{F/u_s, F/d_{min}, NF/(u_s + \Sigma u_i)\}$$

tăng tuyến tính với *N* …

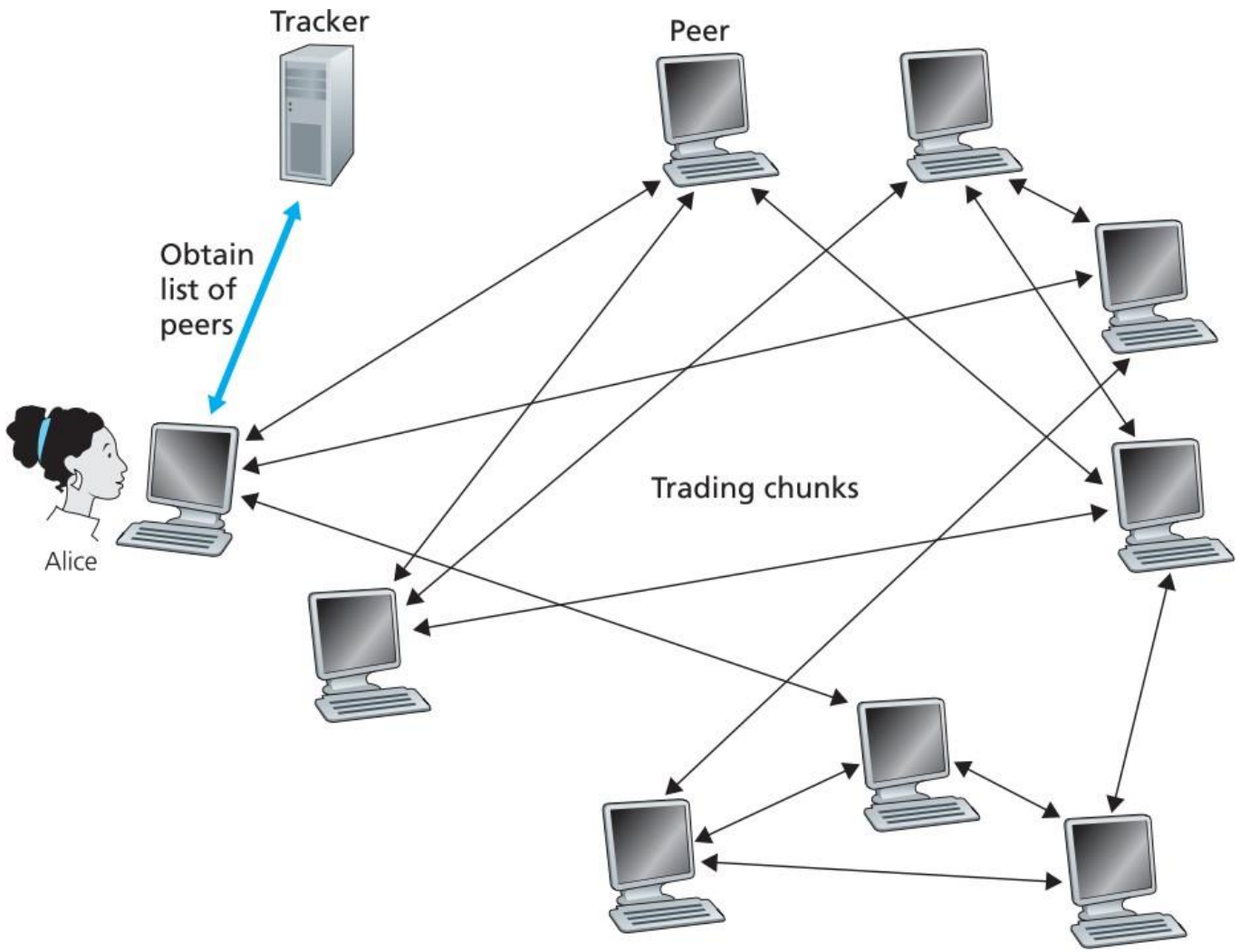
… nhưng client cung cấp thêm capacity

Khả năng tùy biến quy mô của kiến trúc P2P

tốc độ upload của client = u, F/u = 1 giờ, $u_s = 10u$, $d_{min} \geq u_s$



Truyền file dùng BitTorrent



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
  - P2P

Tóm tắt

- trao đổi bản tin yêu cầu và bản tin trả lời:
  - client yêu cầu thông tin hoặc dịch vụ
  - server trả lời với dữ liệu và mã trạng thái
- cấu trúc của bản tin:
  - header: các trường cung cấp thông tin về dữ liệu (data)
  - data: thông tin giao tiếp
- *một số nguyên tắc*
  - bản tin dữ liệu và điều khiển: in-band, out-of-band
  - tập trung vs. phân tán
  - không trạng thái (stateless) vs. có trạng thái (stateful)
  - truyền tin cậy (reliable transfer) vs. truyền không tin cậy

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