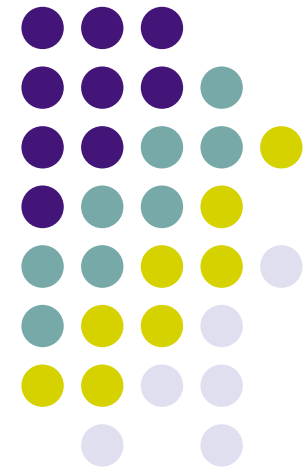
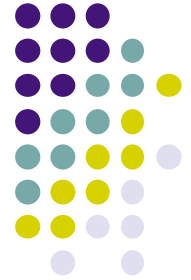


Lecture 9: Application layer

Reading Chapter 7
Computer networks, Tanenbaum

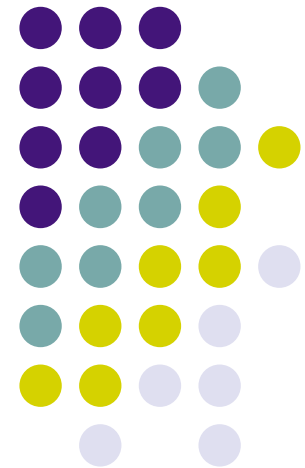


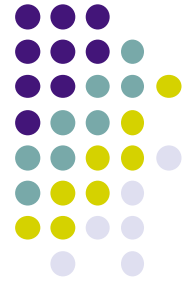


Contents

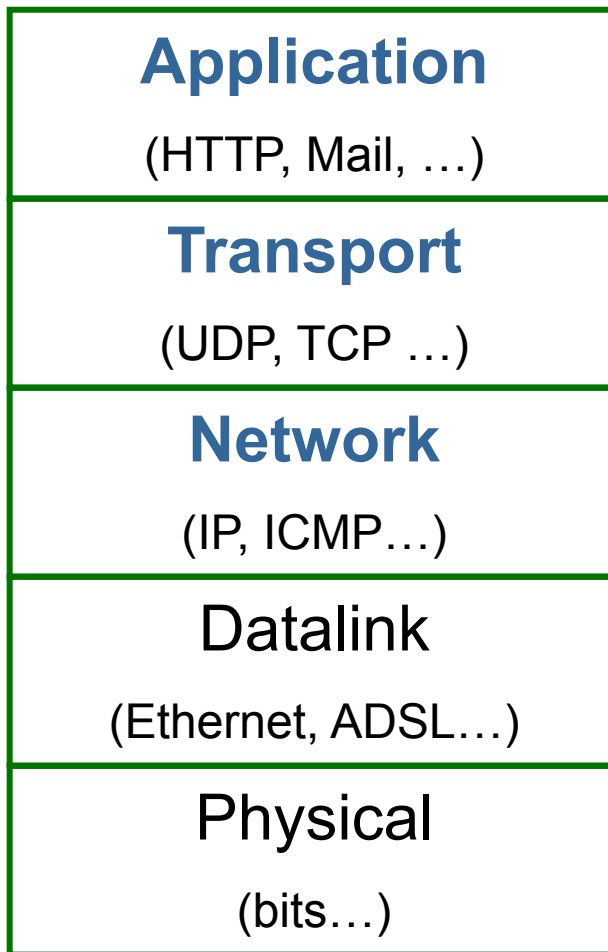
- Application layer
 - Fundamental concepts
 - Case study: HTTP, Mail, FTP...

Fundamental concepts



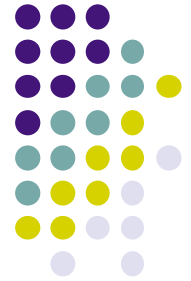


Application layer in OSI model



**Protocols communication
between parties of the
application**

Transmission data between application



Application and service?

VoIP

MUSIC ONLINE

GAME ON LINE

CHAT

VoD

SMS

e-Office

e-BANK

MAIL

E-learning

WEB

YOUTUBE

VIDEO CONFERENCE

FTP

EBAY

GOOGLE

SKYPE

Social networks

SSH

NEWS

BITTORENT

E-COMMERCE

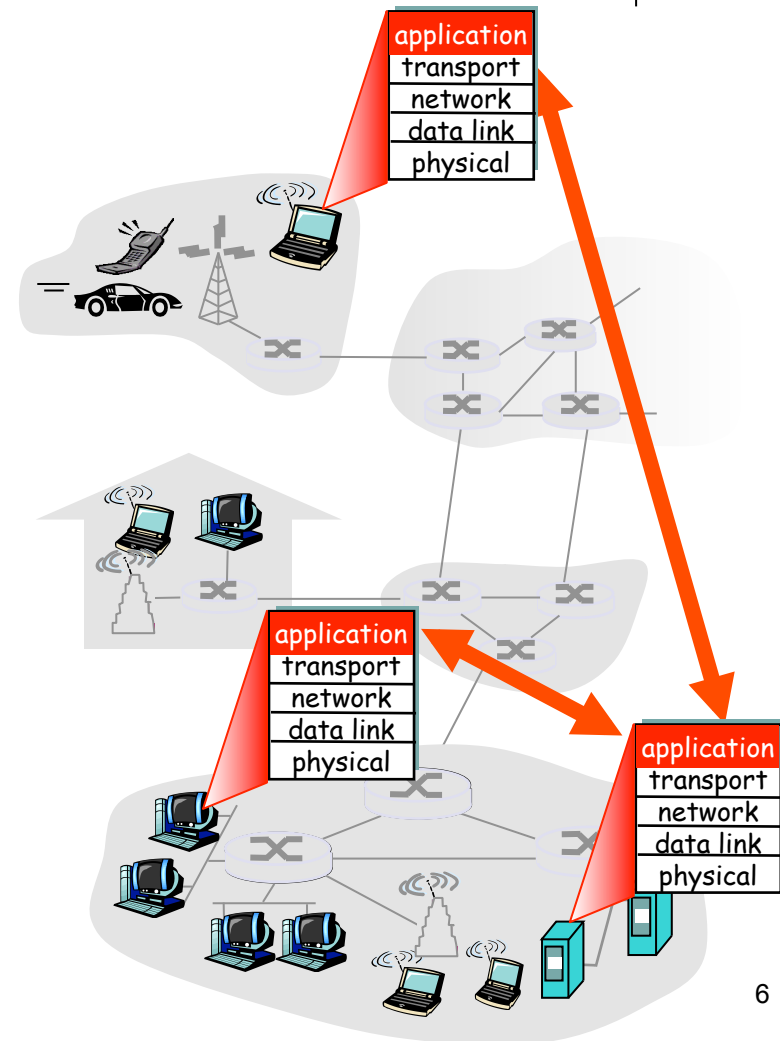
GRID

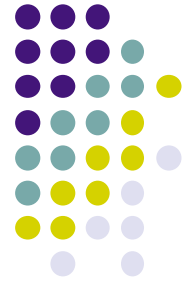
e-Government

Application and application protocol



- Application protocol
 - Define communication rule
 - Use service of transport layer (TCP/UDP...)
- Application:
 - Is a process on the internet. They communicate to each other by exchanging messages.
 - Runs on end systems
 - Use application protocol for providing service
- Example of application/protocol:
 - Web (HTTP)
 - Mail (SMTP/POP/IMAP) ...





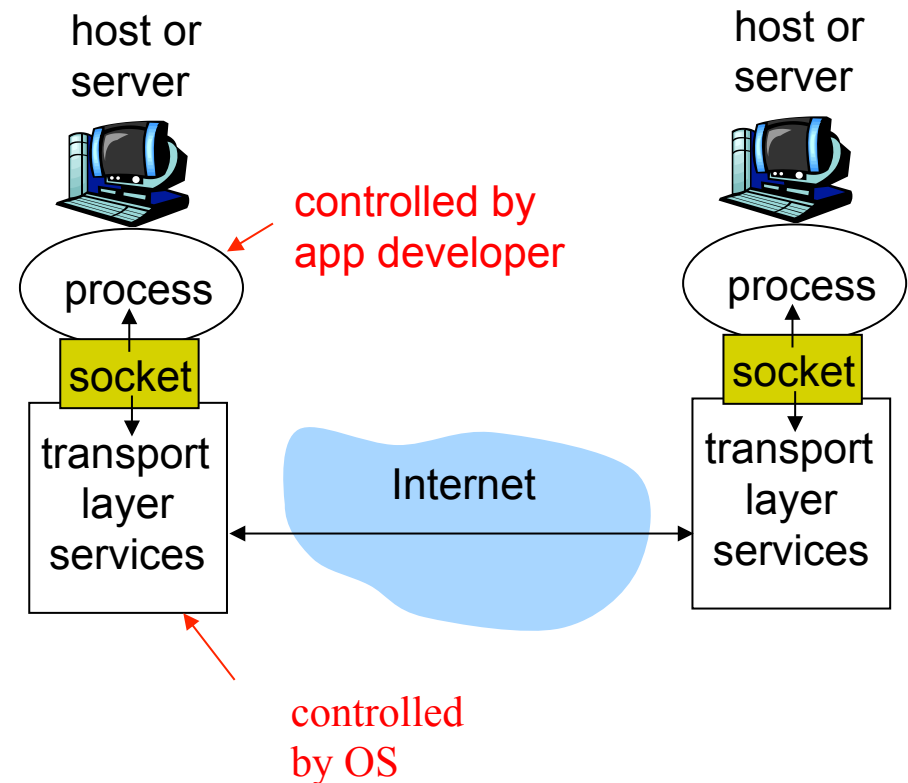
Components of an application

- **Application software is compose of**
 - **User interface:**
 - Interfacing with users,
 - e.g. Web browser (Firefox, IE), mail reader(Thunderbird, Outlook,..)
 - Implement one part of application protocol
 - **Server program:**
 - Cung cấp dịch vụ cho người sử dụng
- **Application process:** the application software running on an OS

Communication between process on the Internet

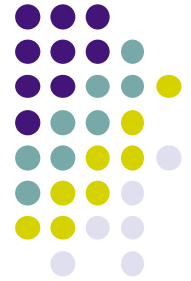


- Socket is an interface between an application process and transport layer
- Socket is defined by
 - Số hiệu cổng
 - Địa chỉ IP
 - Kiểu giao thức giao vận (TCP hay UDP)
- Socket API (Application Programming Interface): Allow application to choose parameters for transport service
 - Choose transport protocol
 - Type of IO communication ...

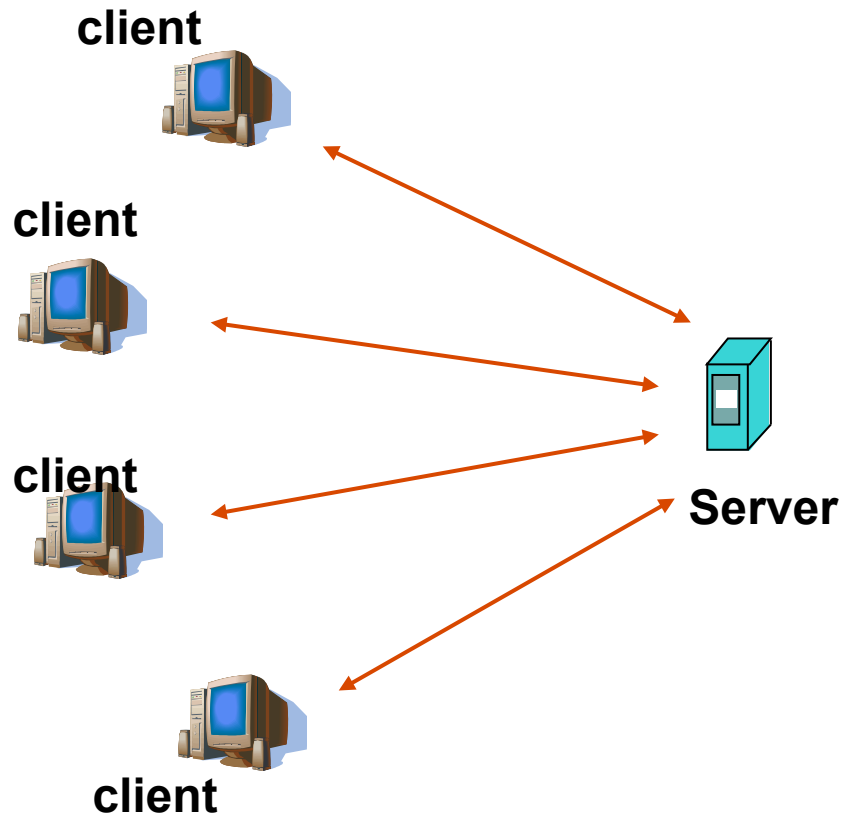


Application architecture

- Client-server
- P2P
- Hybrid



Client-server

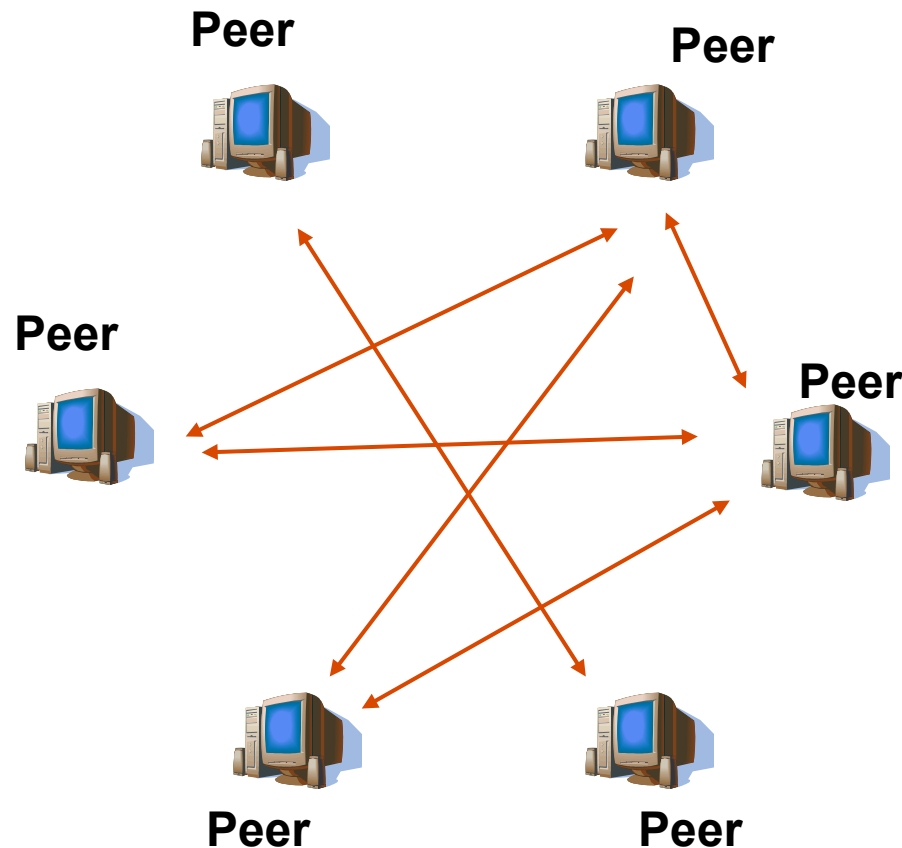


- Two kind of components: client and server
- **Client**
 - Client sends requests for service to server
 - Clients do not contact directly to each other
- **Server**
 - Always online waiting for service requests from clients
 - There may be backup servers for assuring high availability in failures
- e.g. Web, Mail, ...





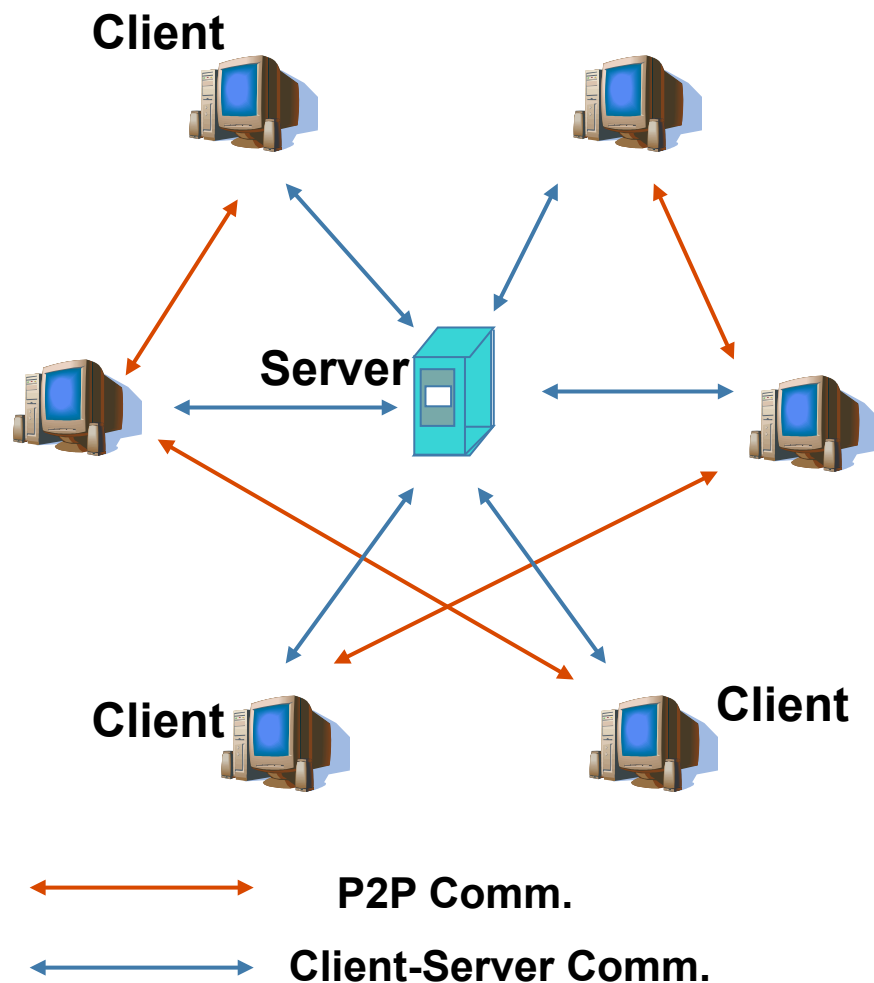
Pure Peer-to-peer architecture



- No center server, only peers as components
- Peers have equal role in the system
- Any two peers can communicate directly to each other but only when both are online.
- Peer does not need to be online all the time
- E.g. Gnutella, Bittorrent



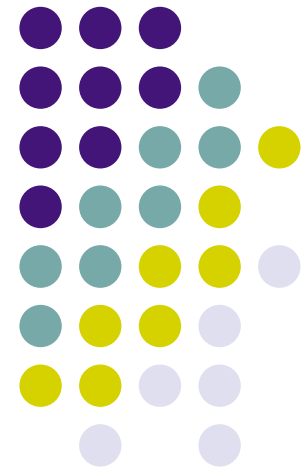
Hybrid architecture



- A center server for user management, indexing for search purpose.
- Clients communicate directly to each other after authentication process with server.
- E.g. Skype (before 2016)
 - Skype server manage user lists, authentication
 - After authentication users communicate directly to each other

Case study 1: HTTP and WWW

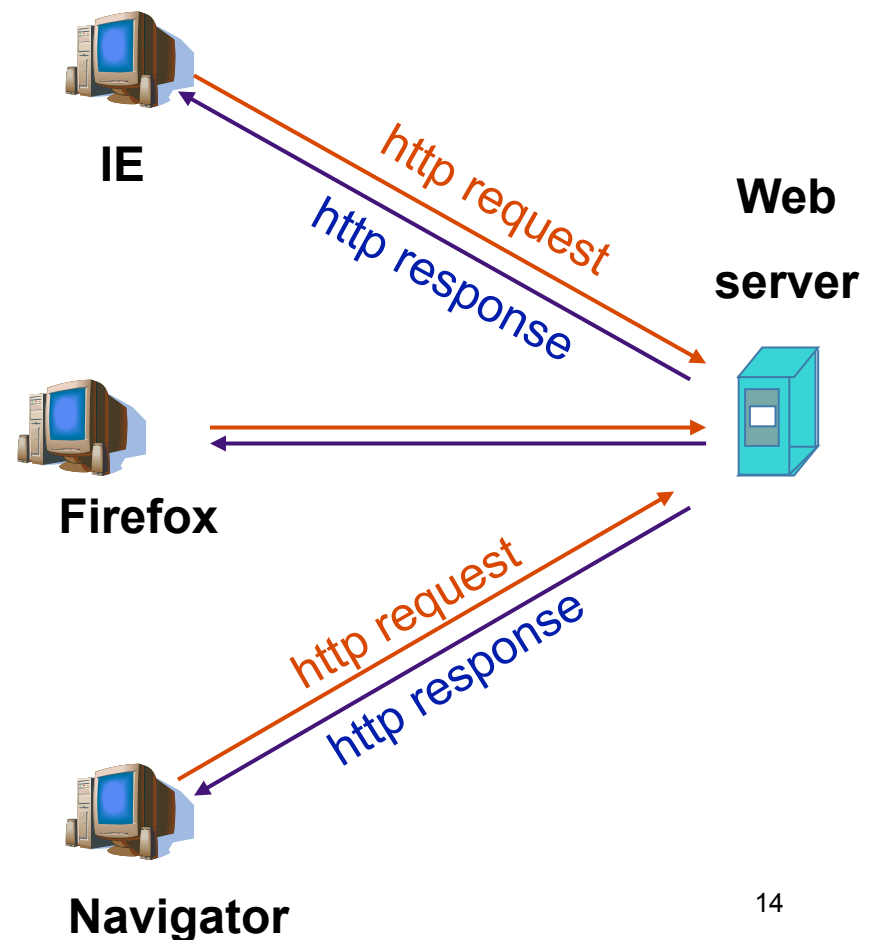
Reading 7.3
Computer Networks, Tanenbaum

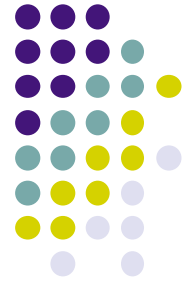


HTTP and Web

- WWW: World Wide Web
 - Application for exchanging the HTML documents (HyperText Markup Language) over Internet
 - WWW use HTTP protocol
- HTTP: HyperText Transfer Protocol
 - Client/Server model
 - Client (Web browser) requests for webpages and displays them on its interface
 - Server: Receive request from client and return results under the form of webpage.

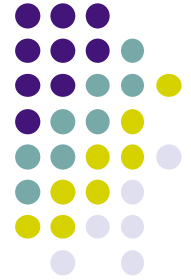
Web clients





How HTTP works?

- Server open a TCP socket by default at port 80 waiting for
- Client initiates a TCP connection to server
- Server accept the connection request
- Exchange HTTP message
 - HTTP Request
 - HTTP Response
- Close connection TCP



Format of HTTP request

- ASCII encoding (readable using text editor)

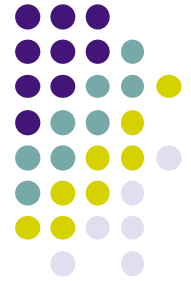
request line
(GET, POST,
HEAD commands)

header
lines

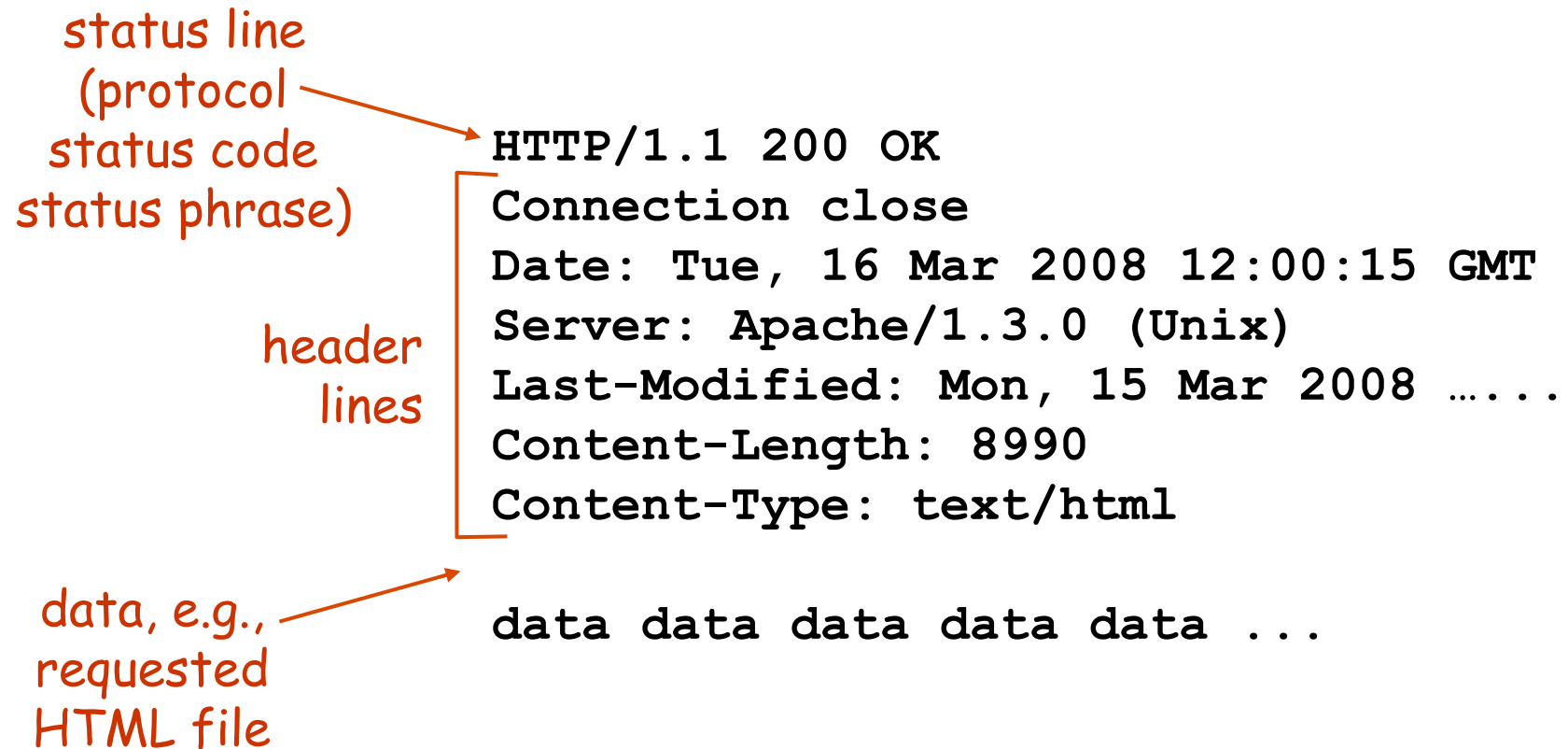
```
GET /dccn/index.html HTTP/1.1
Host: www.it-hut.edu.vn
User-agent: Mozilla/4.0
Connection: close
Accept-language: en-us
```

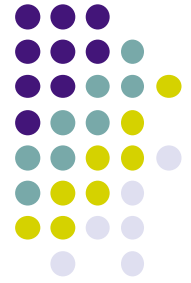
CR, LF
(extra carriage return, line feed)

indicates end
of message



Format of HTTP response





Types of HTTP connections

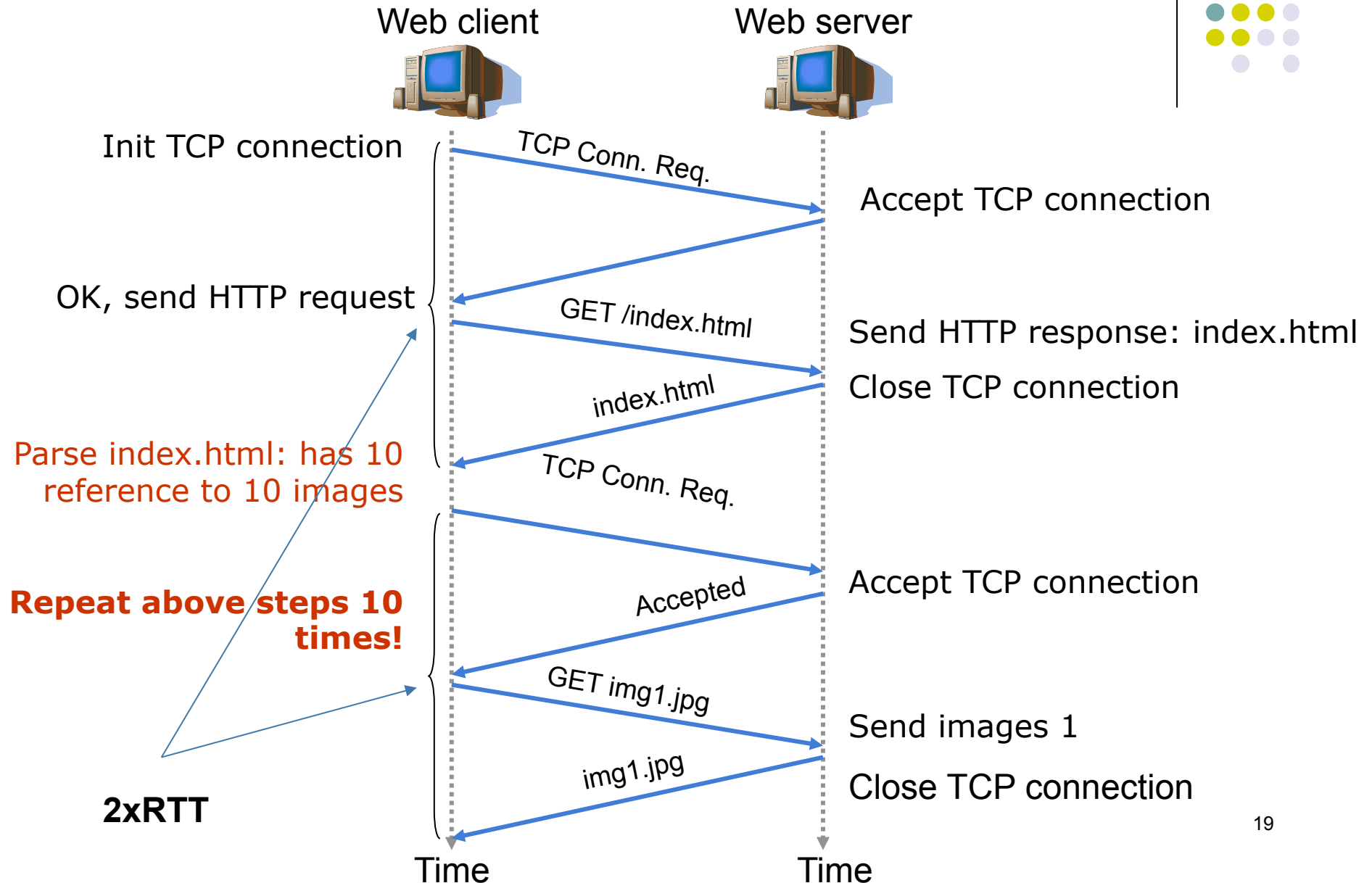
HTTP non-persistent

- Only one web object (text or image) is transferred over a connection TCP
- Option by default in HTTP/1.0
- HTTP 1.0: RFC 1945

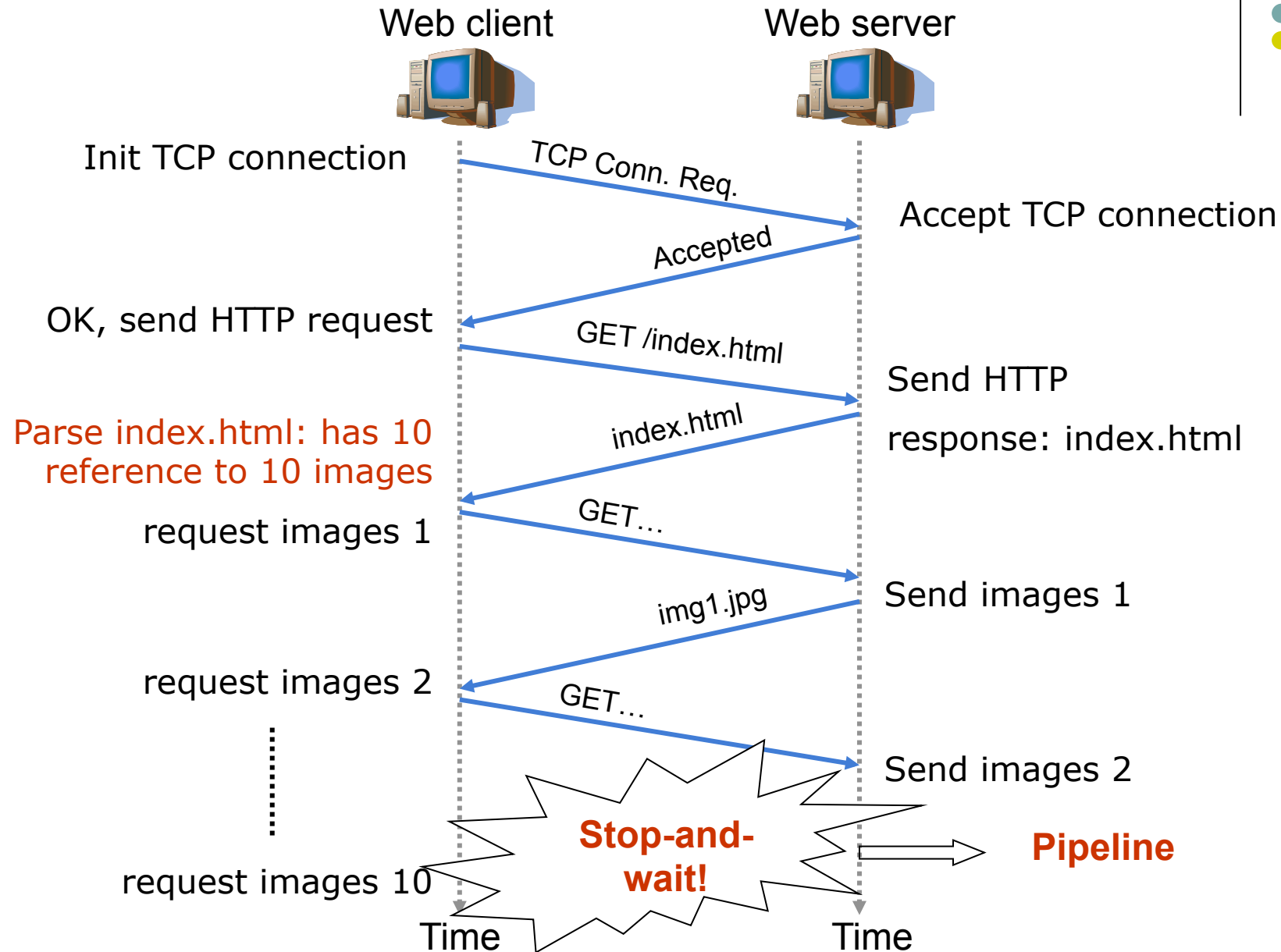
HTTP persistent

- Many web objects can be sent over a connection TCP.
- Option by default in HTTP/1.1
- HTTP 1.1: RFC 2068

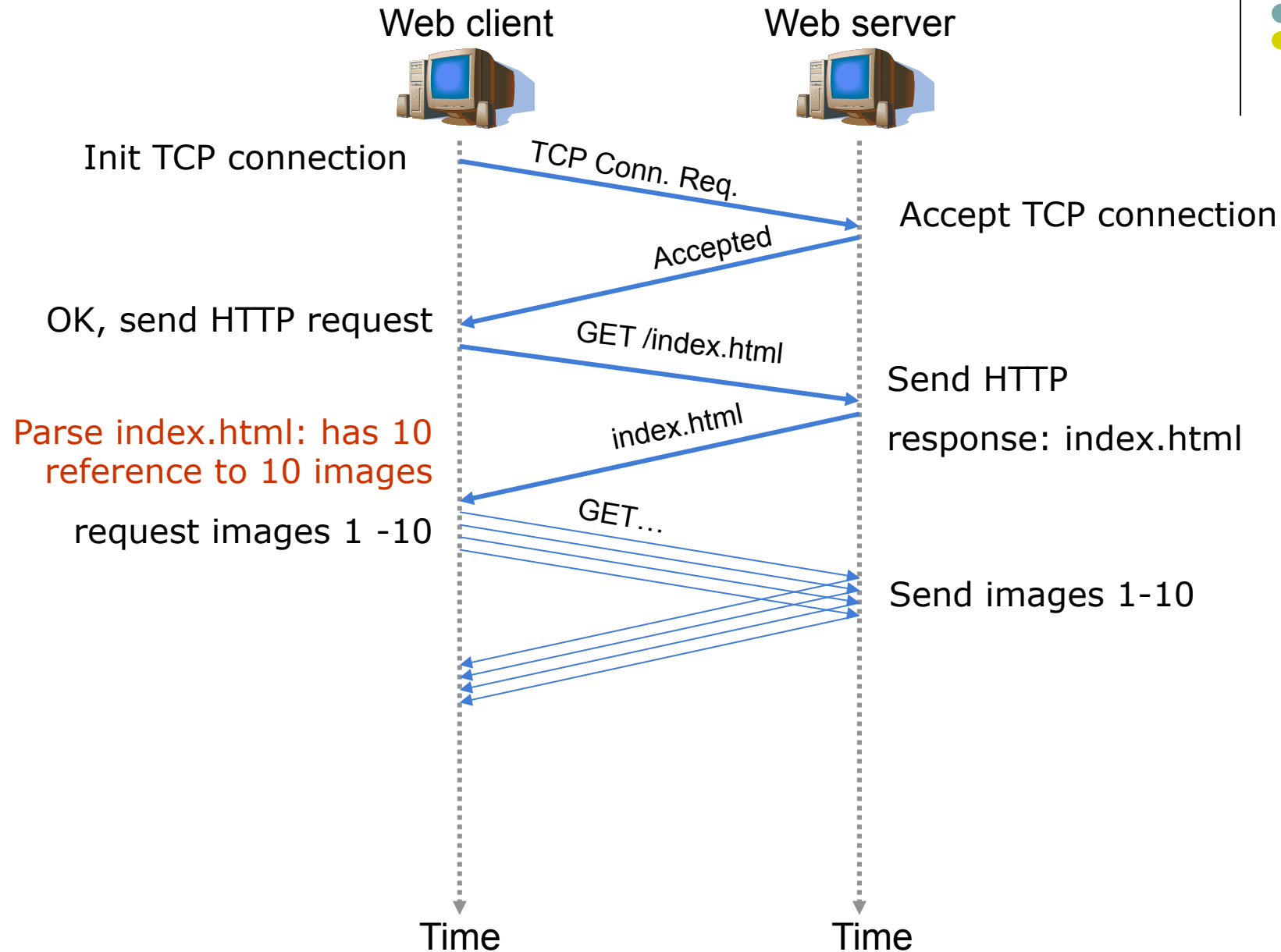
Operation of HTTP/1.0



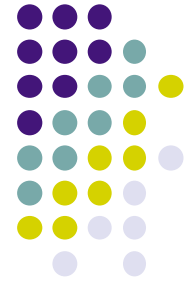
Operation of HTTP/1.1



HTTP/1.1 with pipeline



Methods in HTTP request message



HTTP/1.0

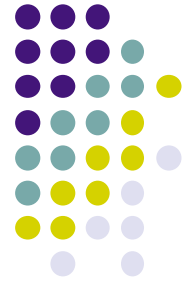
- GET: get an webpage
- POST: submitting a form
- HEAD: ask for the header of an webpage

HTTP/1.1

- GET, POST, HEAD
- PUT
 - Upload an webpage to the server under address given in URI, file content is in the body of the message
- DELETE
 - Delete a file given in the URI

Attention: Even with GET, user can sends parameters to servers in URL. Ex:

<http://www.google.com/search?q=computer+network&flags=68&num=10>



Status code in Response message

Status code is in the first line of the Response message

200 OK

- request succeeded, requested object later in this message

301 Moved Permanently

- requested object moved, new location specified later in this message (Location:)

400 Bad Request

- request message not understood by server

404 Not Found

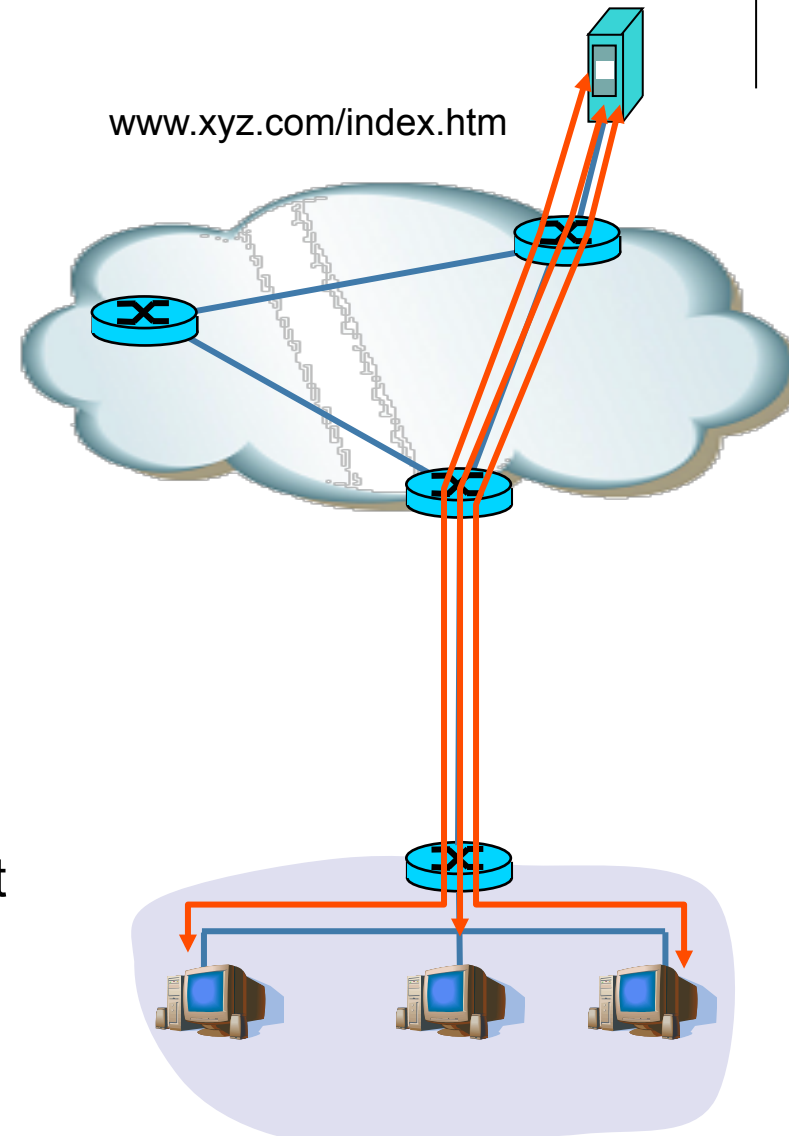
- requested document not found on this server

505 HTTP Version Not Supported



Web cache

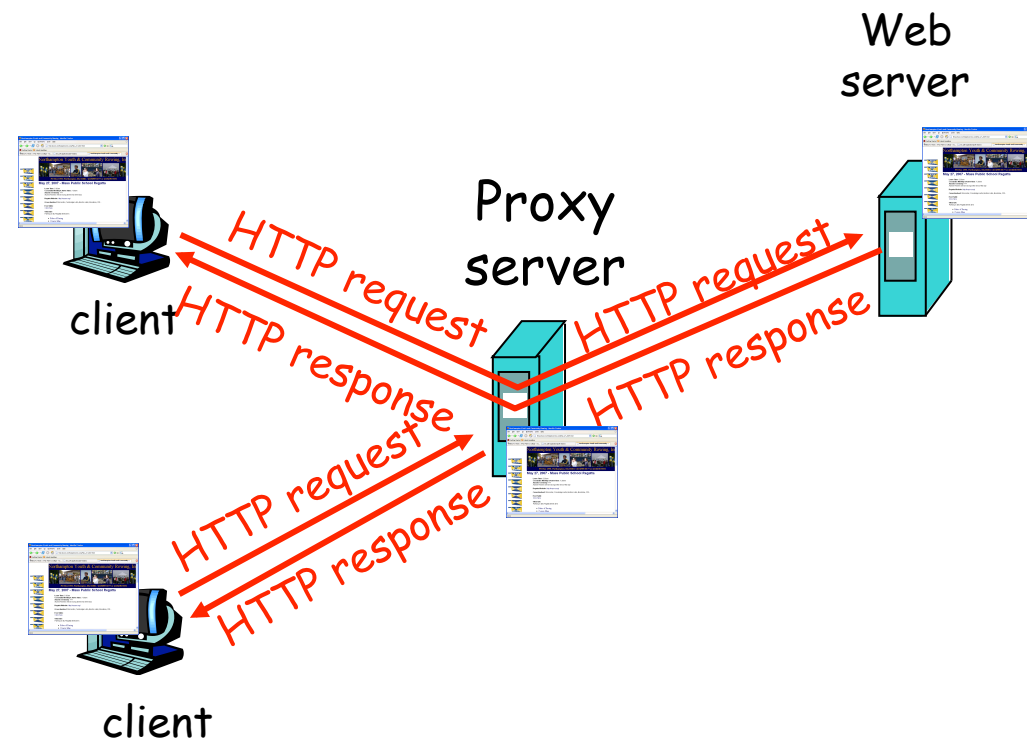
- “Cache”: buffer memory
- Similar notion of computer cache
 - L1 cache, L2 cache
 - “cache miss”, “cache hit”
- Case study:
 - Một tổ chức có một đường nối tới Internet
 - Tất cả lưu lượng truy cập web đều đi qua liên kết này
 - Nhiều NSD web có thể cùng truy nhập tới cùng một nội dung
 - Giải pháp cải tiến?

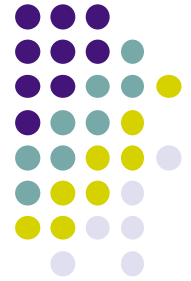


Sử dụng bộ đệm web - proxy



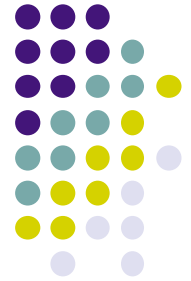
- NSD đặt tham số kết nối truy cập web của trình duyệt qua một máy chủ proxy
- trình duyệt gửi yêu cầu đến proxy
 - Miss: Proxy gửi yêu cầu tới máy chủ web, trả lời trình duyệt và lưu đệm đối tượng web
 - Hit: Proxy trả đối tượng web cho trình duyệt





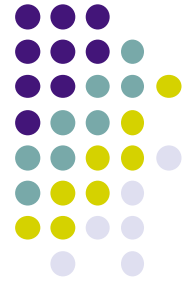
Web caches

- Proxy: Vừa là client, vừa là server
- Sử dụng bởi các ISP nhỏ, các tổ chức như trường học, công ty...
- Ảnh hưởng của proxy
 - Làm giảm lưu lượng web trên đường ra Internet
 - Có thể làm giảm thời gian đáp ứng
 - Thử phân tích vài trường hợp
 - cache hit
 - cache miss
 - proxy bị quá tải
 - Trang web thay đổi/trang web động?



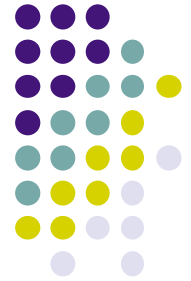
Local cache

- Các trang web còn có thể được lưu trên máy cục bộ
- Sử dụng local cache để
 - Duyệt web offline
 - Duyệt các trang web hiệu quả hơn:

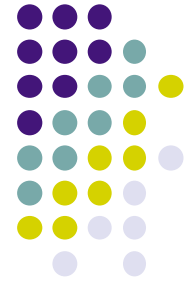
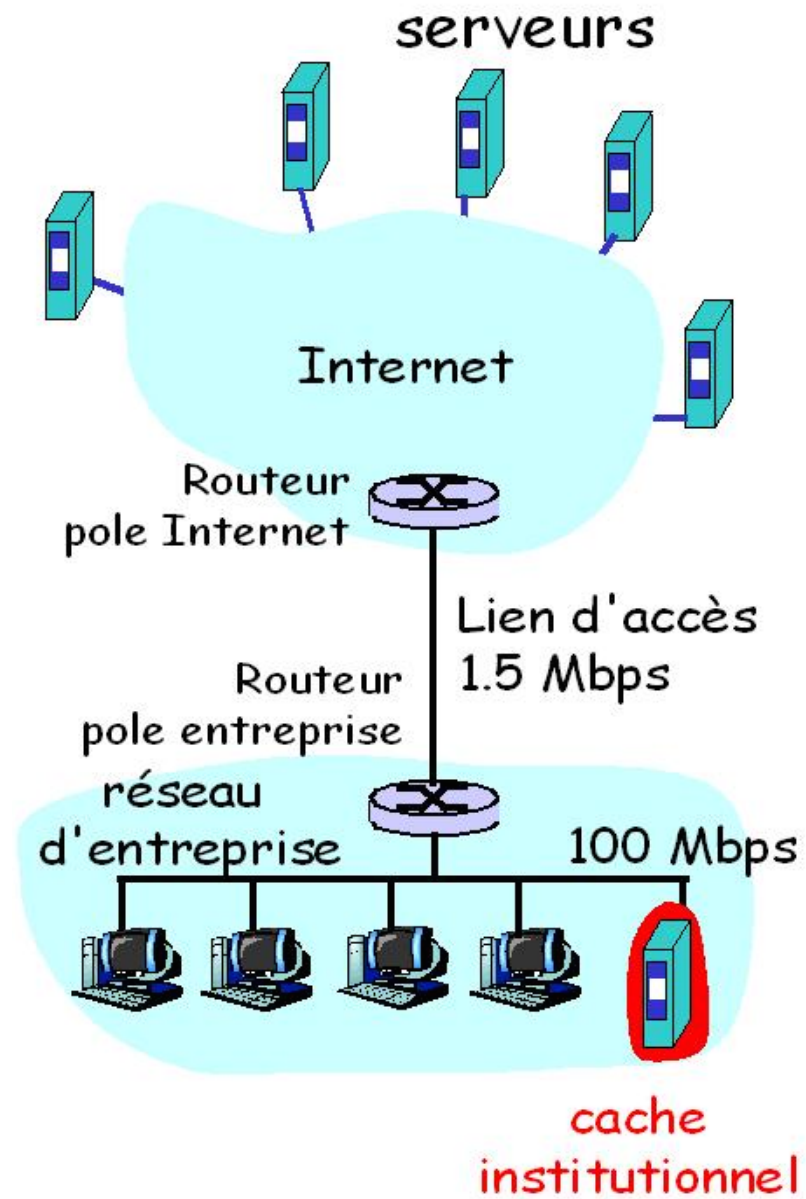


Bài tập

- Mạng công ty kết nối với Internet bằng liên kết tốc độ 1,5 Mbps.
- Các file cần tải có kích thước trung bình 900 kbits
- Tần suất trình duyệt trong máy công ty yêu cầu file trên web = 1,5 lần/s.
- Thời gian gửi yêu cầu HTTP từ router Internet-Server-nhận trả lời = 2 (s)
- Tính thời gian đáp ứng trung bình



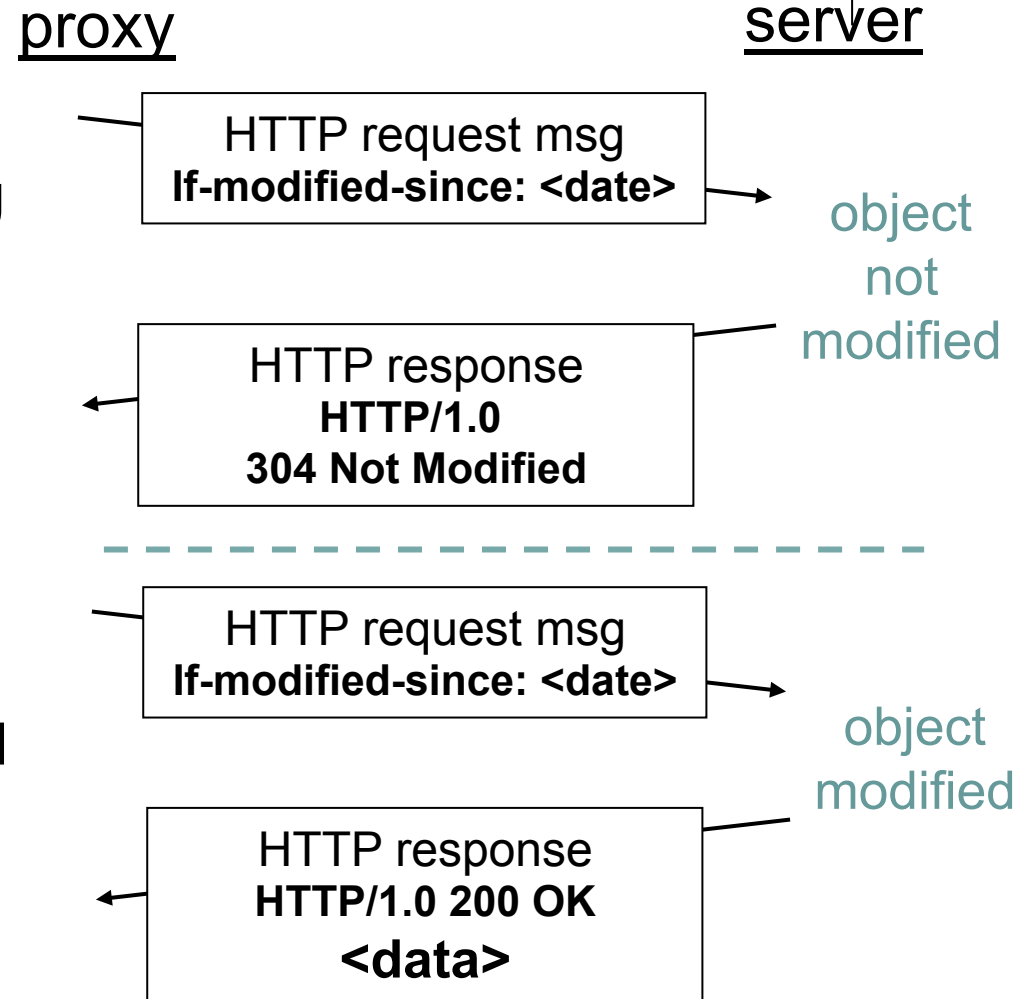
- Trong trường hợp các liên kết đã có tải. Thời gian phát dữ liệu trên liên kết có tải:
 - $= W_m / (1 - \rho)$,
 - W_m : thời gian phát không tải
 - ρ : tải
 - Tìm thời gian đáp ứng trung bình.
- Giả sử có cache
 - Cache hit: 0,4.
 - Tính thời gian đáp ứng trung bình
 - Dùng công thức: $T_m = \sum p_i * T_i$,
 - p_i xác suất có thời gian đáp ứng T_i



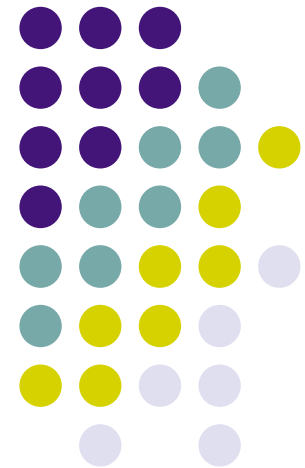
Phương thức GET có điều kiện



- **Mục đích:** Máy chủ sẽ không gửi đối tượng web nếu proxy còn lưu giữ thông tin cập nhật
- Proxy: chỉ ra thời gian cũ của đối tượng
If-modified-since: <date>
- server: Xác nhận lại có thay đổi hay không:
HTTP/1.0 304 Not Modified



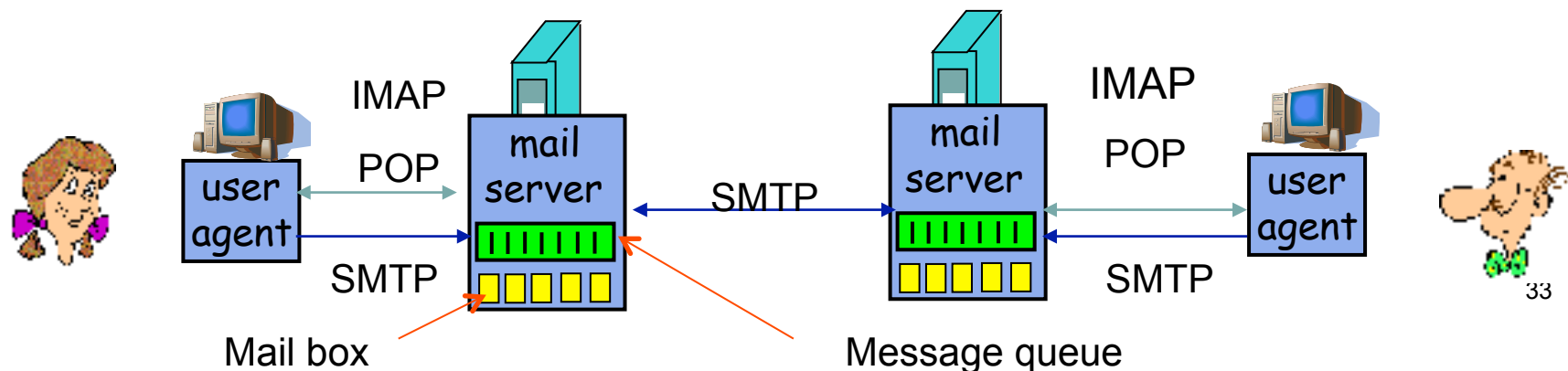
Case study 2: Email

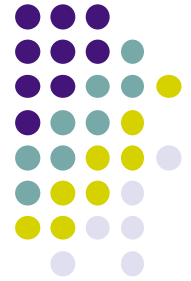




Thư điện tử

- MUA (Mail User Agent)
 - Lấy thư từ máy chủ, gửi thư đến máy chủ
 - e.g. Outlook, Thunderbird...
- MTA (Mail Transfer Agent): :
 - Chứa hộp thư đến của NSD (mail box)
 - Hàng đợi để gửi thư đi
 - e.g. Sendmail, MS Exchange...
- Giao thức:
 - Chuyển thư: SMTP-Simple Mail Transfer Protocol
 - nhận thư
 - POP – Post Office Protocol
 - IMAP – Internet Mail Access Protocol

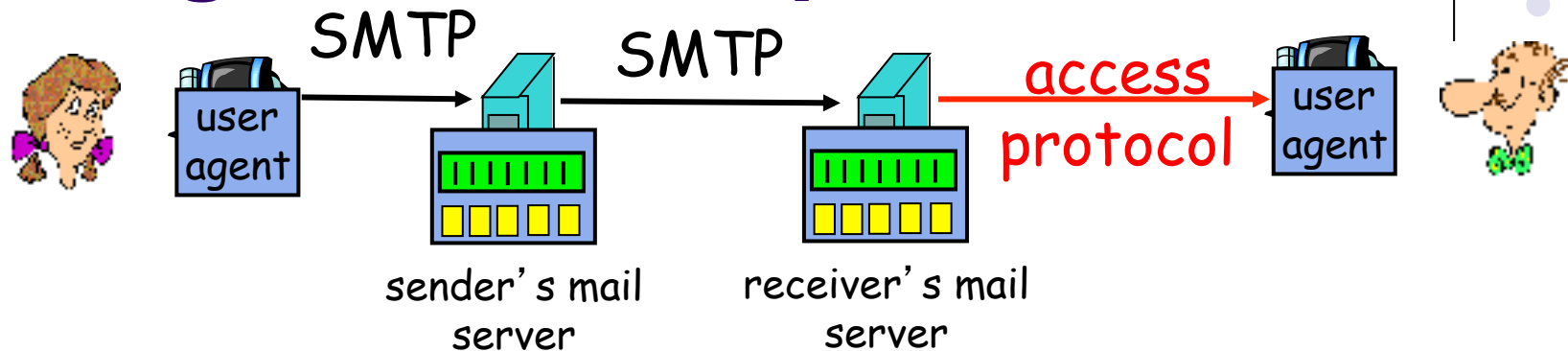




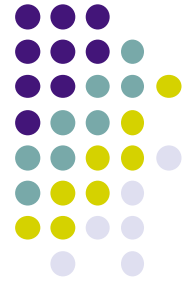
Giao thức SMTP

- RFC 2821
- TCP, port 25: Chuyển thư từ client đến server và giữa các server với nhau
- Tương tác yêu cầu/trả lời
 - Yêu cầu: Lệnh với mã ASCII
 - Trả lời: mã trạng thái và dữ liệu

Các giao thức nhận thư



- POP: Post Office Protocol [RFC 1939]
 - Đăng nhập và lấy hết thư về
- IMAP: Internet Mail Access Protocol [RFC 1730]
 - Phức tạp hơn POP
 - Cho phép lưu trữ và xử lý thư trên máy chủ
 - Quota?



Web Mail

- Sử dụng Web browser như một MUA
- MUA và MTA giao tiếp thông qua HTTP
- Mails được lưu trữ trên máy chủ
- E.g.
 - Gmail,
 - Hotmail,
 - Yahoo! Mail, etc.
- Ngày nay, rất nhiều các MTA cho phép truy cập thông qua giao diện web
 - <http://mail.hut.edu.vn>
 - <http://mail.fit.hut.edu.vn>

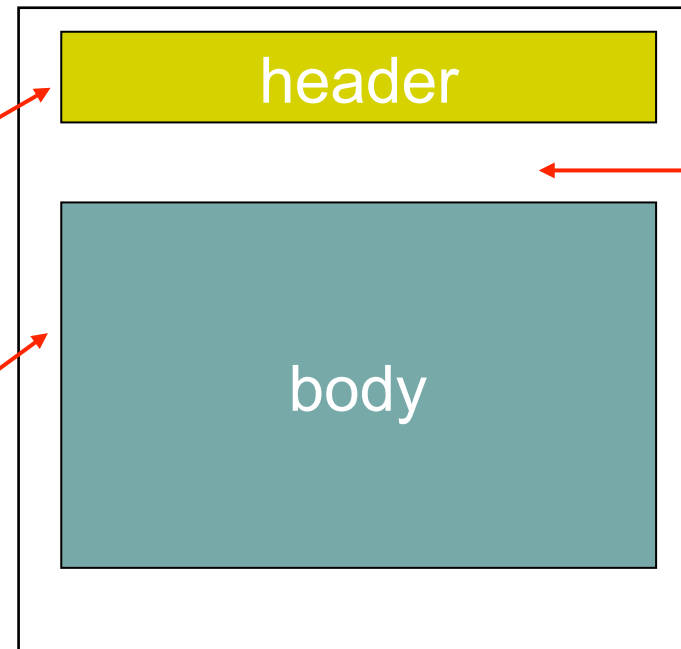
Khuôn dạng thông điệp thư điện tử



SMTP: Giao thức để truyền thư

RFC 822: Định nghĩa khuôn dạng

- Phần đầu
 - To:
 - From:
 - Subject:
- Phần thân
 - mã hóa dưới dạng mã ASCII

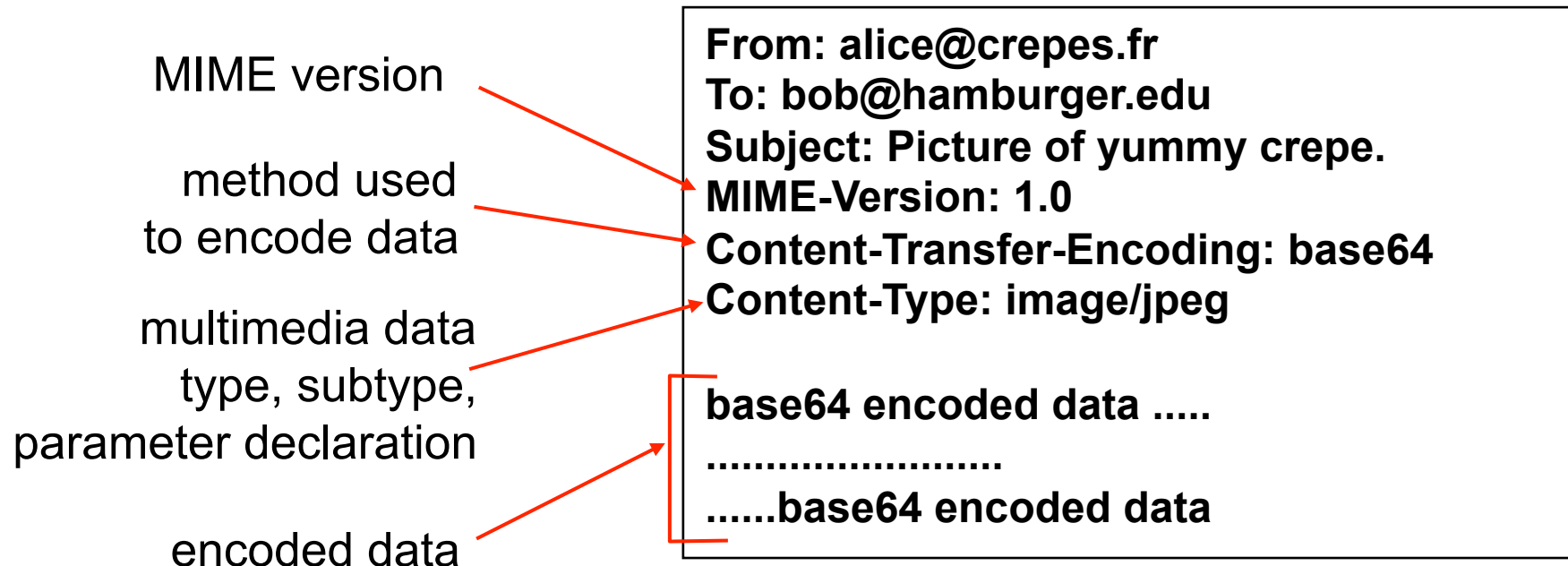


blank
line

Đề chuyển dữ liệu đa phương tiện: multimedia extensions

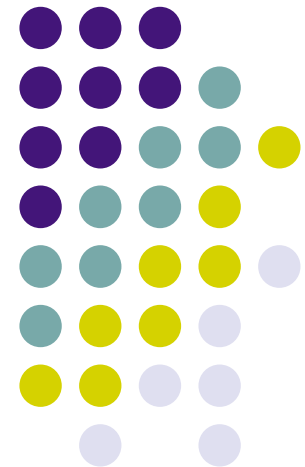


- MIME: multimedia mail extension, RFC 2045, 2056
- Thêm một dòng trong phần đầu chỉ rõ khuôn dạng dữ liệu gửi đi

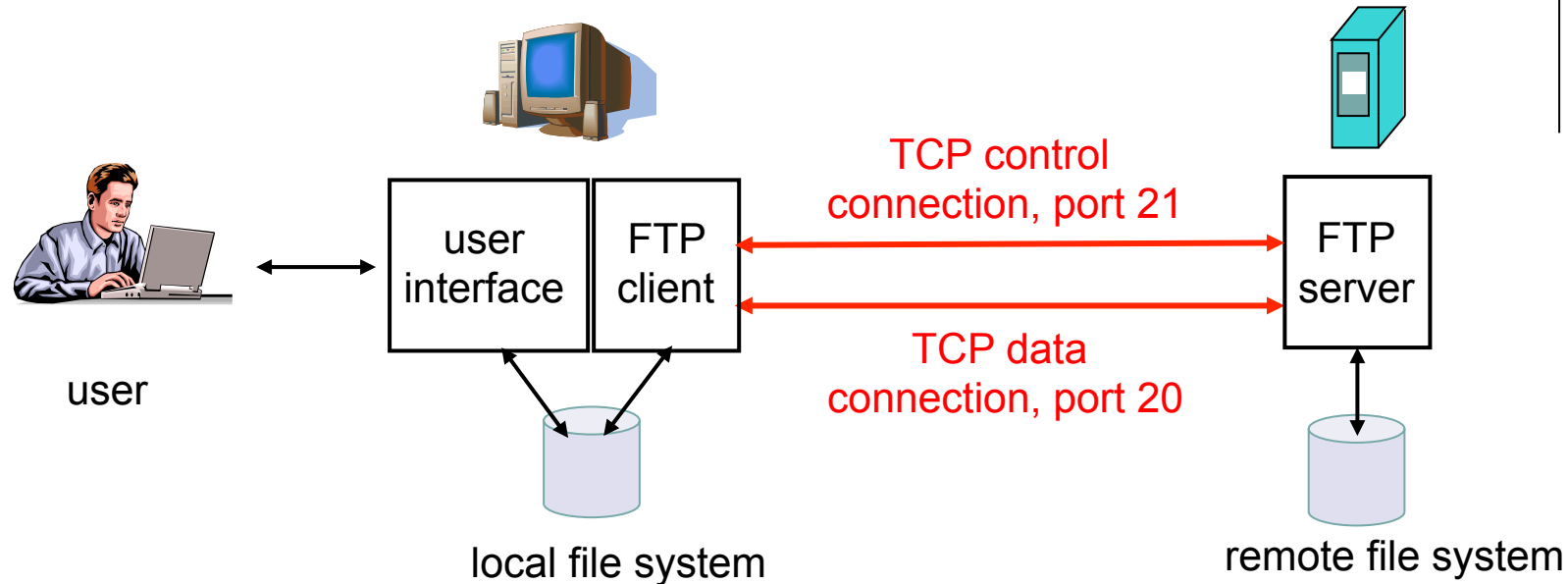


Case Study 3

Ứng dụng truyền tệp



FTP: File Transfer Protocol



- Mô hình Client-server
- Trao đổi file giữa các máy
- RFC 959
- Sử dụng TCP, cổng 20, 21
- Điều khiển **Out-of-band** :
 - Lệnh của FTP : cổng 21
 - Dữ liệu: cổng 20
- NSD phải đăng nhập trước khi truyền file
- Một số server cho phép NSD với tên là anonymous



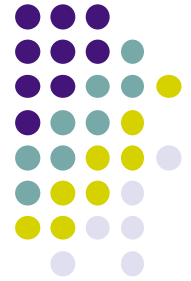
Lệnh và mã trả lời

Một số ví dụ

- **USER *username***
- **PASS *password***
- **LIST** : trả về danh sách file
- **RETR filename** Lấy file
- **STOR filename** Đặt file lên máy chủ

Ví dụ về mã trả lời

- **331 Username OK, password required**
- **125 data connection already open; transfer starting**
- **425 Can't open data connection**
- **452 Error writing file**



Ví dụ về ftp client

Command line

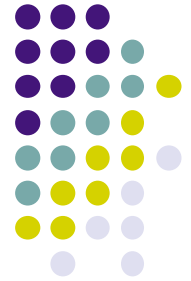
C:\Documents and Settings\hongson>ftp

ftp> ?

Commands may be abbreviated. Commands are:

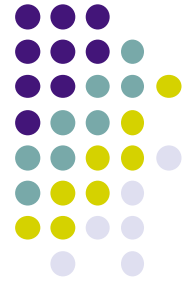
!	delete	literal	prompt	send
?	debug	ls	put	status
append	dir	mdelete	pwd	trace
ascii	disconnect	mdir	quit	type
bell	get	mget	quote	user
binary	glob	mkdir	recv	verbose
bye	hash	mls	remotehelp	
cd	help	mput	rename	
close	lcd	open	rmdir	

GUI FTP clients: IE, Firefox, GFTP,



Tóm tắt

- Mô hình ứng dụng
 - Client-server vs. P2P
- Một số ứng dụng và giao thức
 - HTTP
 - Mail
 - FTP
- Về nhà, hãy tìm hiểu thêm
 - P2P
 -
- Giao diện lập trình Socket



HTTPS

- HTTPS
- Cơ bản về ATTT
 - Assymetric system: Public key – Private key
 - Chứng chỉ số (certificate)
 - Symmetric system
 - Chữ ký điện tử
- Ví dụ ứng dụng trong HTTPS.



Acknowledgment

- Bài giảng có sử dụng các tư liệu và hình vẽ từ:
 - Tài liệu của trường đại học Keio và Ritsumeikan
 - Tài liệu “Computer Network, a top down approach” của J.F Kurose và K.W. Ross