

A decorative graphic on the left side of the slide, featuring a blue square, a red square, and a yellow square, with a black crosshair and a horizontal line passing through them.

Internet và các giao thức

Internet and Protocols

Nội dung học phần Internet và giao thức (30 tiết=3đvht, Lớp chính quy)

- **Lý thuyết:** 24 tiết
 - C1- Các nguyên lý lớp ứng dụng mạng Internet
 - C2- WEB và giao thức http
 - C3- Truyền tệp và thư điện tử
 - C4- Dịch vụ tên miền DNS
 - C5- Các ứng dụng ngang hàng P2P
 - C6- Kết nối mạng đa phương tiện
 - C7- Xu hướng phát triển ứng dụng và dịch vụ trên nền Internet
 - 2 tiết kiểm tra
 - 2 tiết ôn tập
- **Bài tập:** 6 tiết – làm nhóm.
- **Thi cuối kỳ:** Thi viết
- **Giờ tự học:** 15 tiết tự học trên lớp, ~120 giờ tự học ngoài lớp

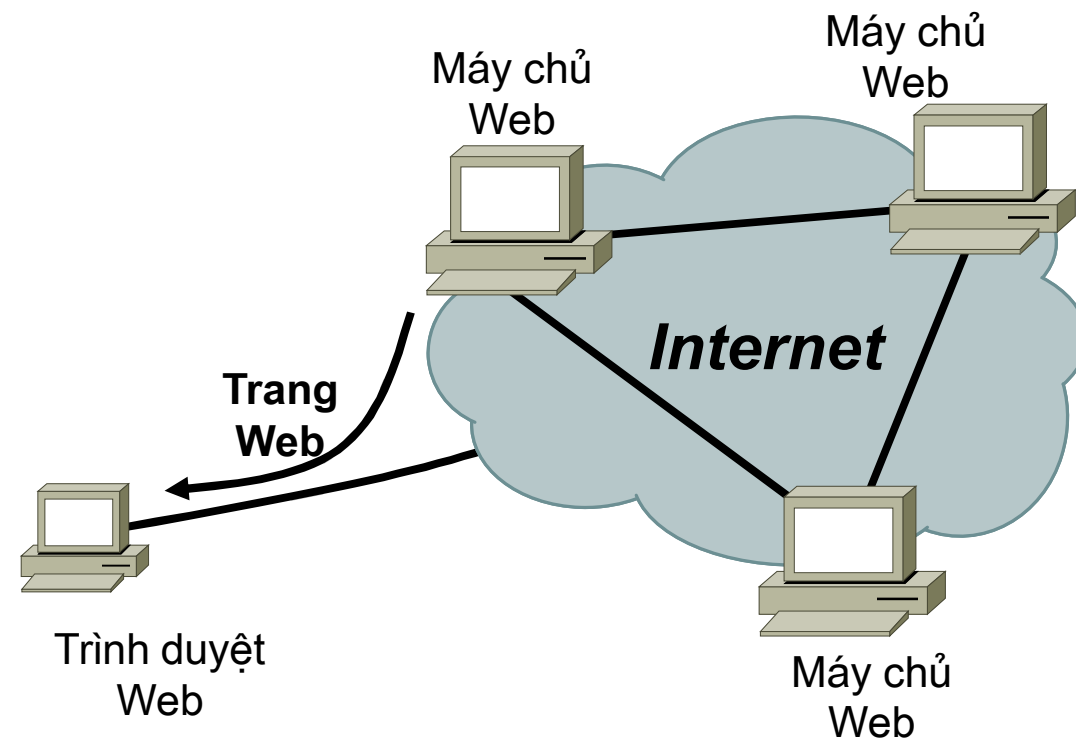
Chương 2: Web và HTTP

- Nội dung chính
 - Tổng quan về HTTP
 - Các kết nối HTTP
 - Khuôn dạng bản tin HTTP
 - Tương tác người sử dụng-máy chủ: Cookie
 - Lưu đệm Web (Web caching)
 - Cơ chế GET có điều kiện



WWW- World Wide Web

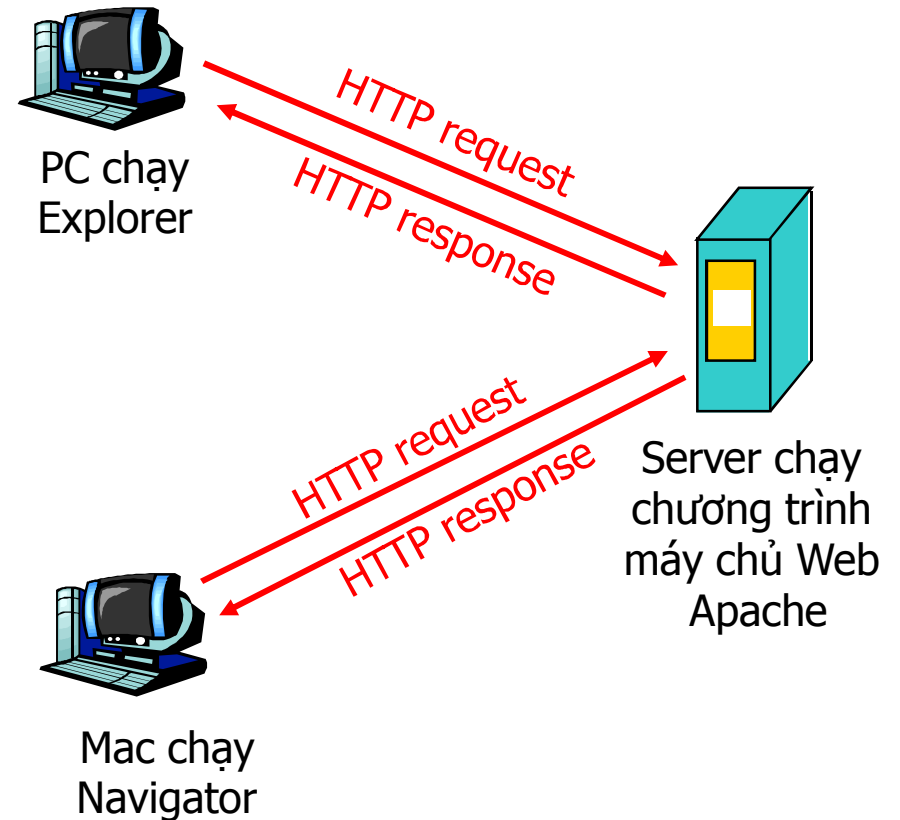
- Dịch vụ rất phổ biến
- Biểu diễn thông tin dưới dạng siêu văn bản HTML
- Hoạt động theo mô hình client/server



Tổng quan về HTTP

HTTP: hypertext transfer protocol (giao thức truyền siêu văn bản)

- RFC 1945, RFC 2616
- Là giao thức lớp ứng dụng của Web
- Mô hình khách chủ (client/server)
 - **client**: trình duyệt yêu cầu (request), nhận, hiển thị các đối tượng Web
 - **server**: Server Web gửi các đối tượng để đáp ứng (response) lại yêu cầu của client



Web và HTTP

Trang web chứa các đối tượng (object)

- **Đối tượng** có thể là file HTML, hình ảnh JPEG, ứng dụng Java, file audio,...
- **Trang web** chứa file HTML cơ bản bao gồm nhiều đối tượng tham chiếu
- Mỗi đối tượng được xác định bằng địa chỉ URL
 - Ví dụ về URL:

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

host name

path name



Các kết nối HTTP

Sử dụng TCP:

1. client khởi tạo kết nối TCP (tạo socket) tới server, cổng 80
2. server chấp nhận kết nối TCP từ client
3. Các bản tin HTTP (bản tin giao thức lớp ứng dụng) được trao đổi giữa trình duyệt (HTTP client) và server web (HTTP server)
4. Đóng kết nối TCP

HTTP không có trạng thái

- server không duy trì thông tin về những yêu cầu trước đó của client

Bên cạnh đó

Các giao thức có duy trì trạng thái thì rất phức tạp!

- ❑ Trạng thái cũ (lịch sử) phải được lưu trữ
- ❑ Nếu server/client bị phá hỏng, trạng thái có thể không đồng nhất và phải thống nhất lại



Kết nối HTTP

1: Kết nối không ổn định

Non-persistent HTTP

- Nhiều nhất **một** đối tượng được gửi qua một kết nối TCP .

2: Kết nối ổn định

Persistent HTTP

- **Nhiều** đối tượng có thể được gửi trên một kết nối TCP giữa client và server.

HTTP không ổn định (Non persistent HTTP)

Giả sử người sử dụng vào URL

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

11 đối tượng: Tập HTML cơ bản và 10 hình ảnh jpeg

1a. Client HTTP **khởi tạo** kết nối TCP tới server HTTP (process-tiến trình) tại `www.someSchool.edu` trên cổng 80

2. Client HTTP gửi bản tin yêu cầu HTTP (chứa URL) vào socket kết nối TCP. Bản tin này cho biết client muốn đối tượng `someDepartment/home.index`

1b. Server HTTP tại trạm chủ `www.someSchool.edu` chờ kết nối TCP ở cổng 80. “chấp nhận” kết nối và báo cho client

3. Server HTTP nhận bản tin yêu cầu, tạo ra bản tin đáp ứng chứa đối tượng yêu cầu và gửi bản tin vào socket của nó.

5. Client HTTP nhận bản tin đáp ứng chứa file html, hiển thị html. Phân tích cú pháp file html, tìm 10 đối tượng jpeg tham chiếu.

4. Server HTTP **đóng** kết nối TCP.

6. **Lặp lại** các bước từ 1-5 với từng đối tượng jpeg đó.

time

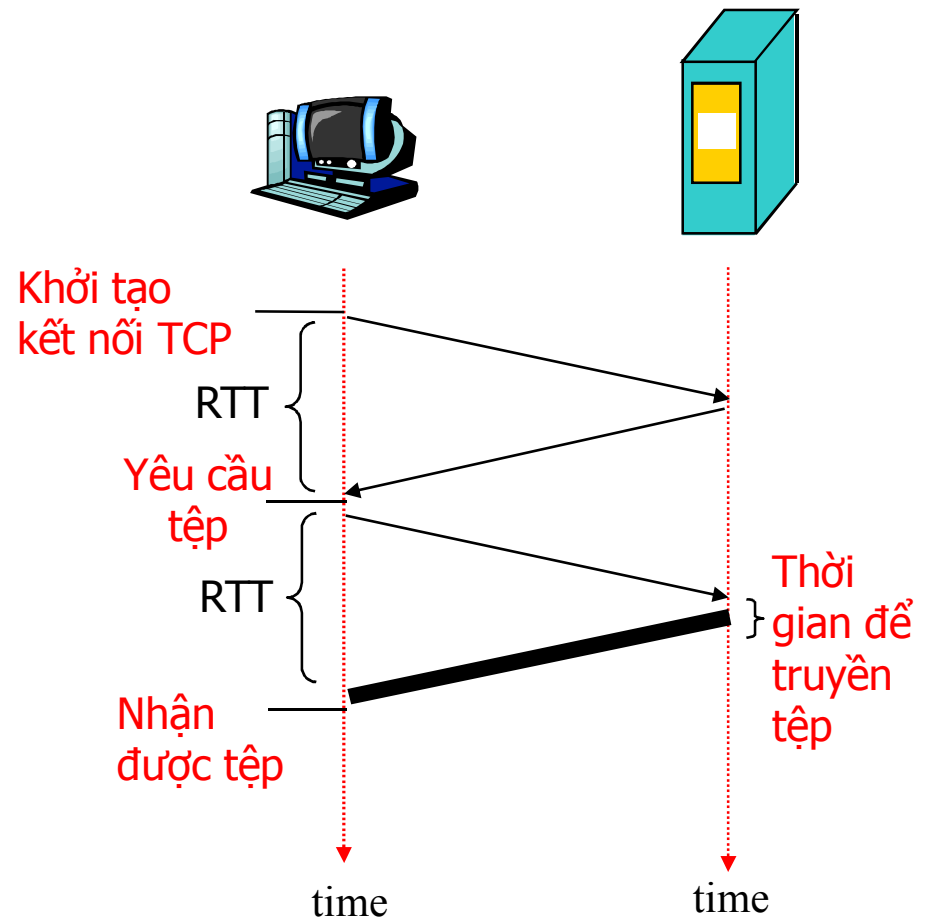
HTTP không ổn định (Non-Persistent HTTP): Thời gian đáp ứng

Định nghĩa RTT: thời gian cho một gói tin đi từ client đến server và phản hồi trở lại.

Thời gian đáp ứng:

- Một RTT dùng để khởi tạo kết nối TCP
- Một RTT cho yêu cầu HTTP và một vài byte đầu của đáp ứng HTTP được phản hồi lại
- Thời gian truyền file

Tổng = $2RTT +$ thời gian truyền file



HTTP ổn định (Persistent HTTP)

Non-persistent HTTP :

- Mỗi đối tượng cần 2 RTT
- Mào đầu OS cho mỗi kết nối TCP
- Các trình duyệt thường mở song song một số kết nối TCP để lấy các đối tượng tham chiếu

Persistent HTTP

- Server để kết nối mở sau khi gửi đáp ứng
- Các bản tin HTTP của cùng cặp client/server được gửi trên kết nối mở đó
- client gửi các yêu cầu bất cứ khi nào nó gặp đối tượng tham chiếu
- Chỉ cần một RTT cho tất cả các đối tượng tham chiếu

Khuôn dạng bản tin HTTP

- Hai loại bản tin HTTP: *request (yêu cầu)*, *response (đáp ứng)*
- Bản tin **yêu cầu** HTTP :
 - ASCII (khuôn dạng mà con người đọc được)

Dòng yêu cầu
(các lệnh GET, POST,
HEAD)

Các dòng tiêu đề

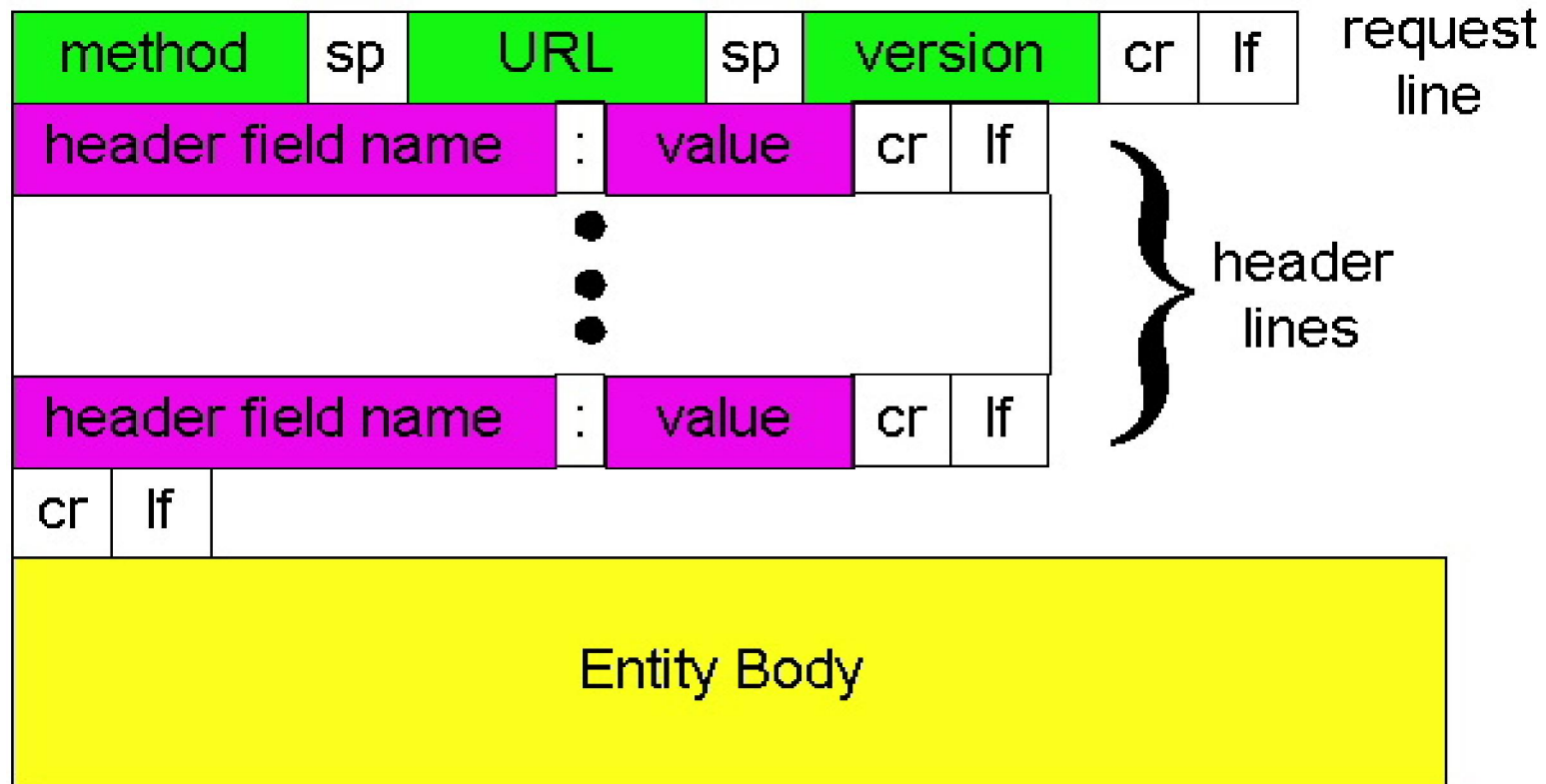
```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

Xuống dòng,
chuyển dòng
chỉ thị kết thúc
bản tin

(extra carriage return, line feed)

Lệnh: command

Khuôn dạng chung của bản tin yêu cầu HTTP



Tải lên đầu vào khuôn mẫu (Uploading form input)

Phương thức POST:

- Trang web thường bao gồm đầu vào khuôn mẫu (form input)
- Đầu vào được tải lên server trong khối thực thể

Phương thức URL :

- Sử dụng phương thức GET
- Input được tải trong trường URL trong dòng lệnh yêu cầu:

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



Các kiểu phương thức

HTTP/1.0

- GET
- POST
- HEAD
 - Yêu cầu server bỏ đối tượng yêu cầu ra khỏi đáp ứng

HTTP/1.1

- GET, POST, HEAD
- PUT
 - Tải lên file trong khối thực thể vào đường dẫn xác định trong trường URL
- DELETE
 - Xoá bỏ file xác định trong trường URL

Bản tin đáp ứng HTTP

Dòng trạng thái
(giao thức
mã trạng thái
mệnh đề trạng thái)

Các dòng tiêu đề

```
HTTP/1.1 200 OK
Connection close
Date: Sat, 07 Jul 2007 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Sun, 6 May 2007 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html
```

Dữ liệu, tức
là
Tập HTML
yêu cầu

```
(data data data data data ... )
```




Các mã trạng thái phản hồi HTTP

Trong dòng đầu bản tin phản hồi server→client
Một vài mã mẫu:

200 OK

- Yêu cầu thành công, đối tượng yêu cầu xuất hiện phía sau trong bản tin này.

301 Moved Permanently

- Đã bỏ đối tượng yêu cầu, vị trí mới sẽ được đặc tả phía sau trong bản tin này (Location:)

400 Bad Request

- Server không hiểu bản tin yêu cầu

404 Not Found

- Tài liệu yêu cầu không tìm thấy trong server này

505 HTTP Version Not Supported

- Phiên bản giao thức HTTP yêu cầu không được máy chủ hỗ trợ.



Tự thử nghiệm HTTP (từ phía client)

1. Telnet tới Web server yêu thích :

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

Mở kết nối TCP tới cổng 80 (cổng mặc định của server HTTP) tại cis.poly.edu. Toàn bộ những gì ta gõ sẽ được gửi tới cổng 80 tại địa chỉ cis.poly.edu

2. Gõ yêu cầu GET HTTP :

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

Khi gõ lệnh này (ký hiệu xuống dòng 2 lần), phía client gửi bản tin yêu cầu GET tối thiểu (nhưng vẫn hoàn chỉnh) tới server HTTP

3. Hãy xem bản tin phản hồi từ server HTTP!



Tương tác người dùng-máy chủ: Cookie

Rất nhiều trang Web sử dụng cookie. Các cookie (RFC 2965) cho phép điểm truy nhập bám vết người sử dụng.

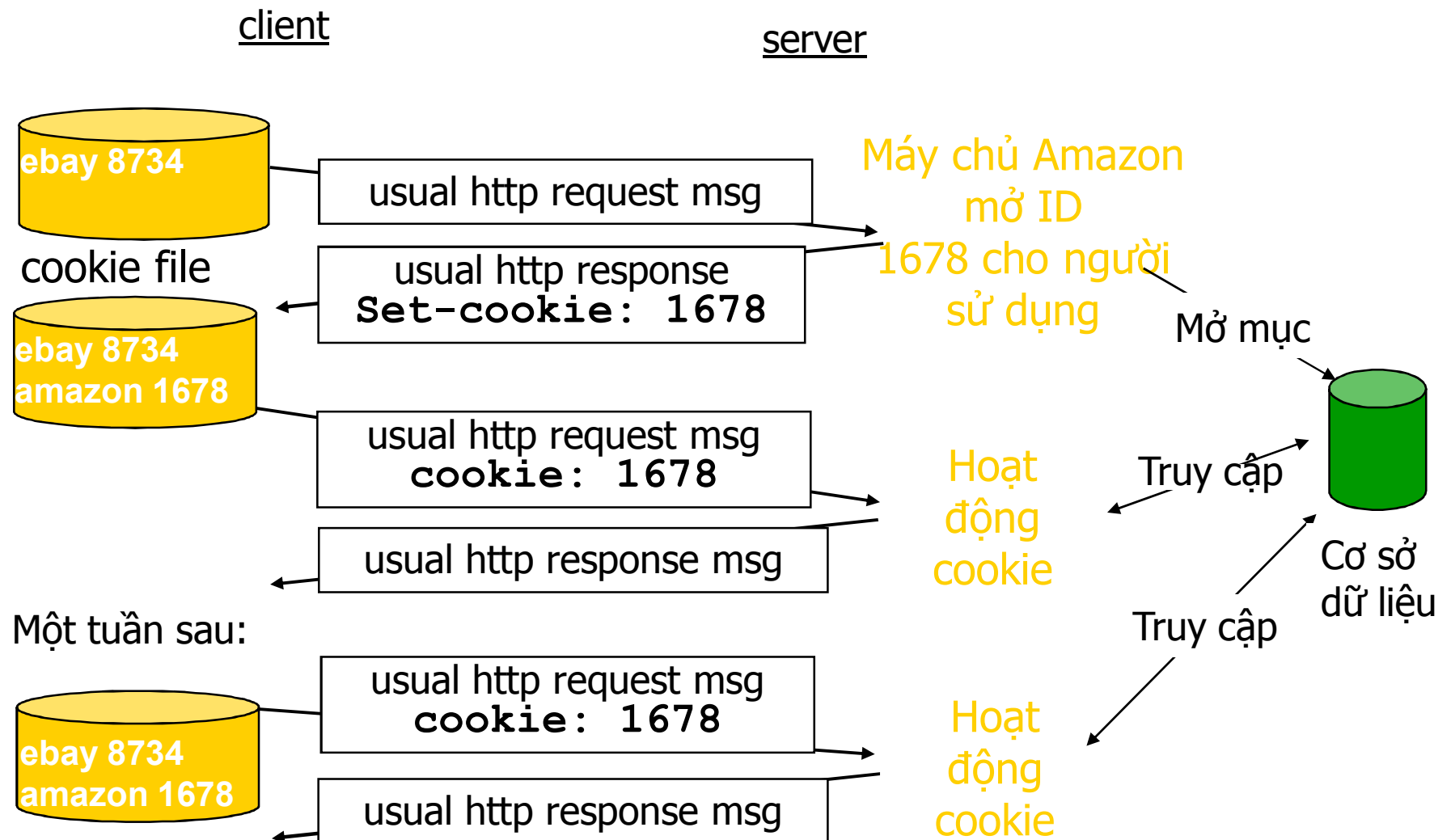
Bốn thành phần:

- 1) Dòng header cookie trong bản tin đáp ứng HTTP
- 2) Dòng header cookie trong bản tin yêu cầu HTTP
- 3) Tập cookie giữ ở máy trạm user, do trình duyệt của user điều khiển
- 4) Cơ sở dữ liệu đầu cuối (back-end) ở trang Web

Ví dụ:

- Susan thường truy nhập Internet từ PC
- Lần đầu vào một trang thương mại điện tử cụ thể
- Khi khởi động yêu cầu HTTP tới trang này thì trang này sẽ tạo ra:
 - ID duy nhất
 - Một mục trong cơ sở dữ liệu đầu xa cho ID đó

Cookies: giữ trạng thái người sử dụng





Cookies (cont.)

Cookies có thể mang lại :

- Ủy quyền
- Xe hàng hoá
- Khuyến nghị
- Trạng thái phiên người dùng (Web e-mail)

Cookies và sự riêng tư:

- ☐ cookies cho phép các trang web biết nhiều về bạn
- ☐ Có thể bạn sẽ cung cấp tên và địa chỉ email cho các trang này

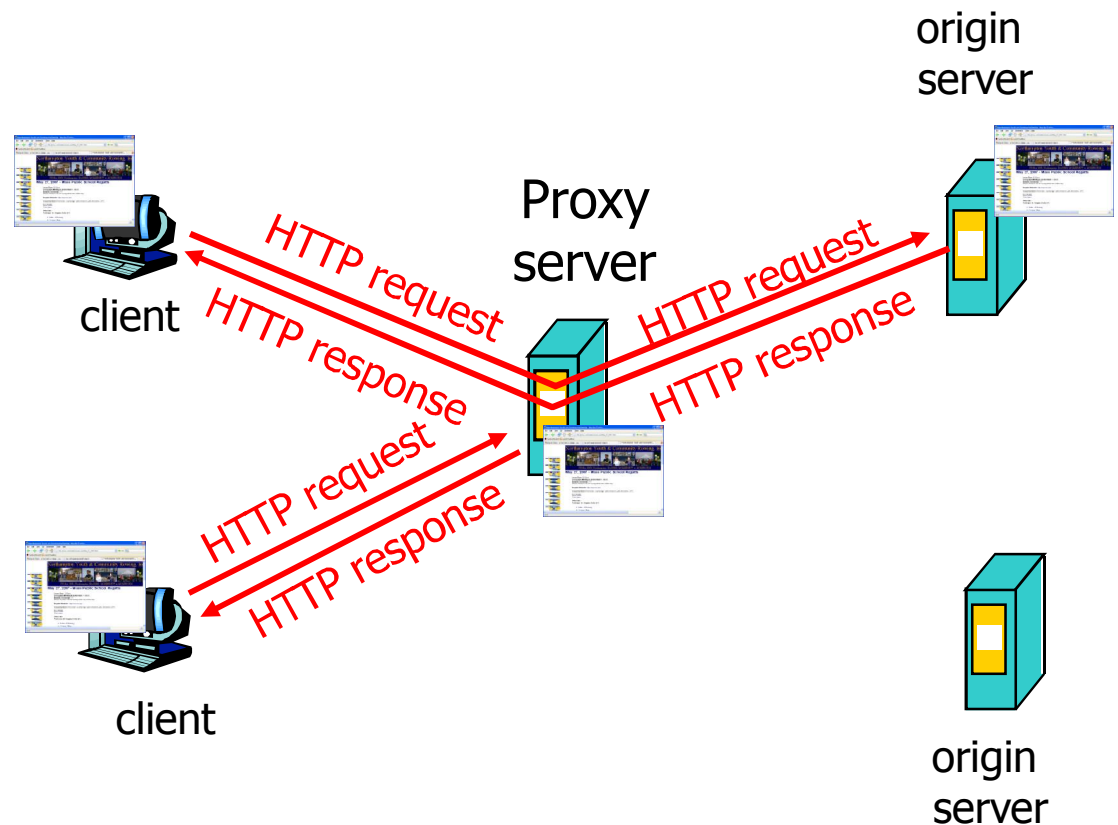
Làm thế nào để giữ "trạng thái":

- ☐ Điểm cuối giao thức: duy trì trạng thái ở phía gửi/nhận qua các giao dịch
- ☐ cookies: bản tin http mang trạng thái

Lưu đệm Web (Web cache - proxy server)

Mục đích/thành quả: đáp ứng yêu cầu của client mà không cần sự tham gia của server ban đầu

- Người sử dụng thiết lập trình duyệt: truy nhập web qua cache (máy chủ đệm là nơi lưu trữ/kho)
- Trình duyệt sẽ gửi toàn bộ yêu cầu HTTP tới cache
 - Đối tượng trong cache: cache sẽ trả về các đối tượng
 - Nếu không thì cache sẽ yêu cầu đối tượng từ server ban đầu, sau đó trả đối tượng cho client





Hiểu thêm về Web caching

- cache hoạt động như client và server
- Thường thì cache do ISP khởi tạo (trường đại học, công ty, ISP dân cư)

Vì sao lại lưu đệm Web?

- Để giảm thời gian đáp ứng yêu cầu của client
- Để làm giảm lưu lượng trên liên kết truy nhập của tổ chức.
- Internet đầy đặc cache: cho phép các nhà cung cấp nội dung “nghèo nàn” có thể truyền nội dung hiệu quả (cũng giống chia sẻ file P2P)

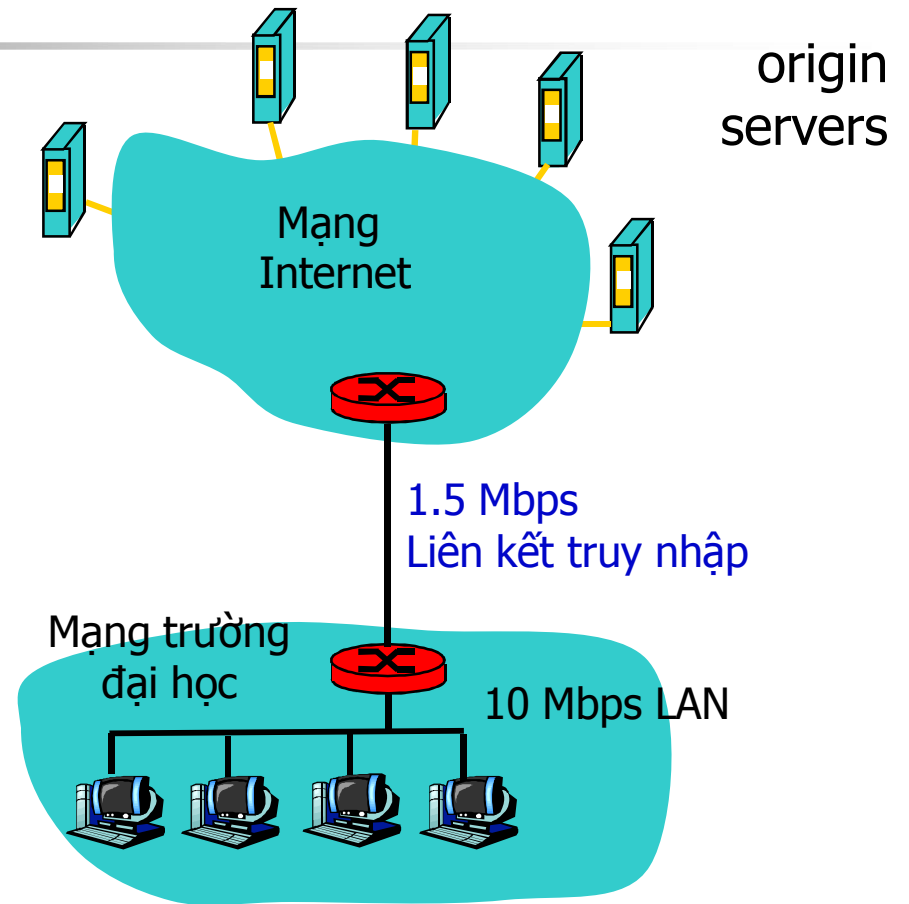
Ví dụ về cache

Giả thiết

- Kích thước trung bình của đối tượng là 100,000 bits
- Tốc độ yêu cầu trung bình từ các trình duyệt của trường tới server ban đầu là = 15 yêu cầu/giây
- Trễ từ router của trường tới bất kỳ server nào và phản hồi về là 2 giây

Kết quả là

- Sử dụng trên LAN = 15%
- Sử dụng trên đường truy nhập = 100%
- Tổng trễ = trễ Internet + trễ truy nhập + trễ LAN
= 2 sec + minutes + milliseconds



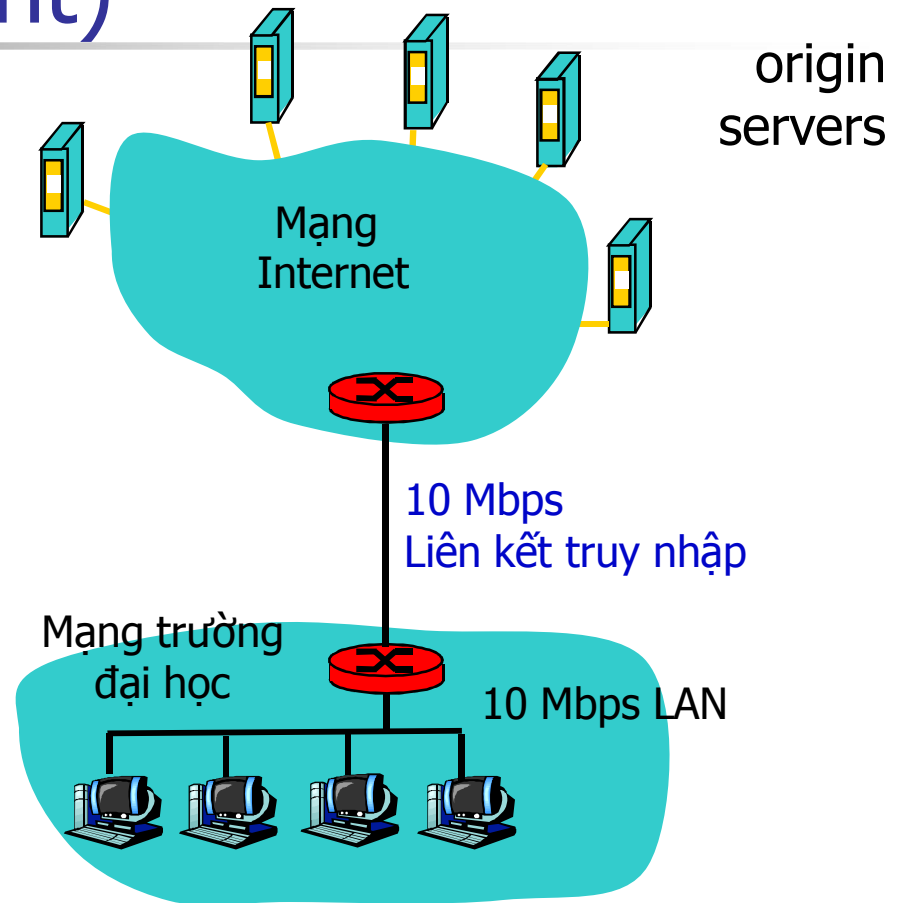
Ví dụ về cache (cont)

Giải pháp khả thi

- Tăng băng thông đường truy nhập, giả dụ là 10 Mbps

Kết quả

- Sử dụng trên LAN = 15%
- Sử dụng trên đường truy nhập = 15%
- Tổng trễ = Internet delay + access delay + LAN delay
= 2 sec + msec + msec
- Thường gây tăng chi phí



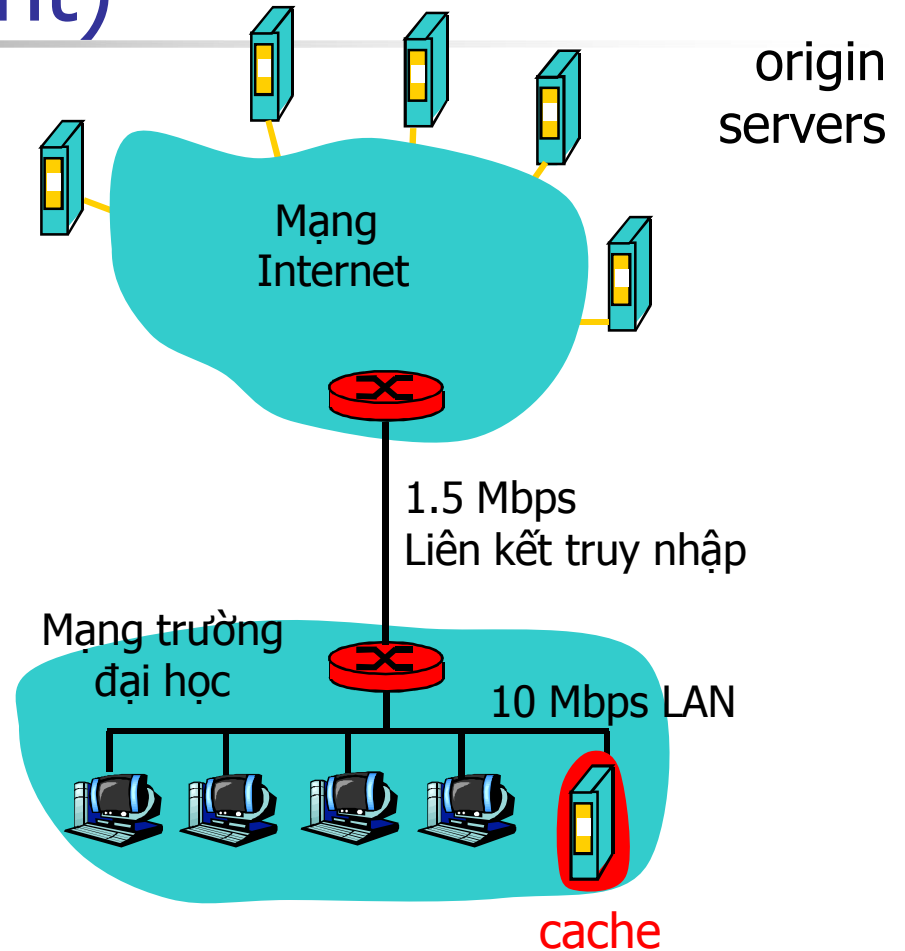
Ví dụ về cache (cont)

Giải pháp khả thi: lắp đặt cache

- Giả sử tỷ lệ truy vấn vào cache là 0.4

Kết quả

- 40% yêu cầu gần như thoả mãn tức thì
- 60% yêu cầu do server ban đầu đáp ứng
- Độ sử dụng trên liên kết truy nhập giảm xuống còn 60%, dẫn đến trễ giảm xuống rất nhỏ (10 msec)
- Tổng trễ trung bình = Internet delay + access delay + LAN delay = ($0,6 * 2\text{secs} + 0,4 * \text{milliseconds}$) = xấp xỉ 1,2 giây

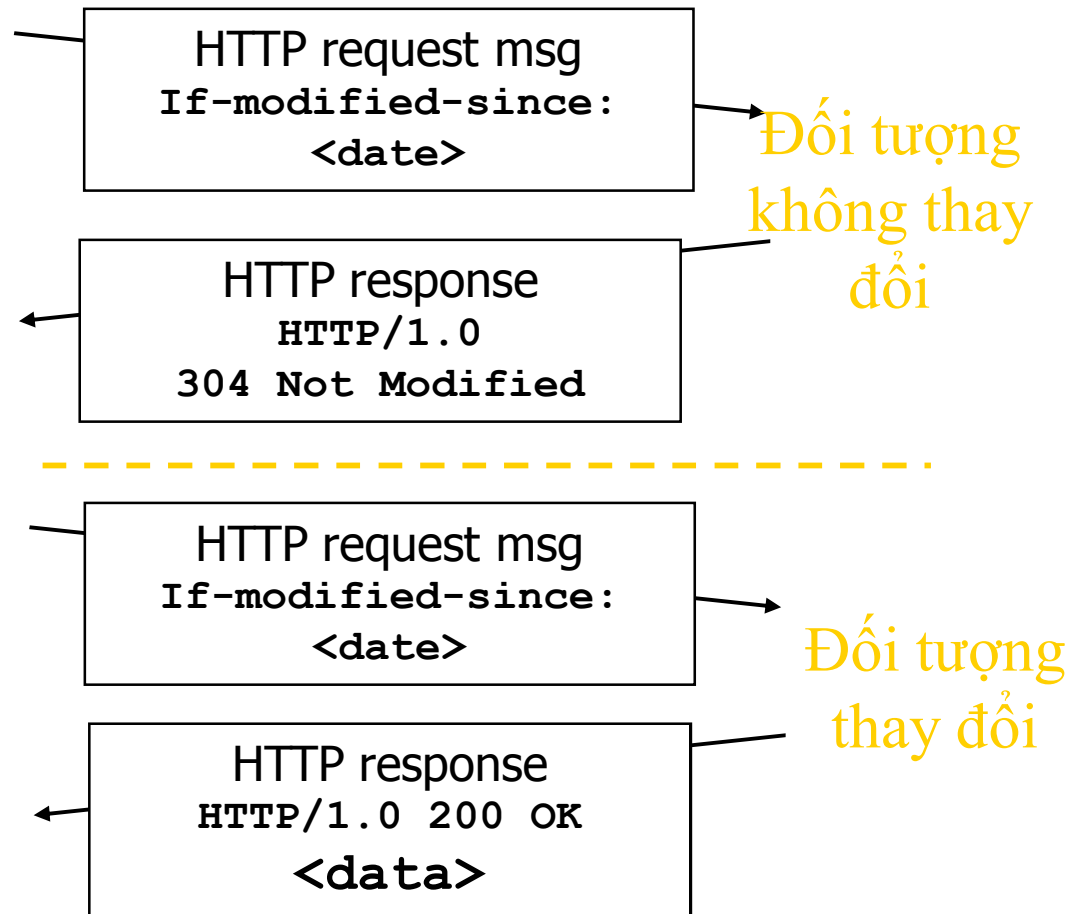


Conditional GET

- HTTP có cơ chế cho phép tra cứu việc đối tượng trong cache đã được cập nhật hay chưa
- Nó sẽ không gửi đối tượng nếu cache đã cập nhật phiên bản mới.
- cache: chỉ ra ngày của bản sao cached trong yêu cầu HTTP
`If-modified-since: <date>`
- server: phản hồi nhưng không gửi đối tượng nào nếu copy của cache là cập nhật :
`HTTP/1.0 304 Not Modified`

cache

server



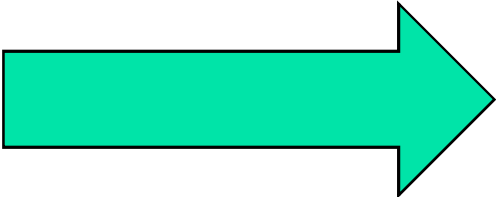


Tổng kết

- Khái niệm HTTP
 - Tổng quan HTTP
 - Phân biệt Non-persistent HTTP với Persistent HTTP
- Khuôn dạng bản tin HTTP: Bản tin yêu cầu, bản tin đáp ứng.
- Khái niệm cookies, lợi ích và những đặc điểm.
- Giải pháp cache.
- Khái niệm conditional Get và mối liên hệ với cache và server web.



Nội dung chuẩn bị

- 
- Nội dung về nhà và học buổi tối:
 - Chương 3: Truyền tệp và thư điện tử