



MẠNG MÁY TÍNH



CHƯƠNG 6 TẦNG ỨNG DỤNG

Khoa CNTT – HV KTMM

CHƯƠNG 6 – BÀI 1

1

GIAO THỨC VÀ DỊCH VỤ TẦNG ỨNG DỤNG

2

HTTP VÀ DỊCH VỤ WEB

3

FTP VÀ DỊCH VỤ FILE

4

DỊCH VỤ THƯ ĐIỆN TỬ - EMAIL

5

TỔNG KẾT

Nhắc lại kiến trúc phân tầng TCP-IP

Application (HTTP, Mail, ...)
Transport (UDP, TCP ...)
Network (IP, ICMP...)
Datalink (Ethernet, ADSL...)
Physical (bits...)

❖ Tầng ứng dụng:

➤ Hỗ trợ các ứng dụng trên mạng

Ứng dụng và giao thức tầng ứng dụng

❖ Trình ứng dụng: *các tiến trình phân tán truyền thông với nhau*

➤ Chạy trên các thiết bị đầu cuối

➤ Trao đổi thông điệp

➤ Web, mail, FTP,...

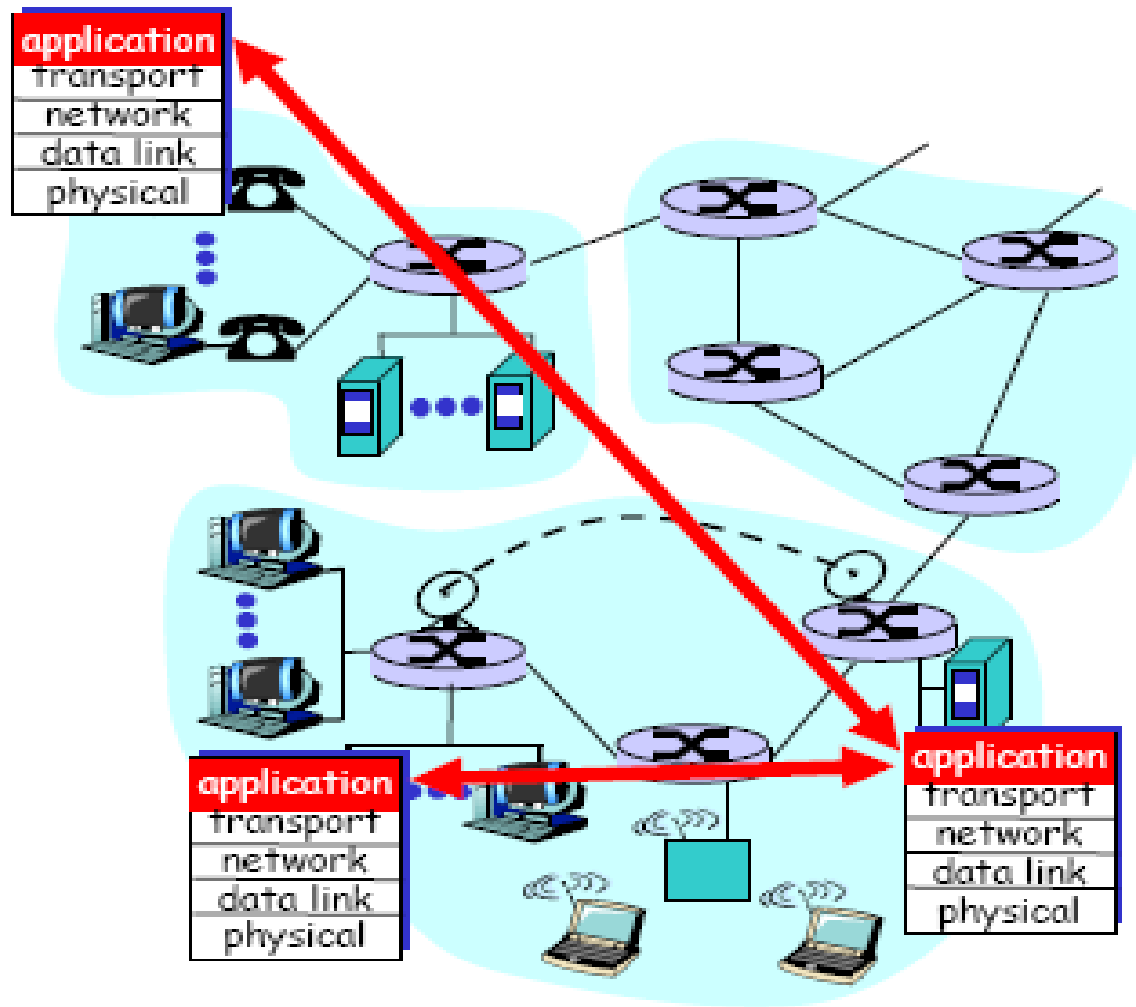
❖ Giao thức tầng ứng dụng:

➤ Là một phần của trình ứng dụng

➤ Xác định *cách thức trao đổi thông điệp giữa các ứng dụng*

➤ *Sử dụng dịch vụ truyền thông do tầng dưới cung cấp*

Ứng dụng và giao thức tầng ứng dụng



Một số thuật ngữ

❖ Tiến trình

- Là một *chương trình chạy* trên thiết bị đầu cuối
- Trên cùng một máy, *các tiến trình có thể truyền thông với nhau do hệ điều hành quản lý*
- Các *tiến trình phân tán, chạy trên các máy khác nhau tuân thủ giao thức tầng ứng dụng*

❖ User Agent

- Phần mềm đóng vai trò *giao diện giữa người dùng và mạng*
- Web-browser
- Mail-mail reader

Mô hình client-server

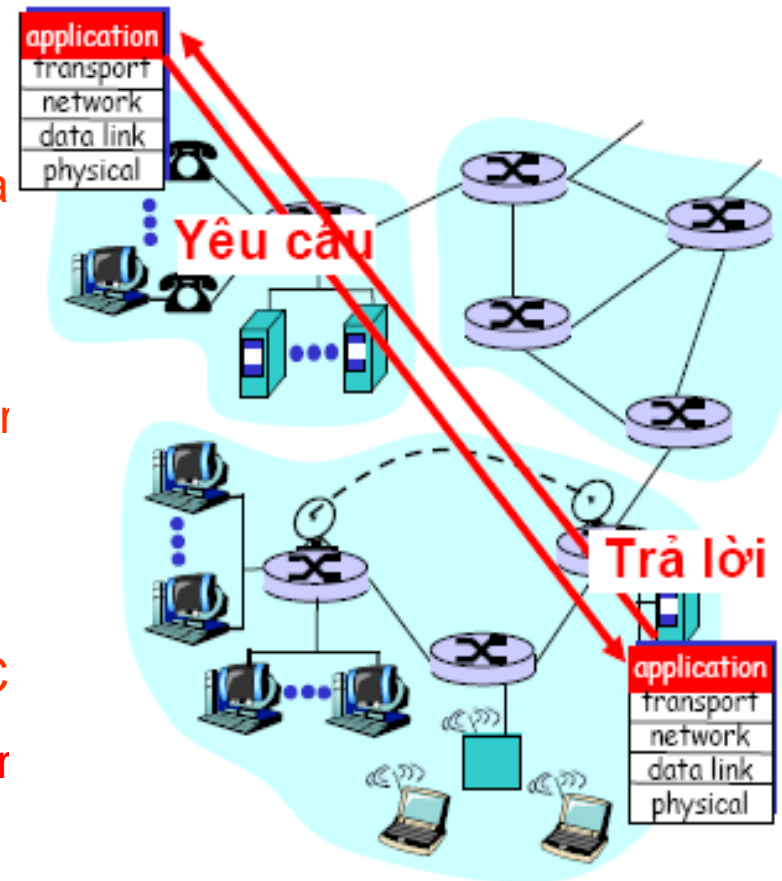
❖ Ứng dụng chia thành **2 thành phần**
client, server

➤ Client:

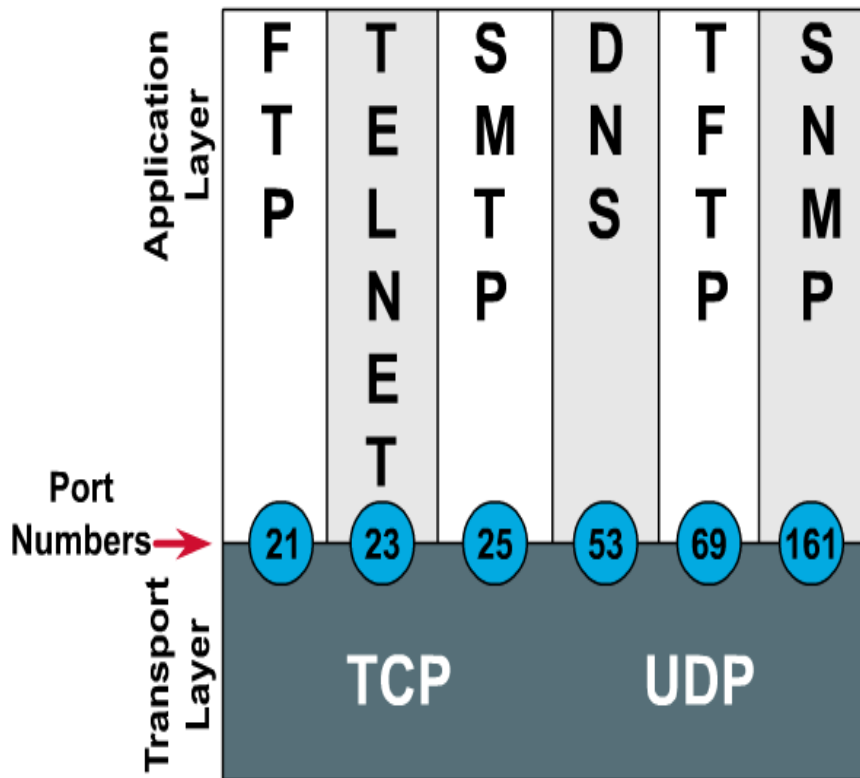
- ✓ Khởi tạo giao tiếp với server (“speak first”)
- ✓ Yêu cầu dịch vụ từ server
- ✓ Web browser hiển thị dữ liệu do server gửi về: page, email,...

➤ Server:

- ✓ Cung cấp dịch vụ theo yêu cầu từ C
- ✓ Web server gửi Web page, Mail server gửi và nhận E-mail



Giao thức tầng ứng dụng



❖ API: Application Programming Interface

➤ Cung cấp giao diện giữa tầng ứng dụng và tầng giao vận

➤ **Socket = IP + cổng dịch vụ**

✓ Các tiến trình trao đổi dữ liệu bằng cách gửi/nhận thông điệp từ socket

Dịch vụ giao vận cần thiết cho ứng dụng

❖ Mất mát dữ liệu

- Ứng dụng có thể chấp nhận mất mát dữ liệu
- Ứng dụng yêu cầu 100% dữ liệu tin cậy

❖ Thời gian

- Ứng dụng thời gian thực: game online, hội thoại, ...
yêu cầu thời gian trễ thấp

Yêu cầu với các ứng dụng điển hình

Ứng dụng	Mất mát	Băng thông	Thời gian
Truyền file	Không	Co dẫn	Không
Thư tín điện tử	Không	Co dẫn	Không
Web	Có thể	Co dẫn	Không
Đa phương tiện	Có thể	audio: 5Kb-1Mb video: 10Kb-5Mb	Có, 100's msec
Thời gian thực stored audio/video	Chấp nhận	same as above	Có, few secs
Trò chơi tương tác	Chấp nhận	few Kbps up	Có, 100's msec
Ứng dụng Tài chính	Không	Co dẫn	Không xác định

Các ứng dụng và giao thức giao vận tương ứng

Ứng dụng	Giao thức Ứng dụng	Giao thức Giao vận tương ứng
Thư tín Điện tử	SMTP [RFC 821]	TCP
Truy cập từ xa	telnet [RFC 854]	TCP
Web	HTTP [RFC 2068]	TCP
Truyền file	FTP [RFC 959]	TCP
Đa phương tiện luồng	Độc quyền (vd. RealNetworks)	TCP hoặc UDP
file server ở xa	NSF	TCP hoặc UDP
Điện thoại Internet	Độc quyền (vd. Vocaltec)	Thường là UDP

CHƯƠNG 6 – BÀI 1

1 GIAO THỨC VÀ DỊCH VỤ TẦNG ỨNG DỤNG

2 HTTP VÀ DỊCH VỤ WEB

3 FTP VÀ DỊCH VỤ FILE

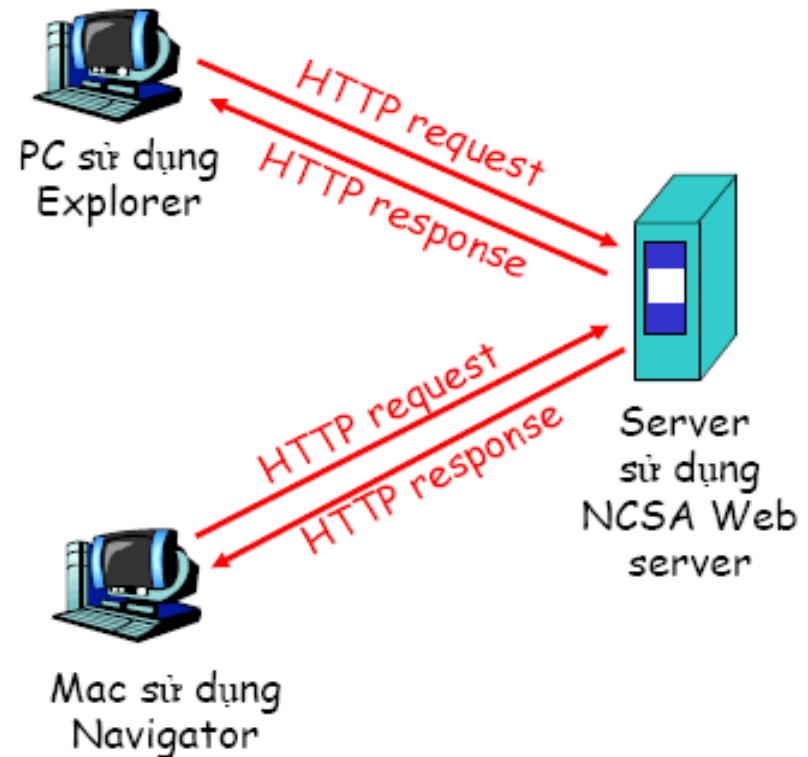
4 DỊCH VỤ THƯ ĐIỆN TỬ - EMAIL

5 TỔNG KẾT

Case study1: Web – giao thức HTTP

❖ Giao thức HTTP – Hyper Text Tranfer Protocol

- Là giao thức dùng cho các ứng dụng web
- Sử dụng mô hình client-server
 - ✓ Client: Browser gửi yêu cầu, nhận, hiển thị các đối tượng web
 - Browser: IE, FireFox, Netscape, Google Chrome, v.v.
 - ✓ Server: webserver gửi các đối tượng khi có yêu cầu
 - Web server: IIS, Apache, Ngnix, v.v.



Giao thức HTTP

- ❖ Client khởi tạo kết nối TCP (socket) tới server thông qua **cổng 80 (cổng web mặc định)**
- ❖ Server chấp nhận kết nối TCP từ client gửi đến
- ❖ Các thông điệp HTTP trao đổi giữa browser-Web server
- ❖ Đóng kết nối TCP

Ví dụ về HTTP

đánh địa chỉ trên trình duyệt

www.someSchool.edu/someDepartment/home.index

(bao gồm file HTML
tham chiếu tới 10
ảnh JPEG)

1a. HTTP client khởi tạo kết nối
TCP tới HTTP server tại địa
chỉ **www.someSchool.edu**.
Công mặc định là **80**.

1b. HTTP server ở địa chỉ
www.someSchool.edu đợi
kết nối TCP ở cổng **80**, chấp
nhận kết nối, thông báo lại
cho client.

2. HTTP client gửi thông điệp
HTTP yêu cầu (bao gồm
URL) qua kết nối TCP vừa
thiết lập

3. HTTP server nhận thông điệp yêu
cầu, lấy các đối tượng được yêu
cầu gửi vào trong thông điệp trả
lời,
(**someDepartment/home.index**)
gửi thông điệp vào socket

Thời gian
↓

Ví dụ về HTTP


Thời gian



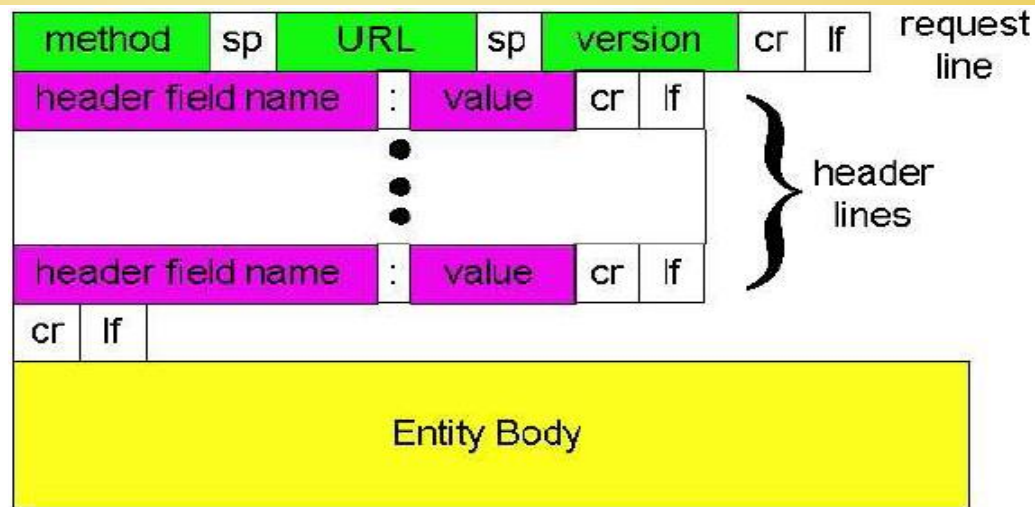
5. HTTP client nhận thông điệp trả lời bao gồm tệp html, hiển thị html. Phân tích tệp html file, tìm 10 jpeg đối tượng được tham chiếu

6. Các bước từ 1 đến 5 được lặp lại cho từng đối tượng trong 10 đối tượng jpeg

4. HTTP server đóng kết nối TCP.



Định dạng thông điệp yêu cầu HTTP



Dòng yêu cầu
(Lệnh GET, POST,
HEAD)

Các dòng header

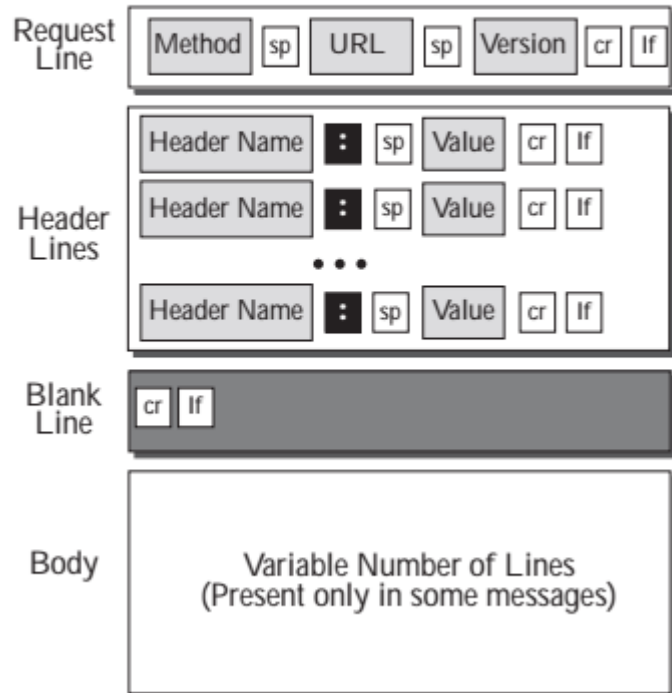
```
GET /somedir/page.html HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language: fr
```

CR,LF: Kết
thúc thông điệp

(CR,LF)

Định dạng thông điệp yêu cầu HTTP

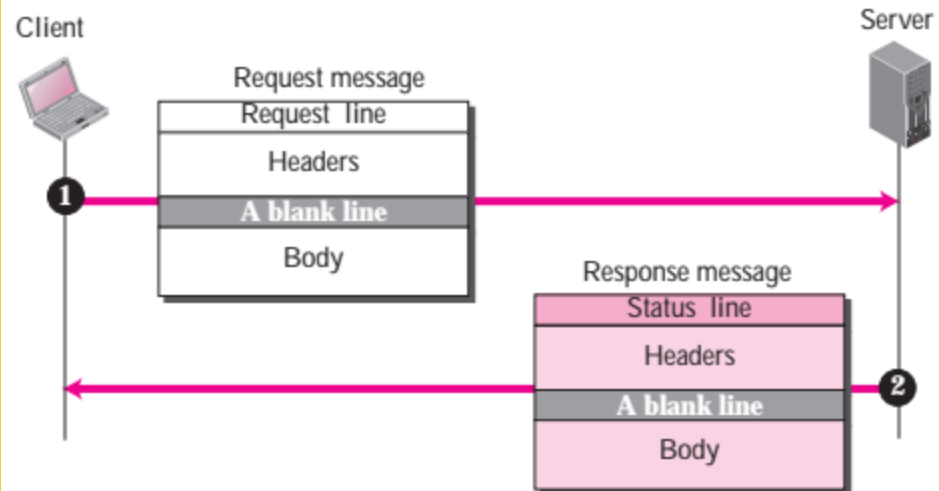
Format of the request message



Legend

sp: Space
cr: Carriage Return
lf: Line Feed

HTTP transaction



Định dạng thông điệp yêu cầu HTTP

- **Method**

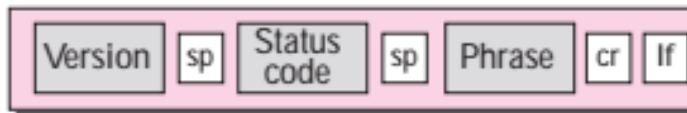
<i>Method</i>	<i>Action</i>
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from the server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
DELETE	Remove the Web page
OPTIONS	Enquires about available options

- **Request Header Names:**

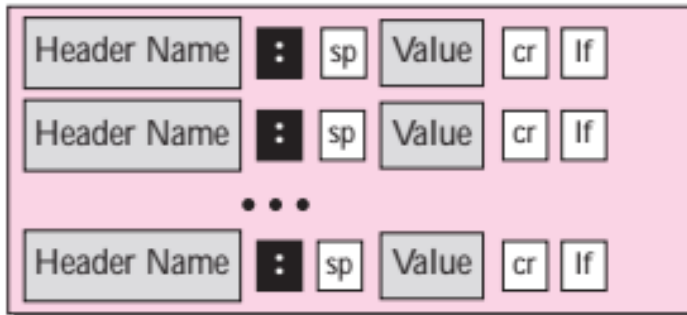
<i>Header</i>	<i>Description</i>
User-agent	Identifies the client program
Accept	Shows the media format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows what permissions the client has
Host	Shows the host and port number of the client
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Cookie	Returns the cookie to the server
If-Modified-Since	Returns the cookie to the server

Định dạng thông điệp trả lời

Status Line



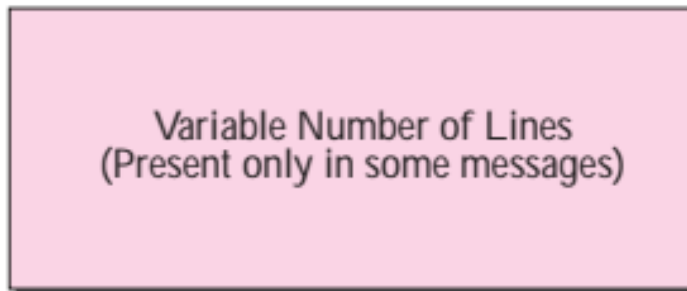
Header Lines



Blank Line



Body



Legend

sp: Space
cr: Carriage Return
lf: Line Feed

Định dạng thông điệp trả lời

Table 22.4 *Response Header Names*

<i>Header</i>	<i>Description</i>
Date	Shows the current date
Upgrade	Specifies the preferred communication protocol
Server	Gives information about the server
Set-Cookie	The server asks the client to save a cookie
Content-Encoding	Specifies the encoding scheme
Content-Language	Specifies the language
Content-Length	Shows the length of the document
Content-Type	Specifies the media type
Location	To ask the client to send the request to another site
Accept-Ranges	The server will accept the requested byte-ranges
Last-modified	Gives the date and time of the last change

Định dạng thông điệp trả lời

Dòng trạng thái
(mã trạng thái)

HTTP/1.0 200 OK

Các dòng tiêu đề

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 (chẳng hạn
file html được
yêu cầu)

data data data data data ...

Mã trạng thái trong thông điệp trả lời

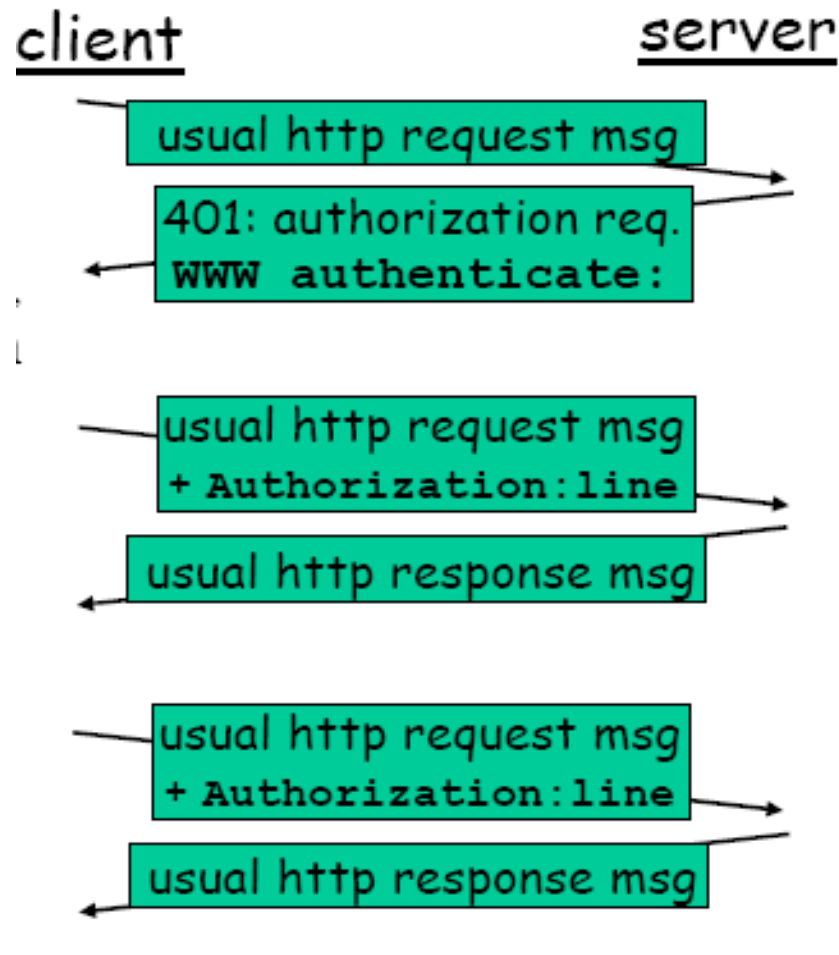
- ❖ Được ghi ở dòng đầu tiên trong thông điệp trả lời của webserver
- ❖ Một số dạng mã
 - **200 OK**: yêu cầu thành công các yêu cầu trong thân thông điệp
 - **301 Moved Permanently**: đối tượng yêu cầu đã được gửi, địa chỉ mới của đối tượng đặt trong trường Location
 - **400 Bad Request**: server không hiểu được thông điệp yêu cầu
 - **404 Not Found**: đối tượng yêu cầu không có trong server
 - **505 HTTP Version Not Supported**: server không hỗ trợ phiên bản giao thức HTTP

Xác thực người dùng

- ❖ Mục tiêu: kiểm soát quyền truy cập các đối tượng lưu trên server.
- ❖ Phương pháp: sử dụng Tên truy cập/mật khẩu
- ❖ Không trạng thái: mỗi lần yêu cầu, client phải chứng minh mình có quyền
- ❖ Kiểm chứng:
 - Tiêu đề authorization: trong mỗi yêu cầu gửi đi
 - Nếu không có quyền, server từ chối truy cập và gửi yêu cầu WWW authenticate
- ❖ Trình duyệt thường cache username/password để user không phải gõ lại trong mỗi lần truy nhập

Xác thực người dùng

❖ Các bước:



Lưu dữ liệu tại client - Cookies

- Do Server tạo ra, lưu lại để sử dụng về sau:

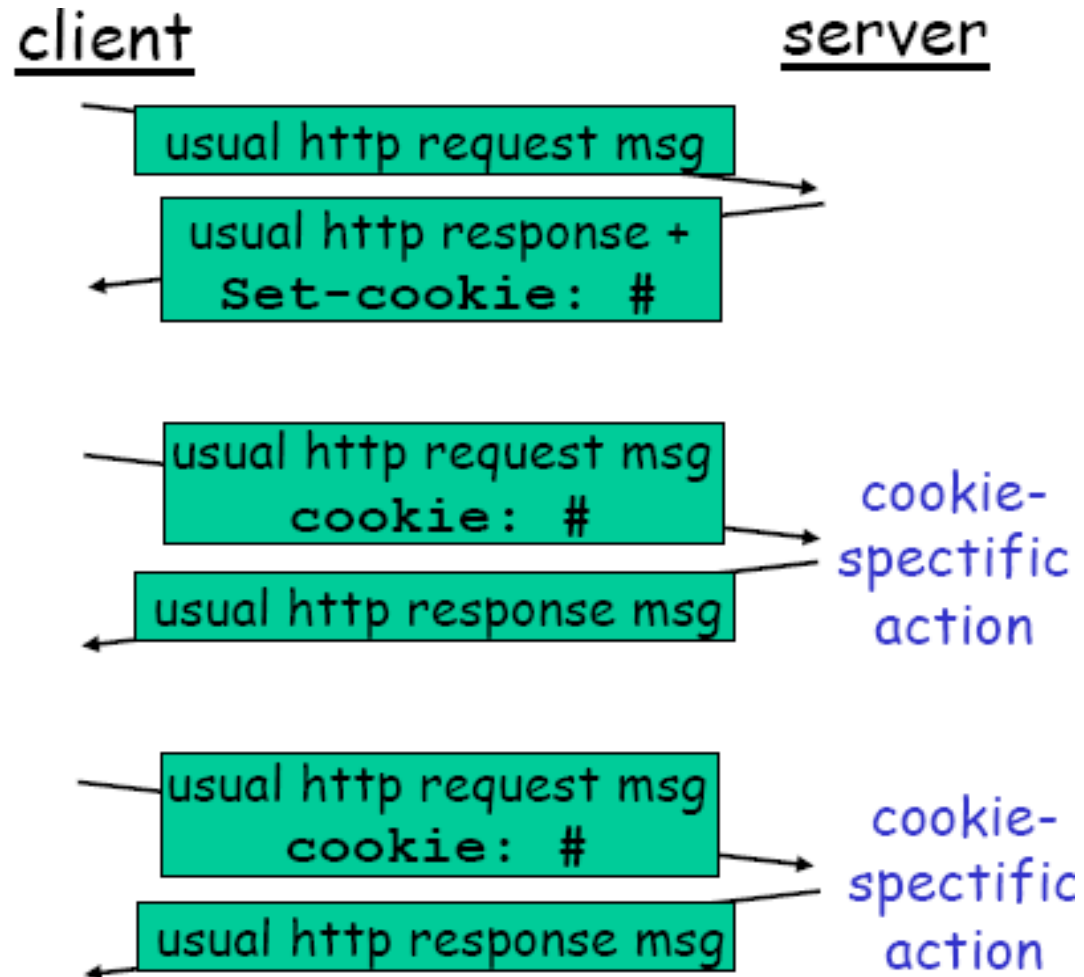
- Thẩm định quyền
- Ghi nhớ các “sở thích”, hoạt động của người dùng.

- Server gửi “cookie” tới client trong thông điệp response.

Set-cookie: 1678453

- Client gửi kèm cookie trong những lần yêu cầu sau.

cookie: 1678453



GET có điều kiện – caching client

❖ Mục đích:

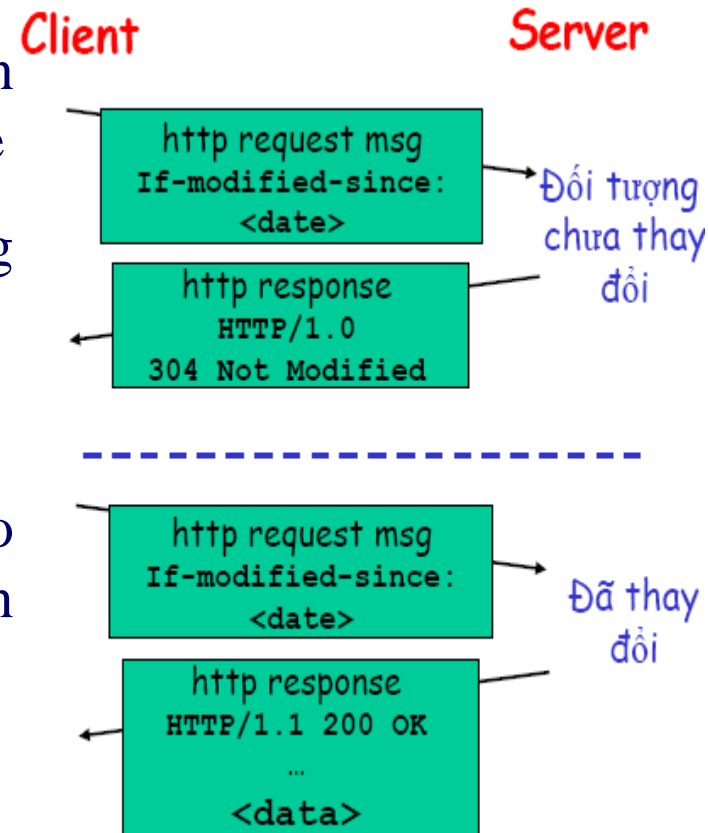
- Không gửi lại nếu client đã có phiên bản mới nhất của đối tượng trong cache

❖ Client: chỉ ra ngày lấy đối tượng trong thông điệp yêu cầu HTTP request

If-modified-since: <date>

❖ Server: thông điệp trả về không kèm theo đối tượng nếu nó không được sửa đổi trên server

HTTP/1.0 304 Not modified

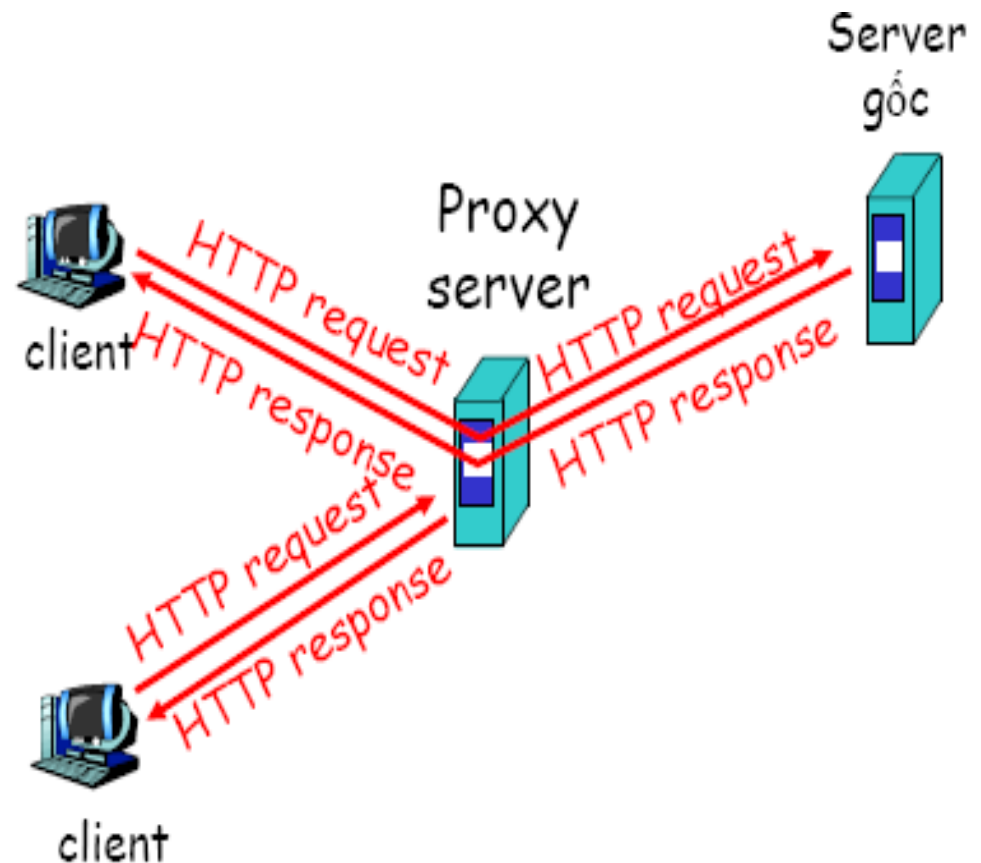


Web Caches (Proxy Server)

Người dùng cấu hình browser truy nhập Web thông qua Web Cache

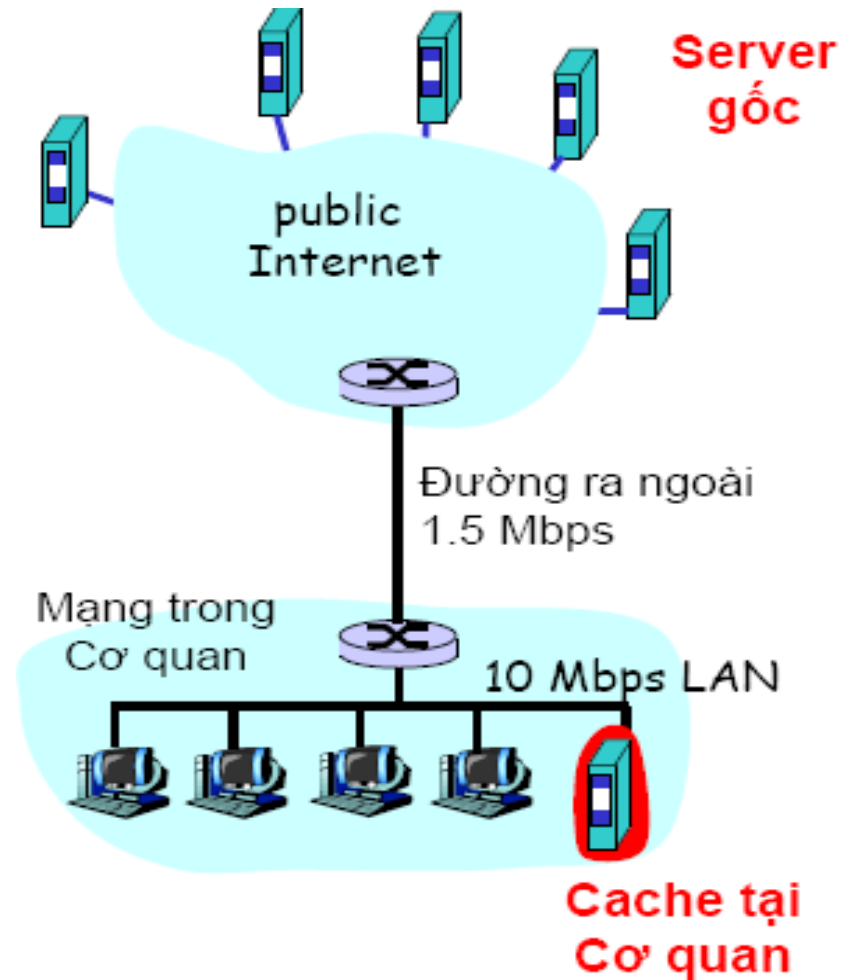
Client gửi tất cả các yêu cầu HTTP request tới Web Cache

- **Nếu có đối tượng**, Web Cache gửi đối tượng về cho client.
- **Nếu không có**, Web Cache yêu cầu đối tượng từ server thực, sau đó gửi đối tượng cho client



Lý do sử dụng Web Caches

- ❖ Cache gần client hơn
 - Có thể trong cùng LAN
- ❖ Giảm thời gian client phải đợi
- ❖ Giảm tải mạng
 - Đường nối từ mạng cục bộ ra ngoài thường xảy ra **Thất cổ chai**



CHƯƠNG 6 – BÀI 1

1 GIAO THỨC VÀ DỊCH VỤ TẦNG ỨNG DỤNG

2 HTTP VÀ DỊCH VỤ WEB

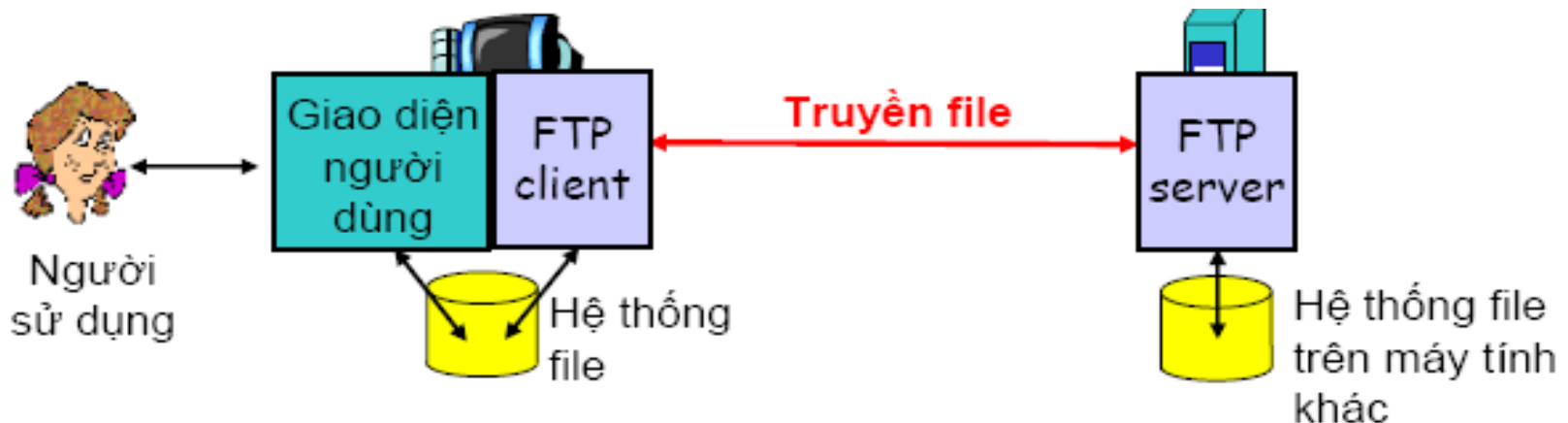
3 FTP VÀ DỊCH VỤ FILE

4 DỊCH VỤ THƯ ĐIỆN TỬ - EMAIL

5 TỔNG KẾT

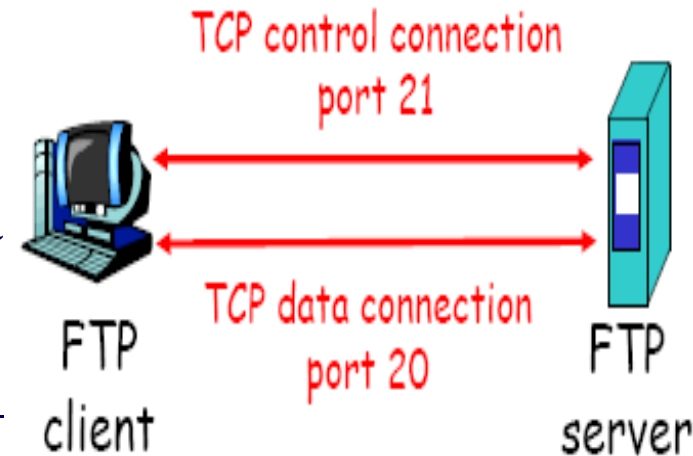
Case study2: Giao thức truyền file FTP

- ❖ Giao thức trao đổi file với máy tính ở xa (remote host)
- ❖ Sử dụng mô hình client-server
 - Client: khởi tạo kết nối
 - Server: máy tính ở xa
 - ✓ Cổng dịch vụ (port): 20/21



Mô tả kết nối FTP

- ❖ FTP client kết nối với FTP server qua cổng 20/21
- ❖ Hai kết nối TCP đồng thời được tạo
 - Trao đổi lệnh phản hồi giữa client và server
 - Dữ liệu: kết nối tải file từ client – server hoặc ngược lại
- ❖ FTP server lưu lại trạng thái truy cập hiện thời, lần truy cập gần đây nhất



Các lệnh và trả lời trong FTP

Các lệnh thường gặp:

- Được mã hoá bằng mã ASCII
- **USER** *username*
- **PASS** *password*
- **LIST** trả về danh sách các file và thư mục trong thư mục hiện thời.
- **RETR filename** lấy file từ thư mục hiện thời.
- **STOR filename** Tải file vào thư mục hiện thời trên máy tính ở xa

Các mã trả về thường gặp

- Tương tự HTTP
- **331** chấp nhận username, yêu cầu password
- **125** kết nối dữ liệu được thiết lập, chuẩn bị truyền dữ liệu.
- **425** Không thể thiết lập kết nối dữ liệu
- **452** Lỗi ghi file.

Ví dụ: Kết nối FTP

```
$ 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.
```

Mô tả kết nối FTP

Table 21.2 *File management commands*

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
CWD	Directory name	Change to another directory
CDUP		Change to parent directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
NLIST	Directory name	List subdirectories or files without attributes
MKD	Directory name	Create a new directory
PWD		Display name of current directory
RMD	Directory name	Delete a directory
RNFR	File name (old)	Identify a file to be renamed
RNTO	File name (new)	Rename the file
SMNT	File system name	Mount a file system

Table 21.3 *Data formatting commands*

<i>Command</i>	<i>Argument(s)</i>	<i>Description</i>
TYPE	A (ASCII), E (EBCDIC), I (Image), N (Nonprint), or T (TELNET)	Define file type
STRU	F (File), R (Record), or P (Page)	Define organization of data
MODE	S (Stream), B (Block), or C (Compressed)	Define transmission mode

CHƯƠNG 6 – BÀI 1

1 GIAO THỨC VÀ DỊCH VỤ TẦNG ỨNG DỤNG

2 HTTP VÀ DỊCH VỤ WEB

3 FTP VÀ DỊCH VỤ FILE

4 DỊCH VỤ THƯ ĐIỆN TỬ - EMAIL

5 TỔNG KẾT

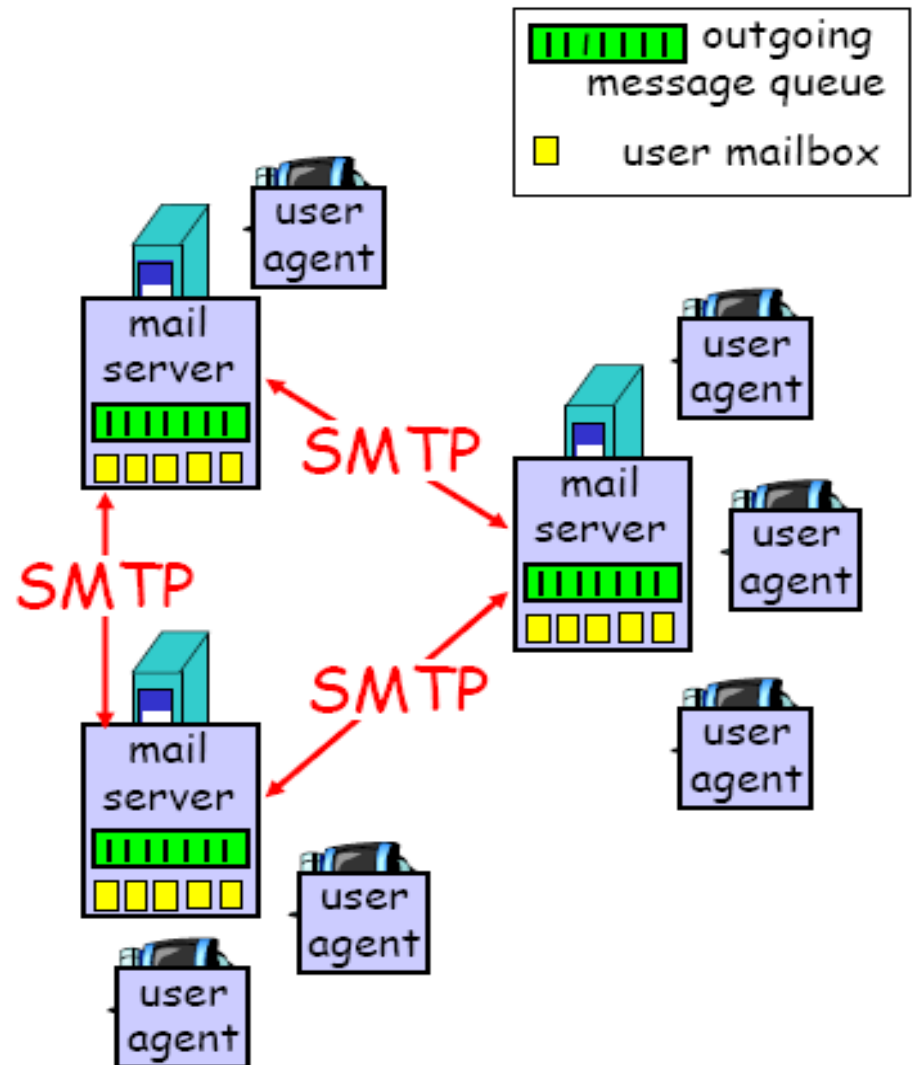
Case study 3: Thư điện tử - Email

3 thành phần chính:

- user agent
- mail server
- Simple Mail Transfer Protocol: **SMTP**

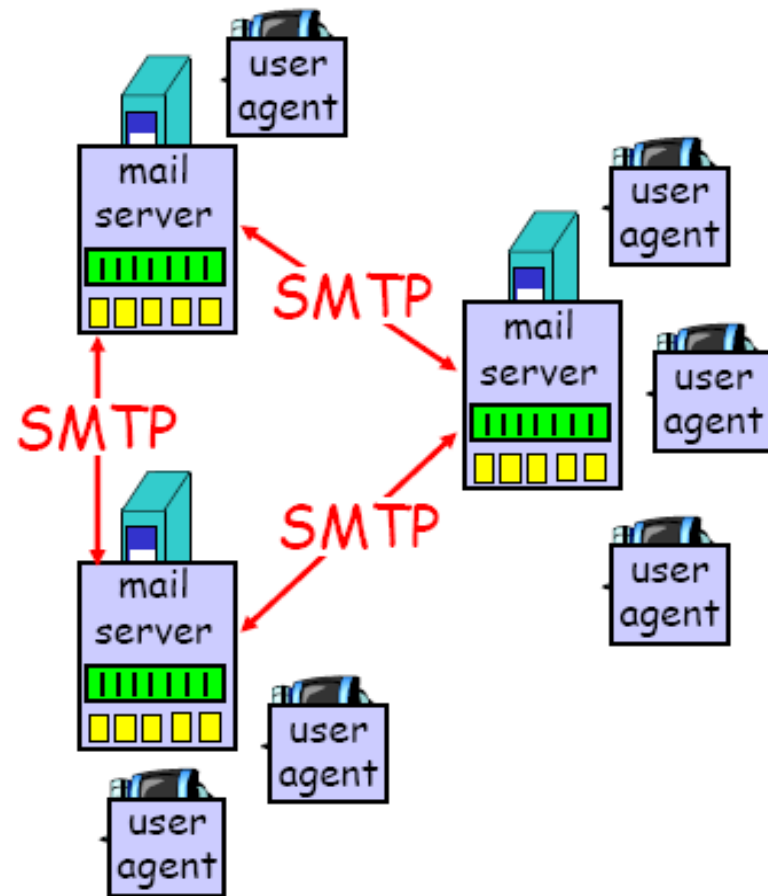
User Agent

- Còn gọi là “mail reader”
- Soạn thảo và Đọc thư
- Ví dụ: Eudora, Outlook, elm, Netscape Messenger
- Các thông điệp outgoing, incoming được lưu trên server



Mail Server [RFC 821]

- ❖ **Lưu thư** của người sử dụng
- ❖ **Hàng đợi** *chứa các thư sẽ được gửi đi*
- ❖ SMTP-Simple Mail Transfer Protocol
 - Các mail-server sử dụng để trao đổi mail
 - Là *client* khi gửi thư
 - Là *server* khi nhận thư



Mail Server [RFC 821]

- ❑ Sử dụng dịch vụ TCP, truyền email tin cậy từ SMTP client tới SMTP server qua cổng 25
- ❑ Truyền trực tiếp : server gửi tới server nhận.
- ❑ Truyền qua ba giai đoạn :
 - “Bắt tay”
 - Truyền các thông điệp (Thư)
 - Đóng kết nối.
- ❑ Tương tác: Lệnh (client => server) và Trả lời (Server => Client)
 - **Lệnh** : mã bằng bảng mã ASCII.
 - **Trả lời** : mã trạng thái và có thể có thêm giải thích
- ❑ Các thông điệp phải được mã bằng bảng mã ASCII 7 bit

Một số lệnh 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|
```

SMTP

- ❑ Giao thức SMTP sử dụng kết nối bền vững.
- ❑ SMTP đòi hỏi thông điệp (header & body) phải được định dạng bằng mã ASCII 7bit.
- ❑ Những chuỗi ký tự đặc biệt (ví dụ CRLF.CRLF) không được phép ghi vào thông điệp (do đó chuỗi ký tự này phải được mã hóa)
- ❑ SMTP server sử dụng CRLF.CRLF để đánh dấu kết thúc thông điệp.

So sánh với HTTP:

- ❑ HTTP: giao thức kiểu “**kéo**” (“kéo” thông tin từ server về)
- ❑ Email :là giao thức kiểu “**đẩy**” (“đẩy” thông tin lên server)
- ❑ Cả hai đều tương tác gửi **lệnh** (mã ASCII)/ **trả lời** (mã trạng thái).
- ❑ HTTP : Mỗi đối tượng nằm trong một thông điệp riêng.
- ❑ SMTP: nhiều đối tượng nằm trong cùng một thông điệp

Định dạng thông điệp Email

SMTP : Giao thức để trao đổi email.

RFC 822 : chuẩn định dạng thông điệp email:

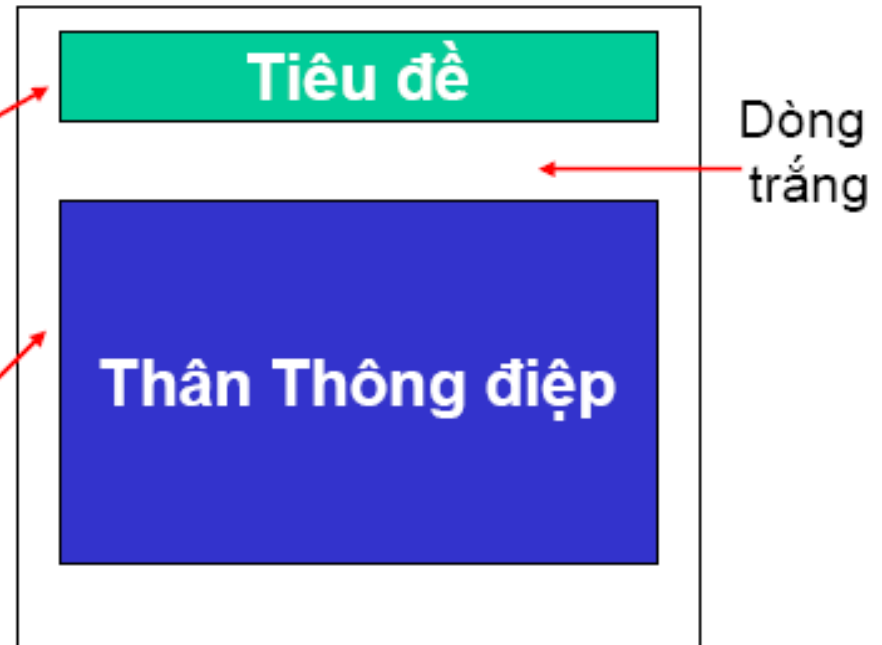
□ Tiêu đề Thông điệp :

- To:
- From:
- Subject:

rất khác so với lệnh của SMTP !

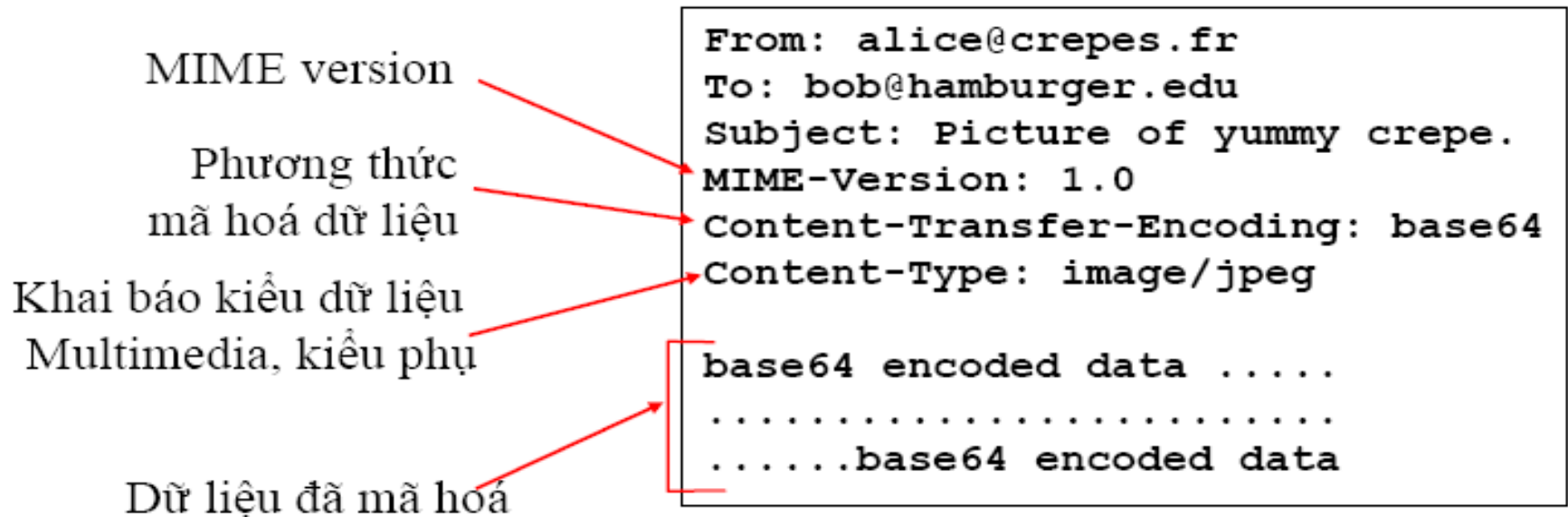
□ Thân thông điệp

- Chỉ bao gồm các ký tự kiểu ASCII



Định dạng thông điệp mở rộng đa phương tiện - MIME

- ❖ MIME: Multipurpose Internet Mail Extension
- ❖ Bổ sung thêm một dòng trong phần tiêu đề của thông điệp để mô tả kiểu MIME



Kiểu MIME

Text

- Kiểu : **plain, html**

Image

- Kiểu : **jpeg, gif**

Audio

- Kiểu : **basic** (loại có quy luật hoá 8-bit),
32kadpcm (mã hóa 32 kbps)

Video

- Kiểu : **mpeg, quicktime**

Application

- Các loại dữ liệu khác phải được xử lý bằng chương trình đọc tương ứng mới có thể đọc, xem được.
- Ví dụ các kiểu: **msword, octet-stream**

Kiểu chứa nhiều loại đối tượng

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=98766789
```

```
--98766789
```

```
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain
```

```
Dear Bob,
Please find a picture of a crepe.
```

```
--98766789
```

```
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
```

```
base64 encoded data .....
```

```
.....
```

```
.....base64 encoded data
```

```
--98766789--
```

Một số giao thức

- ❖ **SMTP- giao thức gửi thư:** Gửi thư từ server chứa thư của người nhận
- ❖ **Giao thức đọc thư:** đặc tả cách lấy và hiển thị thư từ server
 - **POP** : Post Office Protocol [RFC 1939]
 - ✓ *kiểm chứng và tải thư từ server về, lưu trữ tại client*
 - **IMAP**: Internet Mail Access Protocol [RFC 1730]
 - ✓ *phức tạp hơn*
 - ✓ *Thao tác các thư lưu trên server*
 - **HTTP** : Hotmail , Yahoo! Mail, Gmail,...

Giao thức POP3

Giai đoạn kiểm chứng

- Các lệnh Client gửi:
 - **user**: username
 - **pass**: password
- Server trả lời
 - **+OK**
 - **-ERR**

Giai đoạn xử lý, cập nhật :

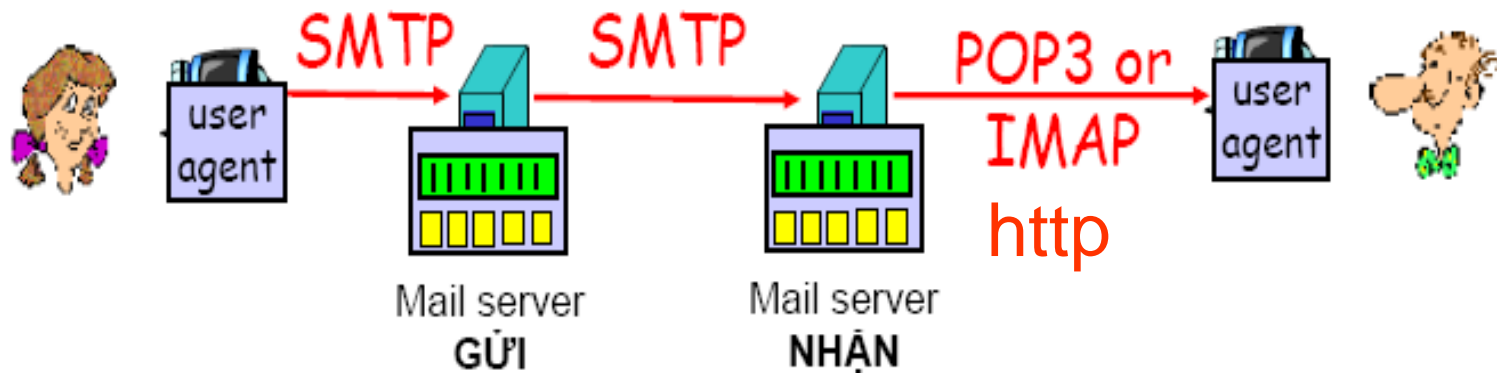
- **list**: in ra danh sách các thư(được đánh số ID).
- **retr**: lấy thư có ID là số nhập vào
- **dele**: xoá thư
- **quit**

```
S: +OK POP3 server ready
C: user alice
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

Một số giao thức

❖ Mô hình hoạt động:

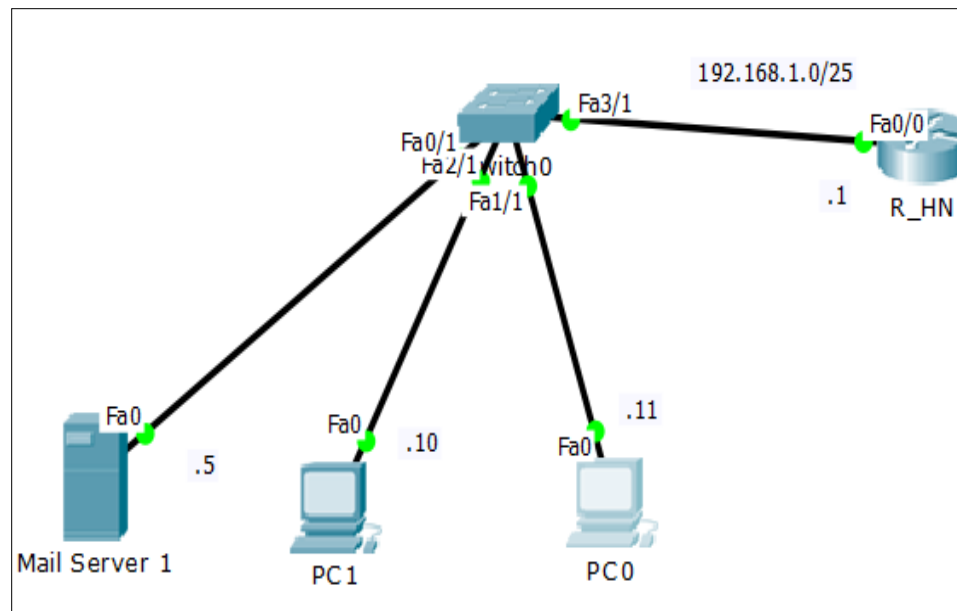


Tổng kết

❖ **Q & A**

Câu hỏi và bài tập

1. So sánh các giao thức hỗ trợ dịch vụ Email: SMTP, POP3, IMAP, MIME (hoạt động, ưu và nhược điểm)
2. Các ứng dụng tầng Application tại sao cần phải sử dụng dịch vụ DNS? Trình bày khái niệm và hoạt động của DNS? (**Chương 19 – Book: TCP/IP Protocols Suite [1]**)
3. **Viết các câu lệnh** cấu hình DHCPv4 trên Router Cisco sử dụng địa chỉ mạng 192.168.1.0/25. Trừ 9 địa chỉ đầu tiên không cấp phát động.



Câu hỏi và bài tập

Bài 4: Cấu hình DHCP Agent trên router Cisco:

- Tham khảo: <https://itforvn.com/tu-hoc-ccna-lab-6-cau-hinh-dhcp/>

Bài 5: Cấu hình IPv6 và DHCPv6 trên router Cisco

- Tham khảo: <https://securityzone.vn/t/lab-15-cau-hinh-ipv6-tren-router-cisco.132/>

Câu hỏi và bài tập

6. Các dịch vụ hoạt động theo mô hình client-server: Liệt kê, hoạt động, giao thức hỗ trợ
7. Tìm hiểu định dạng gói tin và hoạt động của các giao thức: HTTP, FTP, SMTP (POP3, IMAP, MIME)



Thank You for listening!

Khoa CNTT – HV KTMM