

CHƯƠNG IV. INTERNET

NỘI DUNG CỦA CHƯƠNG

- I. Giới thiệu Internet
 - I.1. Giới thiệu Internet
 - I.2. Họ giao thức TCP/IP
- II. Một số dịch vụ trên Internet
 - II.1. Mô hình Client/ Server.
 - II.2. Dịch vụ giao thức : HTTP.
 - II.3. Dịch vụ truyền File : Ftp
 - II.4. Dịch vụ: Electronic Mail
 - II.5. Dịch vụ đặt tên miền DNS
- III. Các dạng hoạt động của Socket
 - III.1. TCP Sockets.
 - III.2. UDP Sockets.
- IV. Thảo Luận



I.1. GIỚI THIỆU INTERNET

Tháng 6/1968 cục các dự án tiên tiến của bộ Quốc phòng Mỹ đã xây dựng án liên kết 4 trung nghiên cứu lớn trong toàn liên bang là : Viện nghiên cứu Stanford, Đại học California Los Angeles, Đại học California Santa Barbara và Đại học Utah thành một hệ thống thống nhất để trao đổi các thông tin. Đến giữa năm 1969, 4 trạm đầu tiên đã kết nối thành công, đánh dấu sự ra đời của mạng **ARPANET** – tiền thân của **INTERNET**. Giao thức truyền thông dùng trong **ARPANET** được gọi là NCP (Network Control Protocol). Tuy nhiên xuất phát từ nhu cầu thực tế, các nhà thiết kế ARPANET đã nhận thức được xây dựng một mạng của các mạng máy tính “**Mạng của các mạng-INTERNET**”, vì vậy giữa những năm 70, họ giao thức TCP/IP được Vint Cerf và Robert Kahn đề xuất và phát triển, ban đầu cùng tồn tại với NCP trong mạng ARPANET và đến năm 1983 thì hoàn toàn thay thế NCP.

Thuật ngữ **INTERNET** được xuất hiện lần đầu tiên vào năm 1974. Nhưng tên gọi ARPANET vẫn tồn tại cho đến những năm

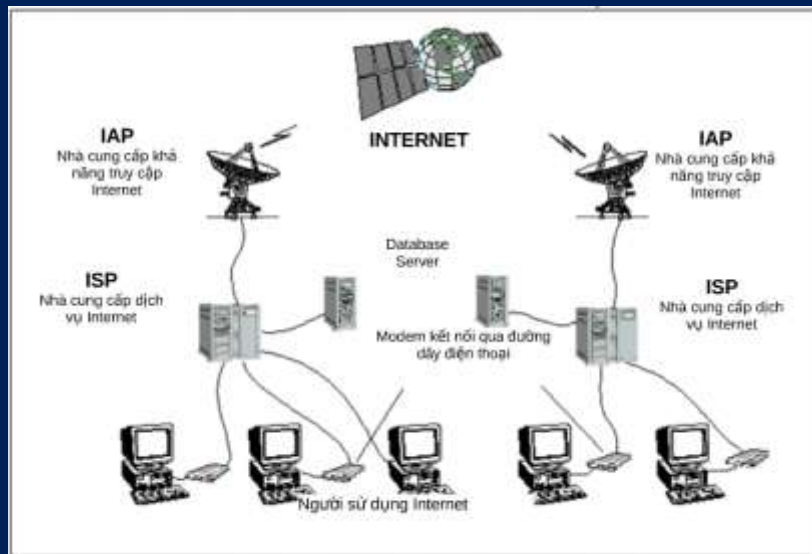
So sánh thời gian đạt được 50 triệu người dùng trên thế giới:

- Telephone sau 74 năm
- Radio sau 38 năm
- PC sau 16 năm
- TV sau 13 năm
- WWW sau 4 năm



• Cấu trúc mạng và kết nối Internet

- Internet là một mạng GAN dựa trên kết nối liên mạng WAN, sử dụng mô hình TCP/IP. Việc kết nối và truy cập Internet của người dùng được cung cấp bởi các nhà cung cấp dịch vụ Internet ISP (Internet Service Provider). Các ISP phải thuê đường và cổng của một IAP.



5

1.2. HỌ GIAO THỨC TCP/IP

TCP/IP là một giao thức cùng làm việc với nhau để cung cấp phương tiện truyền thông liên mạng. So sánh với mô hình tham chiếu OSI ta thấy:

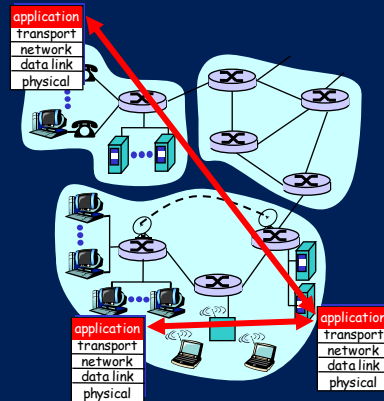
So sánh các mô hình giao thức trên INTERNET

TCP/IP Model	TCP/IP Protocols			OSI Ref Model
Application	FTP	Telnet	HTTP	Application
Transport	TCP	UDP		Session
Internetwork	IP			Transport
Host to Network	Ethernet	Packe Radio	Point-to-Point	Network
				Datalink
				Physical

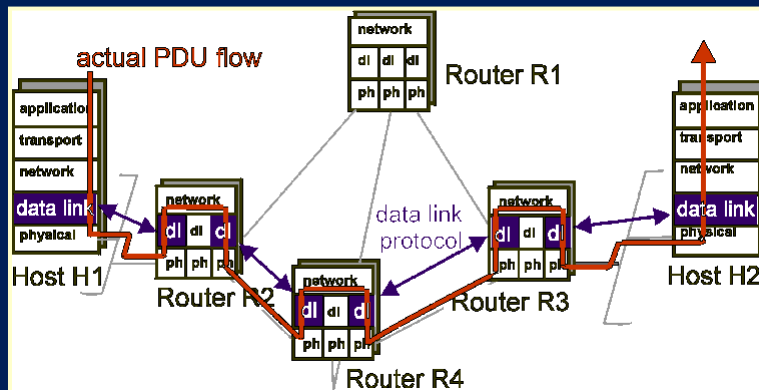


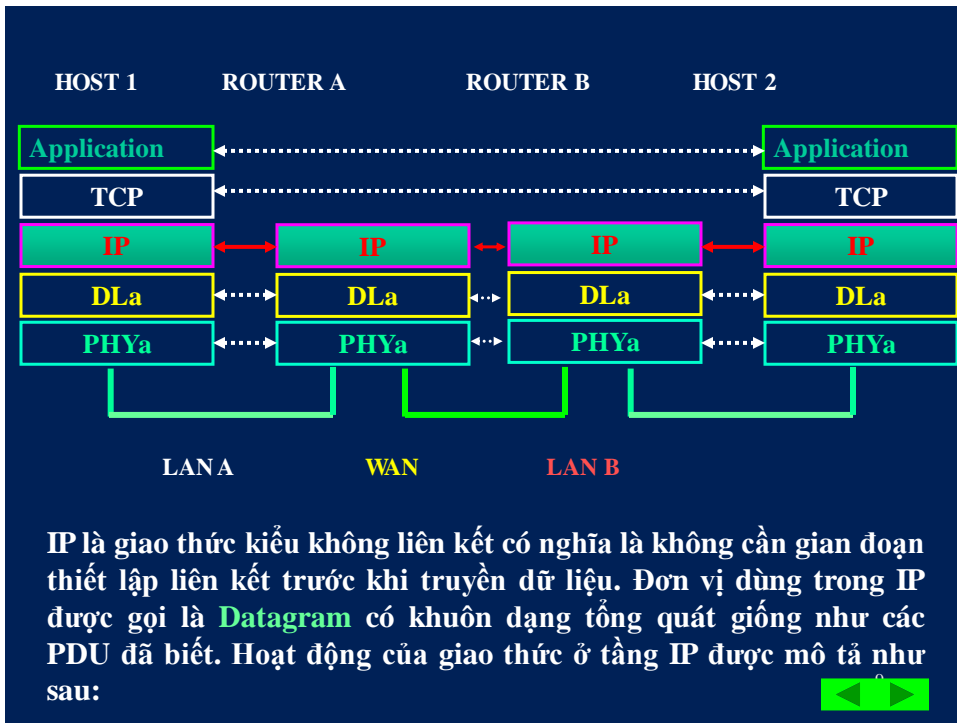
1. Hoạt động của giao thức IP-Internet Protocol

- Application: Truyền thông và các quá trình phân tán
 - Thực hiện giao dịch giữa các Hosts theo địa chỉ người dùng
 - Thay đổi các thông báo theo yêu cầu ứng dụng
 - Thực hiện các dịch vụ E-mail, truyền file (FTP), WWW,...



Phân tích quá trình hoạt động của mô hình giao thức IP trên INTERNET ta có thấy việc giao dịch được tiến hành:





Đối với thực thể IP ở trạm nguồn:

1. Tạo một IP Datagram để gửi đi
 2. Tính Checksum và ghép vào Header của Datagram
 3. Ra quyết định chọn đường: hoặc trạm đích thuộc cùng một mạng hoặc khác mạng chuyển tới Router hoặc Gateway tiếp theo.
 4. Chuyển Datagram xuống tầng dưới để truyền qua mạng.
- Đối với Gateway hoặc Router:
1. Tính lại Checksum nếu có lỗi thì loại bỏ.
 2. Giảm giá trị tham số Time to Live. Nếu bằng 0 thì loại bỏ.
 3. Ra quyết định chọn đường.
 4. Phân đoạn Datagram nếu cần.
 5. Kiến tạo lại IP Header, bao gồm giá trị mới của tham số Time to Live, phân đoạn và Checksum.
 6. Chuyển Datagram xuống tầng dưới để truyền qua mạng.

Đối với thực thể IP ở trạm

dịch:

1. Tính Checksum nếu có lỗi thì loại bỏ
2. Tập hợp các Datagram nếu dữ liệu bị phân đoạn.
3. Chuyển Datagram và các tham số điều khiển lên tầng

2. Giao thức TCP – Transfer Control Protocol

TCP là giao thức có liên kết, tức là cần phải thiết lập liên kết (logic) giữa hai thực thể trước khi tiến hành trao đổi dữ liệu. Đơn vị dữ liệu sử dụng trong TCP gọi là **Segment** (đoạn dữ liệu). Khuôn dạng tổng quát như sau:



Source Port								Destination Port							
Sequence Number															
Acknowledgment Number															
Data Offset	Reser-ved	U R G	A C K	P S H	R S T	S Y N	F I N	Window							
Checksum								Urgent Pointer							
Option								Padding							
TCP data															

Source Destination Port Số hiệu cổng.

Option Khai báo các thay đổi.

URG Con trỏ khẩn có hiệu lực. **ACK** Tín hiệu báo nhận. **PSH**

Chức năng đẩy. **RST** Reset. **SYN** Đồng bộ hoá. **FIN** Không còn dữ liệu từ trạm nguồn.



Một cổng kết hợp với một địa chỉ IP tạo thành một **Socket** duy nhất trong liên mạng. Dịch vụ TCP được cung cấp nhờ một liên kết logic giữa một cặp **Socket**. Một Socket có thể tham gia với nhiều **Socket** ở xa. Trước khi truyền dữ liệu giữa 2 trạm cần phải thiết lập liên kết TCP giữa chúng và khi không còn nhu cầu truyền dữ liệu thì liên kết đó sẽ được giải phóng.

3. Giao thức UDP – User Datagram Protocol

UDP là giao thức không liên kết được sử dụng thay thế cho TCP theo yêu cầu của ứng dụng. Nó thường được dùng cho các ứng dụng không đòi hỏi độ tin cậy cao trong giao vận. Đơn vị dữ liệu được gọi là **UDP datagram**. Có khuôn dạng như sau:

Source Port	Destination Port
Message Length	Checksum
Data	



II. MỘT SỐ DỊCH VỤ TRÊN INTERNET

TCP/IP là một giao thức cùng làm việc với nhau để cung cấp phương tiện truyền thông liên mạng. So sánh với mô hình tham chiếu OSI ta thấy:

1. Dịch vụ WWW

2. Dịch vụ Telnet

3. Dịch vụ Email

4. Dịch vụ FTP

5. Dịch vụ tên miền DNS



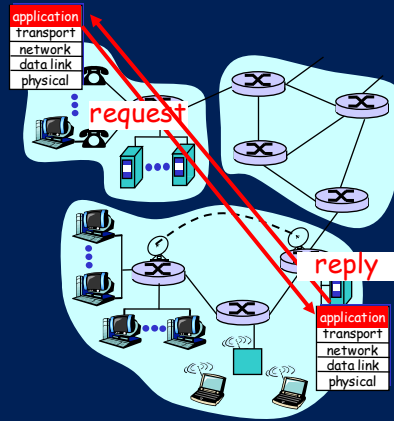
II.1. Mô hình Client-server

Một trao đổi điển hình giữa:

client và *server*

Client:

- Khởi tạo kết nối với server (“speaks first”)
- Yêu cầu **requests** dịch vụ từ server,
- Các dịch vụ nh- Web, client là công cụ duyệt browser; nh- e-mail là đọc th-

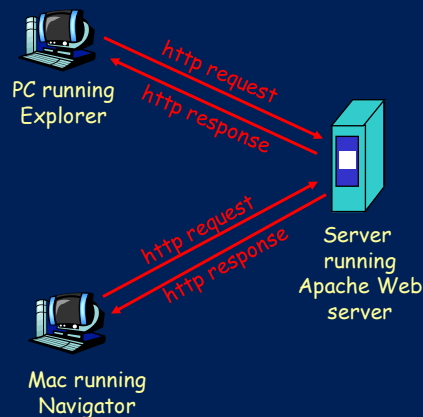


15

II.2. Dịch vụ: giao thức http

http: hypertext transfer protocol

- ✓ Dịch vụ Web thuộc giao thức tầng ứng dụng
- ✓ Mô hình client/server
 - **client:** duyệt và tiếp nhận các yêu cầu, “displays” Web objects
 - **server:** Web server chuyển các đối tượng để đáp ứng các yêu cầu
- ✓ http1.0: RFC 1945
- ✓ http1.1: RFC 2068

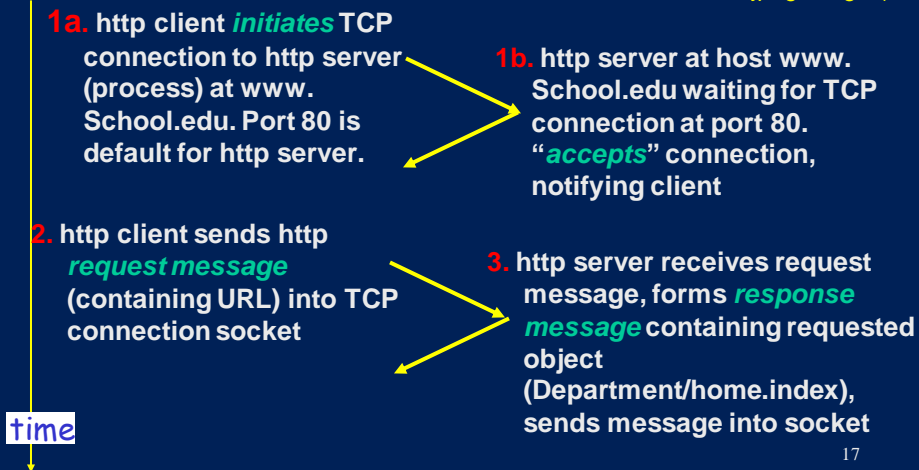


16

Ví dụ về http

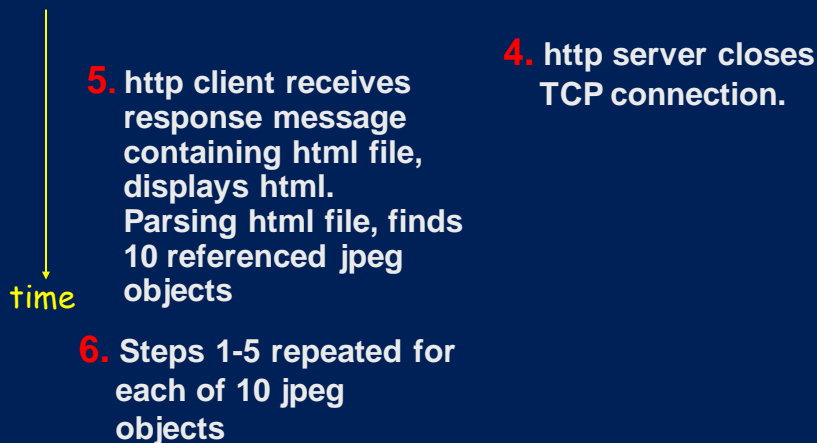
Giả sử người sử dụng đưa vào địa chỉ URL sau **www.School.edu/Department/home.index**

(contains text,
references to 10
jpeg images)



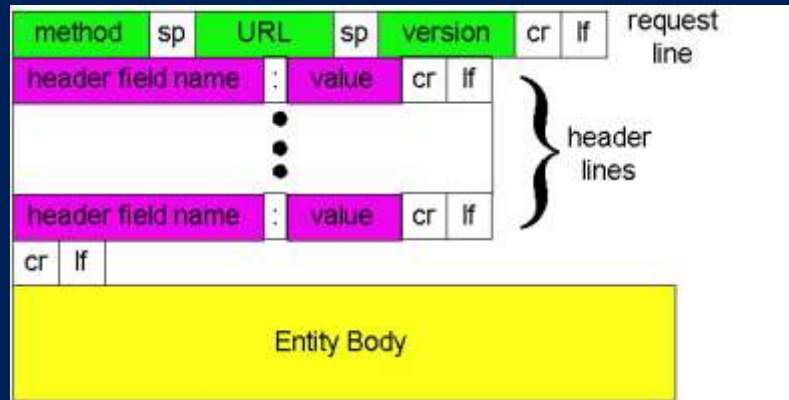
17

http example (cont.)



18

Dạng tổng quát của http *request* message



19

Khuôn dạng của http message: response

status line
(protocol
status code
status phrase)

header
lines

data, e.g.,
requested
html file

```

HTTP/1.0 200 OK
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 .....
Content-Length: 6821
Content-Type: text/html

data data data data data ...
    
```

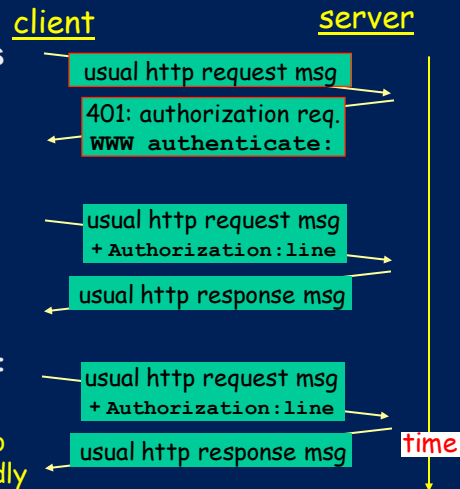
20

User-server interaction: authentication

Authentication goal: control access to server

- **stateless:** client presents authorization in each request
- **authorization:** typically name, password
 - authorization: header line in request
 - Sends header line
WWW authenticate:
if unauthorized

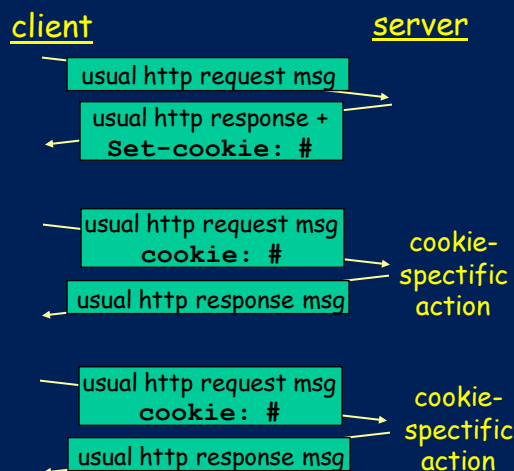
Browser caches name & password so that user does not have to repeatedly enter it.



21

User-server interaction: cookies

- server sends "cookie" to client in response msg
Set-cookie: 1678453
- client presents cookie in later requests
cookie: 1678453
- server matches presented-cookie with server-stored info
 - authentication
 - remembering user preferences



22

User-server interaction: *conditional* GET

- **Goal:** don't send object if client has up-to-date stored (cached) version

- **client:** specify date of cached copy in http request

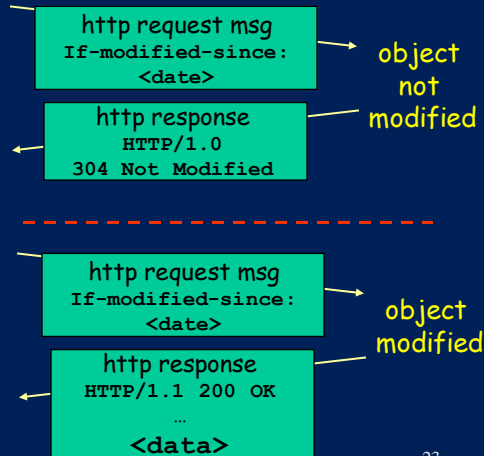
If-modified-since:
<date>

- **server:** response contains no object if cached copy up-to-date:

HTTP/1.0 304 Not Modified

client

server

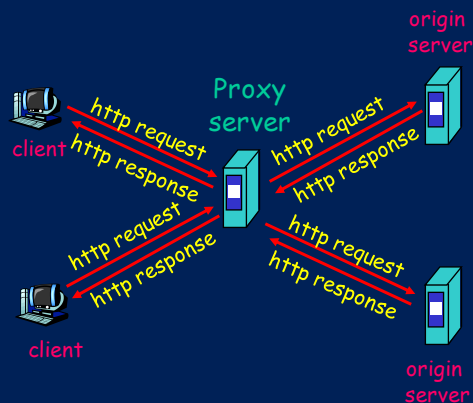


23

Web Caches (proxy server)

Goal: satisfy client request without involving origin server

- user sets browser: Web accesses via web cache
- client sends all http requests to web cache
 - if object at web cache, web cache immediately returns object in http response
 - else requests object from origin server, then returns http response to client

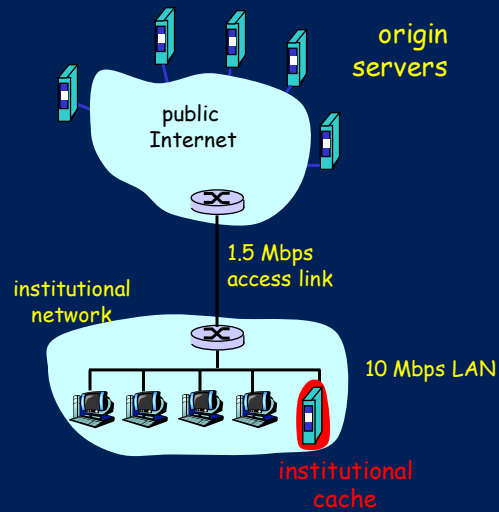


24

Why Web Caching?

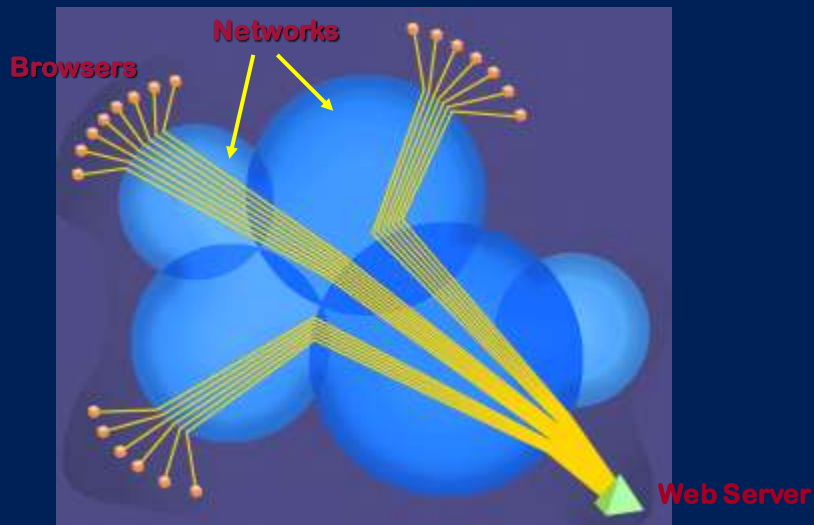
Assume: cache is “close” to client (e.g., in same network)

- smaller response time: cache “closer” to client
- decrease traffic to distant servers
 - link out of institutional/local ISP network often bottleneck



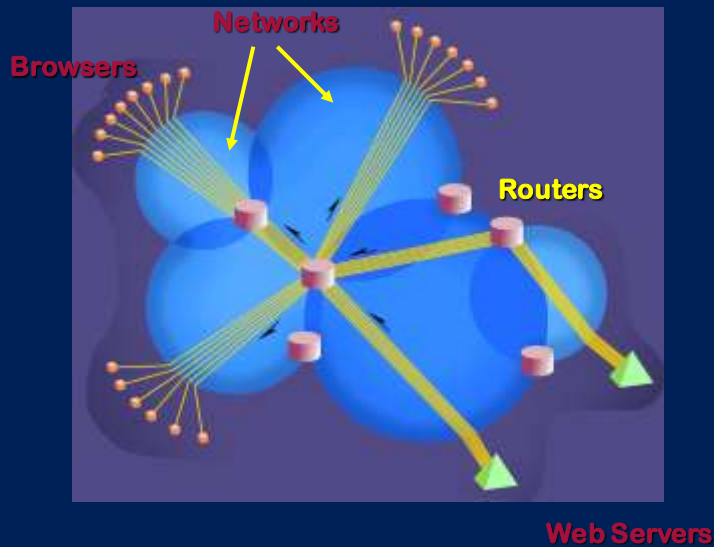
25

Mô hình phân phối nội dung



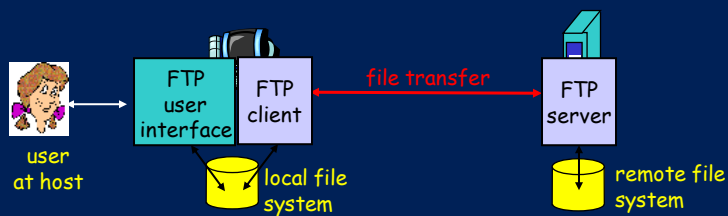
26

Xảy ra tắc nghẽn trên đường truyền



27

II.3. Dịch vụ truyền File ftp

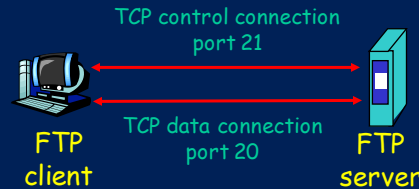


- transfer file to/from remote host
- client/server model
 - **client**: side that initiates transfer (either to/from remote)
 - **server**: remote host
- ftp: RFC 959
- ftp server: port 21

28

ftp: separate control, data connections

- ftp client contacts ftp server at port 21, specifying TCP as transport protocol
- two parallel TCP connections opened:
 - **control**: exchange commands, responses between client, server.
“out of band control”
 - **data**: file data to/from server
- ftp server maintains “**state**”: current directory, earlier authentication



29

ftp commands, responses

Sample commands:

- sent as ASCII text over control channel
- **USER** *username*
- **PASS** *password*
- **LIST** returns list of file in current directory
- **RETR** *filename* retrieves (gets) file
- **STOR** *filename* stores (puts) file onto remote host

Sample return codes

- status code and phrase (as in http)
- **331** Username OK, password required
- **125** data connection already open; transfer starting
- **425** Can't open data connection
- **452** Error writing file

30

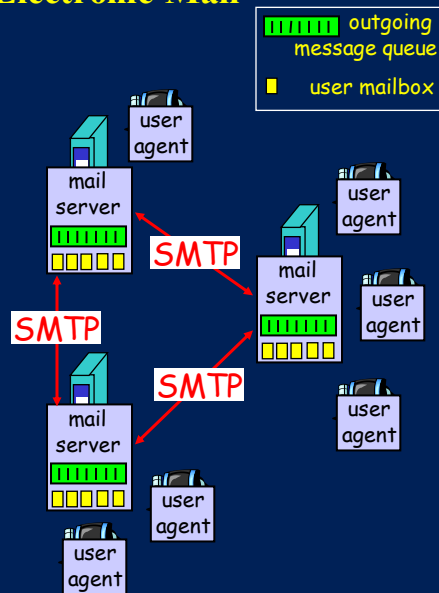
II.4. Dịch vụ: Electronic Mail

Three major components:

- user agents
- mail servers
- simple mail transfer protocol: smtp

User Agent

- a.k.a. “mail reader”
- composing, editing, reading mail messages
- e.g., Eudora, Outlook, elm, Netscape Messenger
- outgoing, incoming messages stored on server

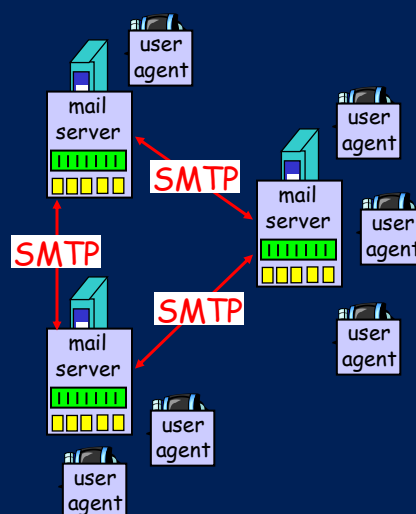


31

Electronic Mail: mail servers

Mail Servers

- **mailbox** contains incoming messages (yet to be read) for user
- **message** queue of outgoing (to be sent) mail messages
- **smtp protocol** between mail servers to send email messages
 - client: sending mail server
 - “server”: receiving mail server



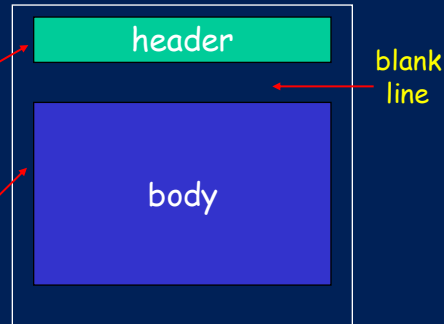
32

Mail message format

smtp: protocol for
exchanging email
msgs

RFC 822: standard for
text message format:

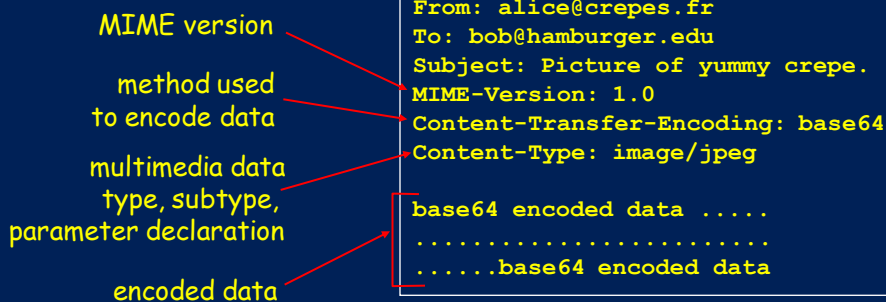
- header lines, e.g.,
 - To:
 - From:
 - Subject:*different from smtp commands!*
- body
 - the “message”, ASCII characters only



33

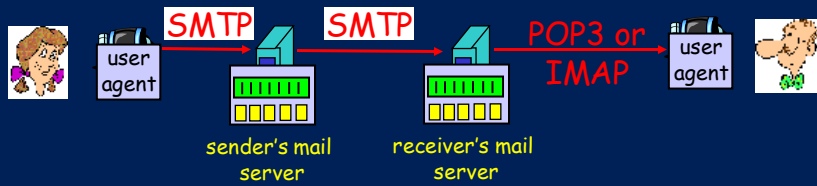
Message format: multimedia extensions

- MIME: multimedia mail extension, RFC 2045, 2056
- additional lines in msg header declare MIME content type



34

Mail access protocols



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
 - POP: Post Office Protocol [RFC 1939]
 - authorization (agent <--> server) and download
 - IMAP: Internet Mail Access Protocol [RFC 1730]
 - more features (more complex)
 - manipulation of stored msgs on server
 - HTTP: Hotmail , Yahoo! Mail, etc.

35

POP3 protocol

authorization phase

- **client commands:**
 - user: declare username
 - pass: password

server responses

- +OK
- -ERR

transaction phase, client:

- list: list message numbers
- retr: retrieve message by number
- dele: delete
- quit

```

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

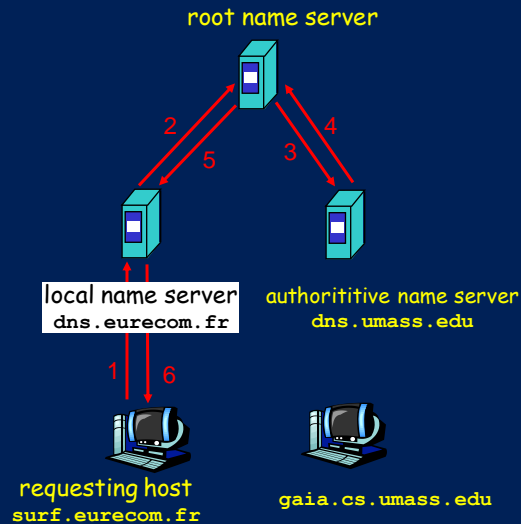
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
  
```

36

II.5. Dịch vụ đặt tên miền DNS

host `surf.eurecom.fr`
wants IP address of
`gaia.cs.umass.edu`

1. Contacts its local DNS server,
`dns.eurecom.fr`
2. `dns.eurecom.fr` contacts root name server, if necessary
3. root name server contacts authoritative name server,
`dns.umass.edu`, if necessary



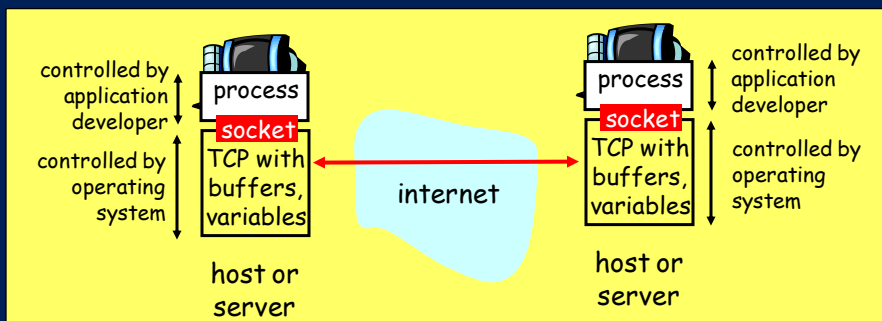
37

III. Các dạng hoạt động của Socket.

III.1. TCP sockets

Socket: Là cửa kết nối giữa một quá trình ứng dụng và đầu cuối giao thức tầng transport (UCP or TCP)

TCP service: dịch vụ vận chuyển tin cậy các byte từ một quá trình này sang một quá trình khác



38

TCP sockets

Client must contact server

- server process must first be running
- server must have created socket (door) that welcomes client's contact

Client contacts server by:

- creating client-local TCP socket
- specifying IP address, port number of server process

- When **client creates socket**: client TCP establishes connection to server TCP
- When contacted by client, **server TCP creates new socket** for server process to communicate with client
 - allows server to talk with multiple clients

application viewpoint

TCP provides reliable, in-order transfer of bytes ("pipe") between client and server

39

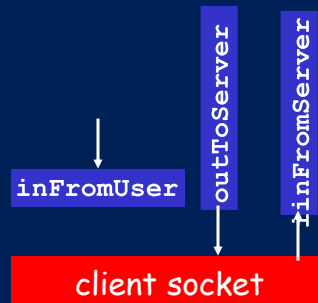
TCP Sockets

Example client-server app:

- client reads line from standard input (`inFromUser` stream), sends to server via socket (`outToServer` stream)
- server reads line from socket
- server converts line to uppercase, sends back to client
- client reads, prints modified line from socket (`inFromServer` stream)

Input stream: sequence of bytes into process

Output stream: sequence of bytes out of process

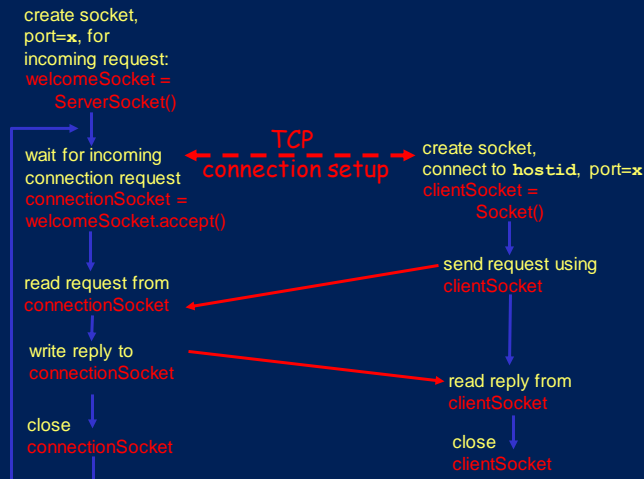


40

Trương tác Client/server socket: TCP

Server (running on `hostid`)

Client



41

III.2. UDP Sockets

UDP: no "connection" between client and server

- no handshaking
- sender explicitly attaches IP address and port of destination
- server must extract IP address, port of sender from received datagram

application viewpoint

UDP provides unreliable transfer of groups of bytes ("datagrams") between client and server

UDP: transmitted data may be received out of order, or lost

42

Client/server socket interaction: UDP

Server (running on `hostid`)

create socket,
port=`x`, for
incoming request:
`serverSocket =`
`DatagramSocket()`

read request from
`serverSocket`

write reply to
`serverSocket`
specifying client
host address,
port number

Client

create socket,
`clientSocket =`
`DatagramSocket()`

Create, address (`hostid`, `port=x`,
send datagram request
using `clientSocket`

read reply from
`clientSocket`

close
`clientSocket`

43