

Chương I: CÁC KHÁI NIỆM CƠ BẢN

• • •

Trường Công Nghệ Thông Tin & Truyền Thông

Đại học Cần Thơ

Giảng viên: TS. Hà Duy An

NỘI DUNG

1. Tổng quan
2. Các thành phần của dịch vụ web
3. Giao thức HTTP
4. URL
5. HyperText & HyperLink
6. Web Cache

NỘI DUNG

1. Tổng quan
2. Các thành phần của dịch vụ web
3. Giao thức HTTP
4. URL
5. HyperText & HyperLink
6. Web Cache

1. TỔNG QUAN

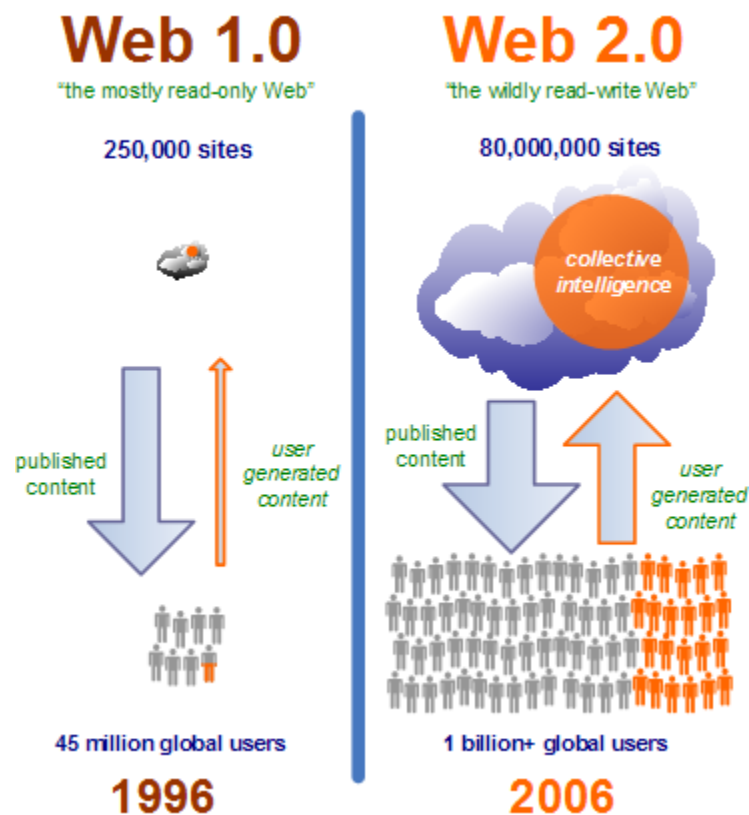
- **Web là gì?**

- World Wide Web, www, web, w3 là một dịch vụ trên Internet
- Là hình thức tổ chức thông tin phổ biến và tiện lợi nhất hiện nay
- Cho phép tra cứu tài nguyên thông tin qua các siêu văn bản (Hypertext) sử dụng các siêu liên kết (Hyperlink)
- Là các tài liệu văn bản thường được lưu trữ với phần mở rộng .html, .htm, ...
- Web được phát minh và đưa vào sử dụng vào khoảng năm 1990 bởi viện sĩ Viện Hàn lâm Anh, Tim Berners-Lee, tại CERN, từ đó đến nay web đã phát triển mạnh mẽ và trở thành một hệ thống siêu phương tiện (Hypermedia).



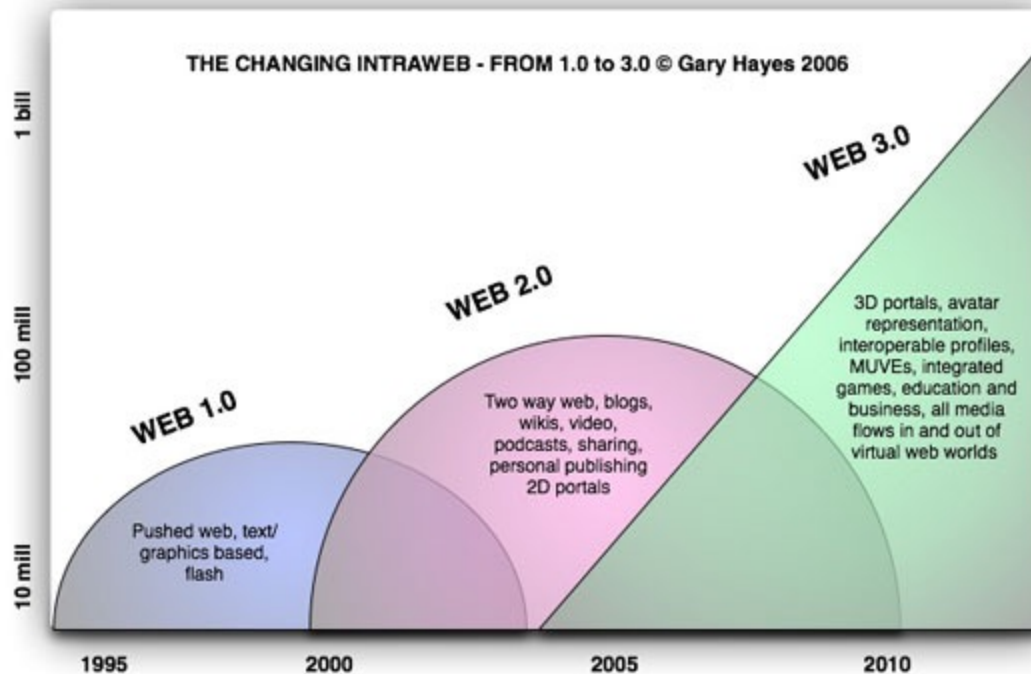
Các thể hệ web

- **Web 1.0 (Static web):**
 - Thông tin có tính chất một chiều
 - Sở hữu website chủ yếu là các công ty, tổ chức, hãng thông tấn xã
 - HTML
 - Thiếu khả năng tương tác
- **Web 2.0 (Dynamic web):**
 - Thông tin có tính chất hai chiều
 - Mạng tính cộng đồng
 - Kỹ thuật máy chủ, máy khách, cơ chế cung cấp nội dung, truyền thông
 - Web có vai trò nền tảng, có thể chạy mọi ứng dụng
 - Có thể chạy trên nhiều thiết bị với giao diện ứng dụng phong phú.



Các thể hệ web (tt)

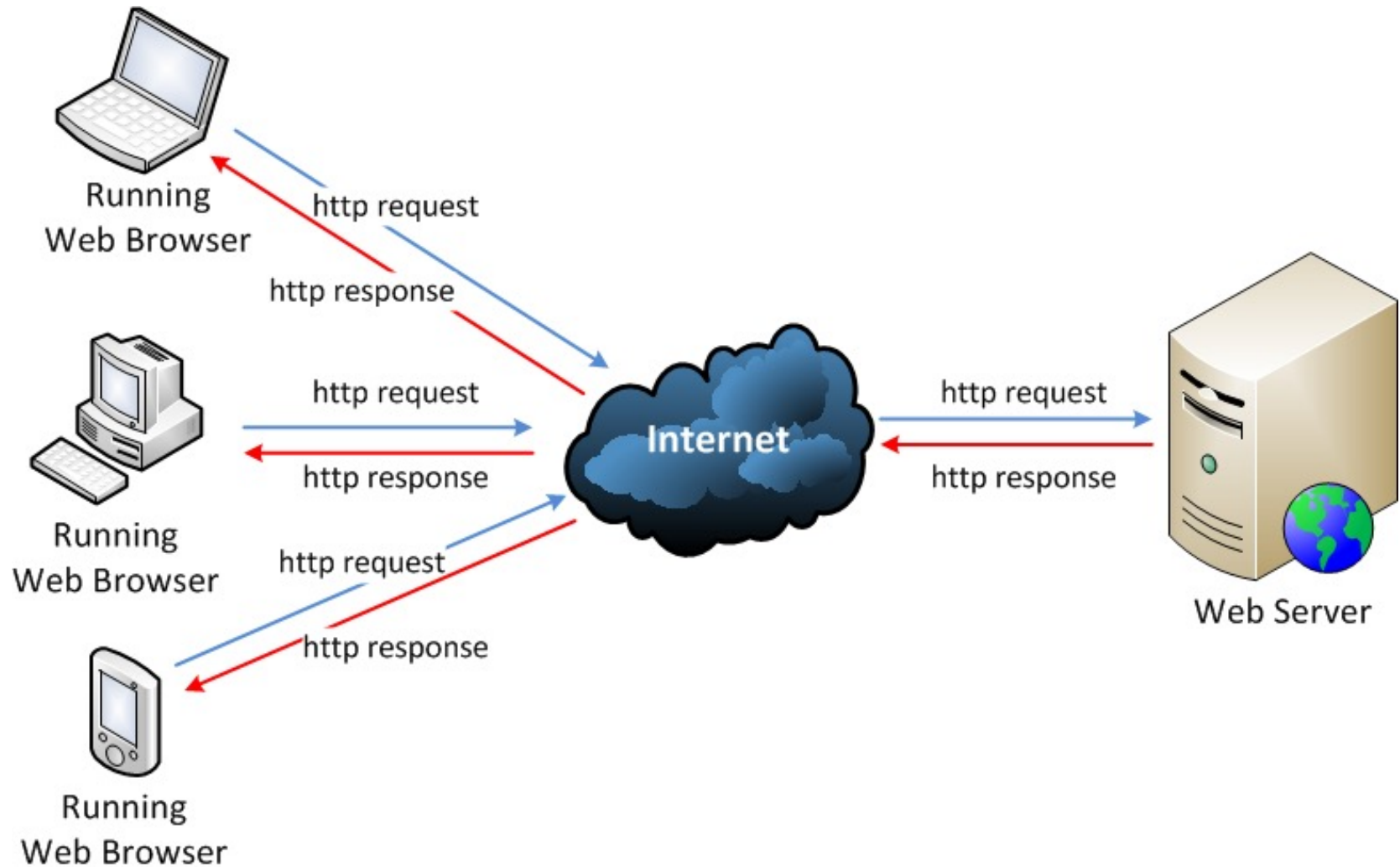
- **Web 3.0 (Web of Data): More Intelligent Web**
 - Là sự mở rộng của web 2.0
 - Các ứng dụng web ngữ nghĩa (semantic web)
 - Các ứng dụng web thời gian thực



NỘI DUNG

1. Tổng quan
- 2. Các thành phần của dịch vụ web**
3. Giao thức HTTP
4. URL
5. HyperText & HyperLink
6. Web Cache

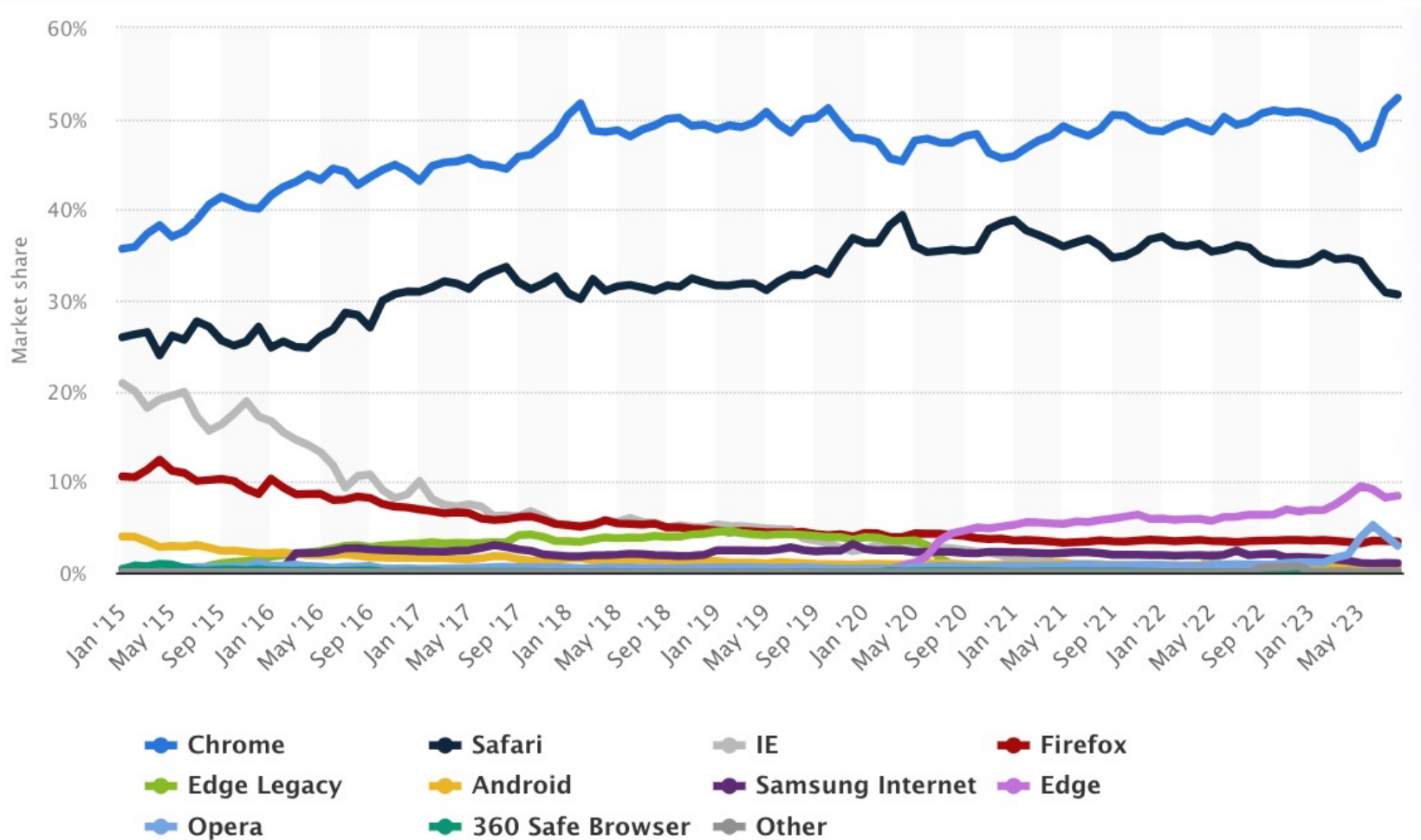
2. CÁC THÀNH PHẦN CỦA DỊCH VỤ WEB



Web client

- *Web Browser* (trình duyệt web): Internet Explorer, Firefox, Chrome, Safari, Opera, Netscape, Mozilla, ...
 - Được cài đặt tại máy khách.
 - Gửi các yêu cầu về web đến Web Server.
 - Nhận kết quả trả về từ Web Server.
 - Hiển thị kết quả.

Web client



Nguồn: statista.com

Web Browser

- Giải quyết vấn đề đa dạng của web browser:
 - Không đầu tư vào các chi tiết sai khác nhỏ
 - Theo các chuẩn chung
 - Sử dụng HTML đúng cú pháp chuẩn, có cấu trúc chính xác, rõ ràng
 - Không sử dụng các HTML elements được hỗ trợ đặc biệt chỉ bởi một hai vài trình duyệt

Web server

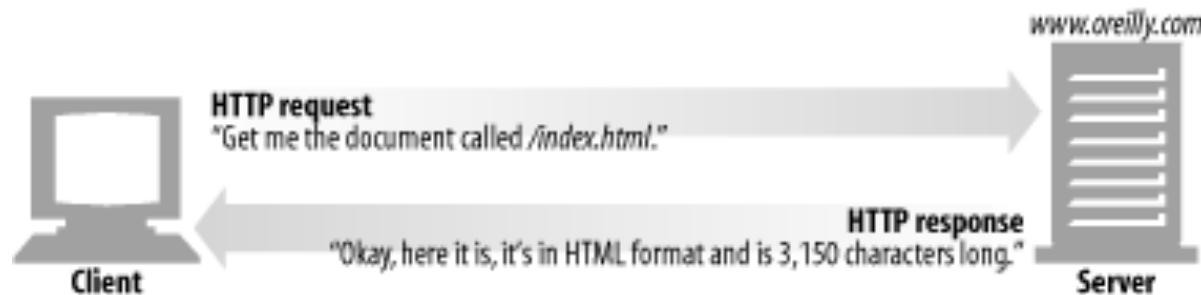
- *Web Server*: Apache, Tomcat, NGINX, MS Internet Information Server, lighttpd, ...
 - Cài đặt tại máy chủ, cung cấp một dịch vụ Web.
 - Lắng nghe các yêu cầu về Web trên một cổng (80).
 - Xử lý các yêu cầu.
 - Tạo kết quả và trả về cho trình duyệt Web.

NỘI DUNG

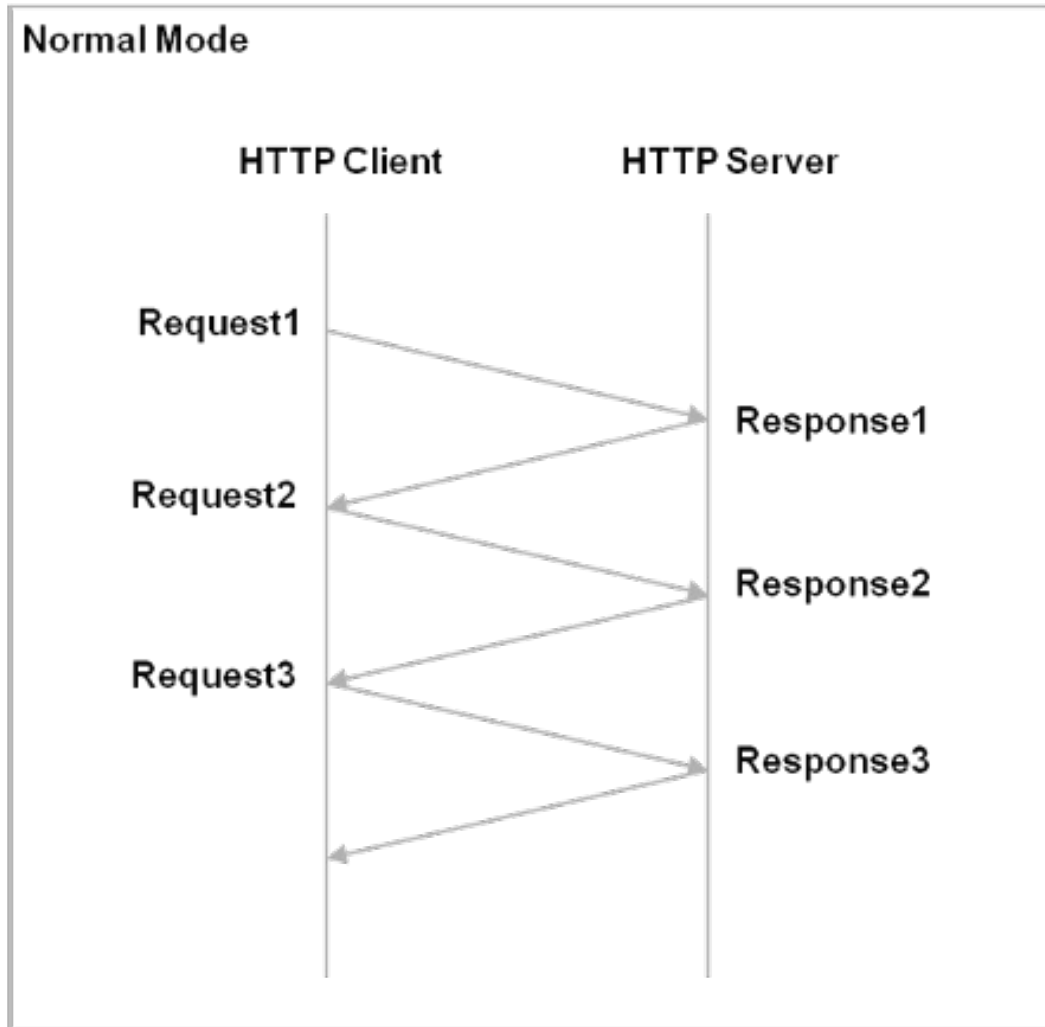
1. Tổng quan
2. Các thành phần của dịch vụ web
- 3. Giao thức HTTP**
4. URL
5. HyperText & HyperLink
6. Web Cache

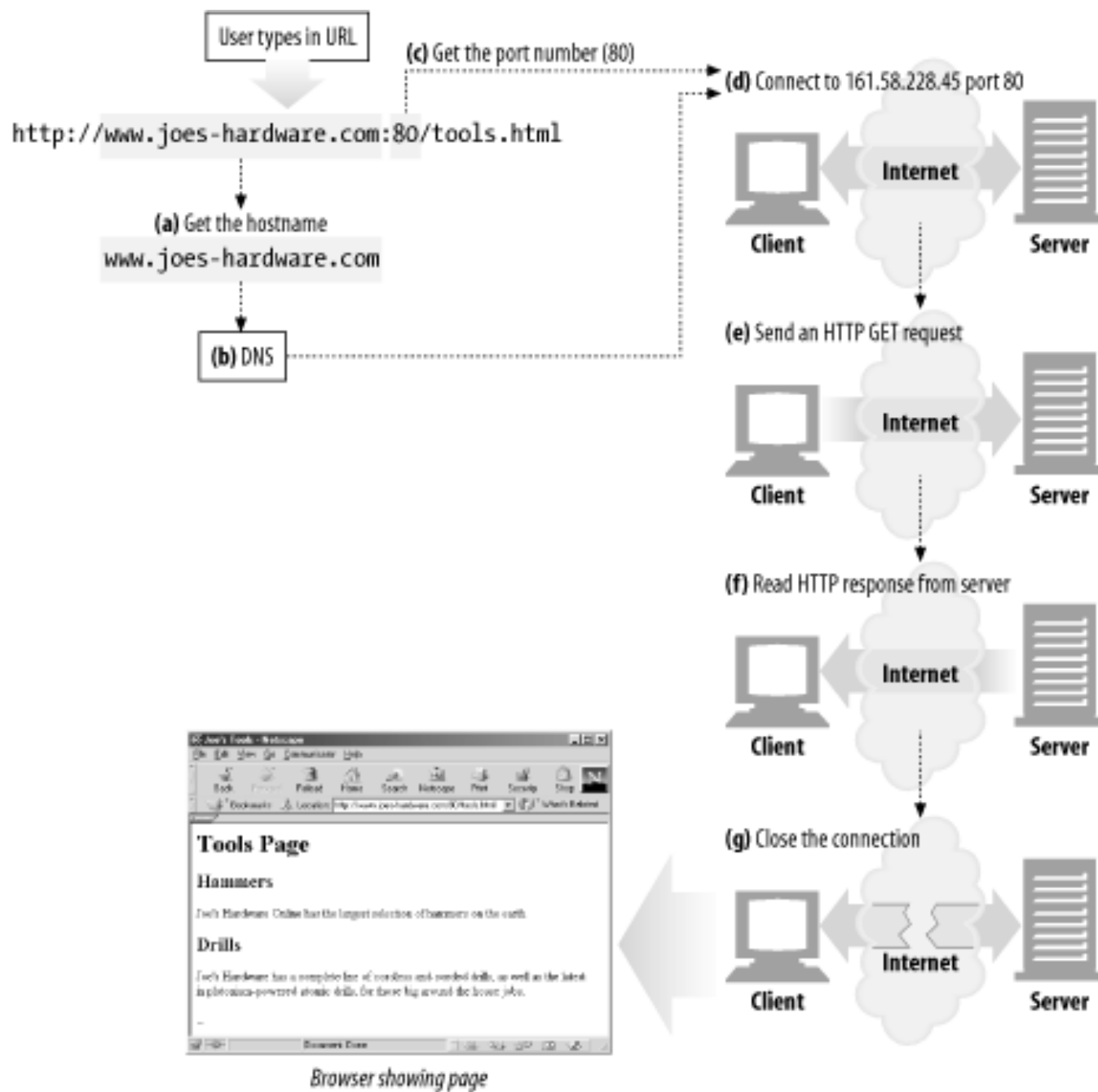
3. GIAO THỨC HTTP

- **HyperText Transfer Protocol**
- Dùng để giao tiếp giữa Web Browser và Web Server
- Giao thức ở tầng ứng dụng trong mô hình OSI, hoạt động trên nền giao thức TCP/IP
- Có 2 kiểu thông điệp: request (webbrowser), response (webserver).
- HTTP Server hoạt động mặc định trên cổng 80
- Là giao thức “không trạng thái” (*stateless*)
- Có thể dùng để truyền tải bất kỳ kiểu dữ liệu nào
- Các phiên bản : HTTP/0.9, HTTP/1.0, HTTP/1.1, HTTP/2, HTTP/3



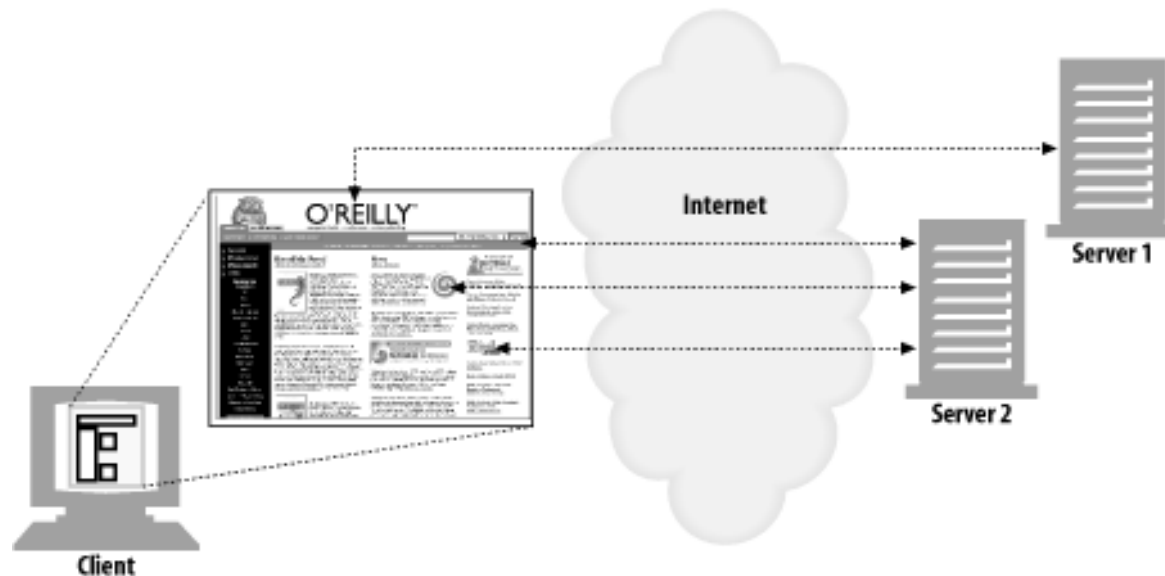
Mô hình hoạt động

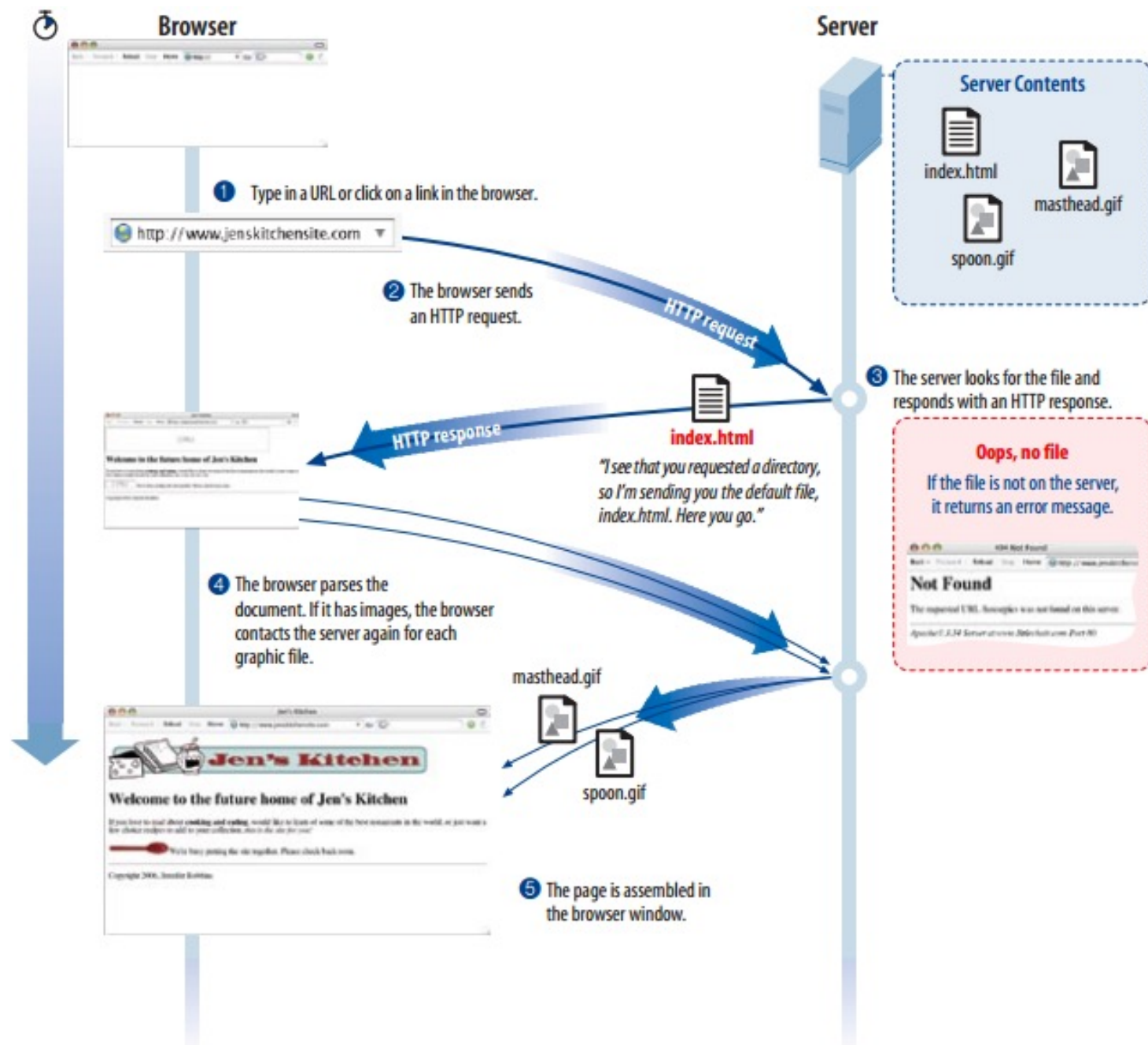




Mô hình hoạt động (tt)

- Các trang web thường chứa nhiều hơn một đối tượng => yêu cầu một trang web thường sinh ra một loạt các thông điệp HTTP cho mỗi đối tượng trên trang web





HTTP Message

- **Có dạng thuần văn bản, gồm:**

- **Request line/Status line (Start line):** là dòng đầu tiên, chứa lệnh yêu cầu của client hay mã trạng thái trả lời của server
- **Header:** có thể không có hoặc có nhiều dòng, mỗi dòng có định dạng như sau:

`<name>: <value>`

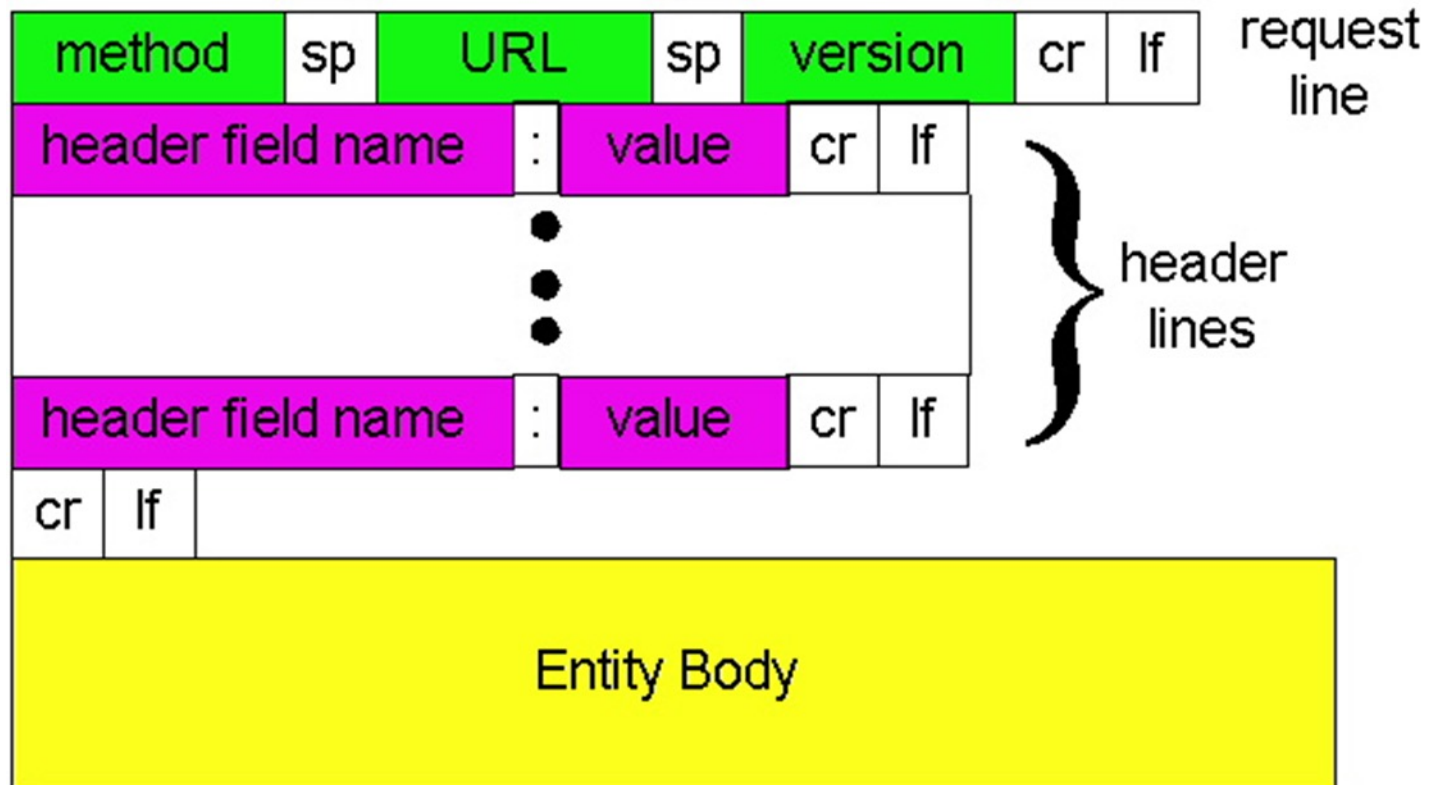
- A blank line: một dòng trống dùng để phân biệt giữa header và body
- **Body:** có thể có hoặc không, có thể chứa bất kỳ loại dữ liệu nào (plain text, binary)

Request line/Status line
Header
A blank line
Body

** Header: có thể tham khảo thêm tại đây:*

http://www.tutorialspoint.com/http/http_header_fields.htm

http request message



http request message (tt)

- **Trong đó:**

- **Method:** là lệnh mà client muốn server thực hiện trên nguồn tài nguyên. Nó là một từ đơn như “GET”, “HEAD” hay, “POST”
- **URL:** địa chỉ nguồn tài nguyên
- **Version:** phiên bản HTTP mà thông điệp sử dụng
- Cr (carriage return) và lf (line feed) là những ký tự trở về đầu dòng và xuống dòng, đánh dấu kết thúc thông điệp

Method	Description
GET	Lấy về tài liệu được xác định trong URL
HEAD	Giống như GET, nhưng trong thông điệp trả về không có body
POST	Cung cấp thông tin cho server
PUT	Tải tài liệu lên server và đặt ở vị trí được xác định trong URL
DELETE	Xóa tài liệu nằm ở vị trí URL trên server

GET vs POST

	GET	POST
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history

GET vs POST

	GET	POST
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

http request message (tt)

request line
(GET, POST,
HEAD commands)

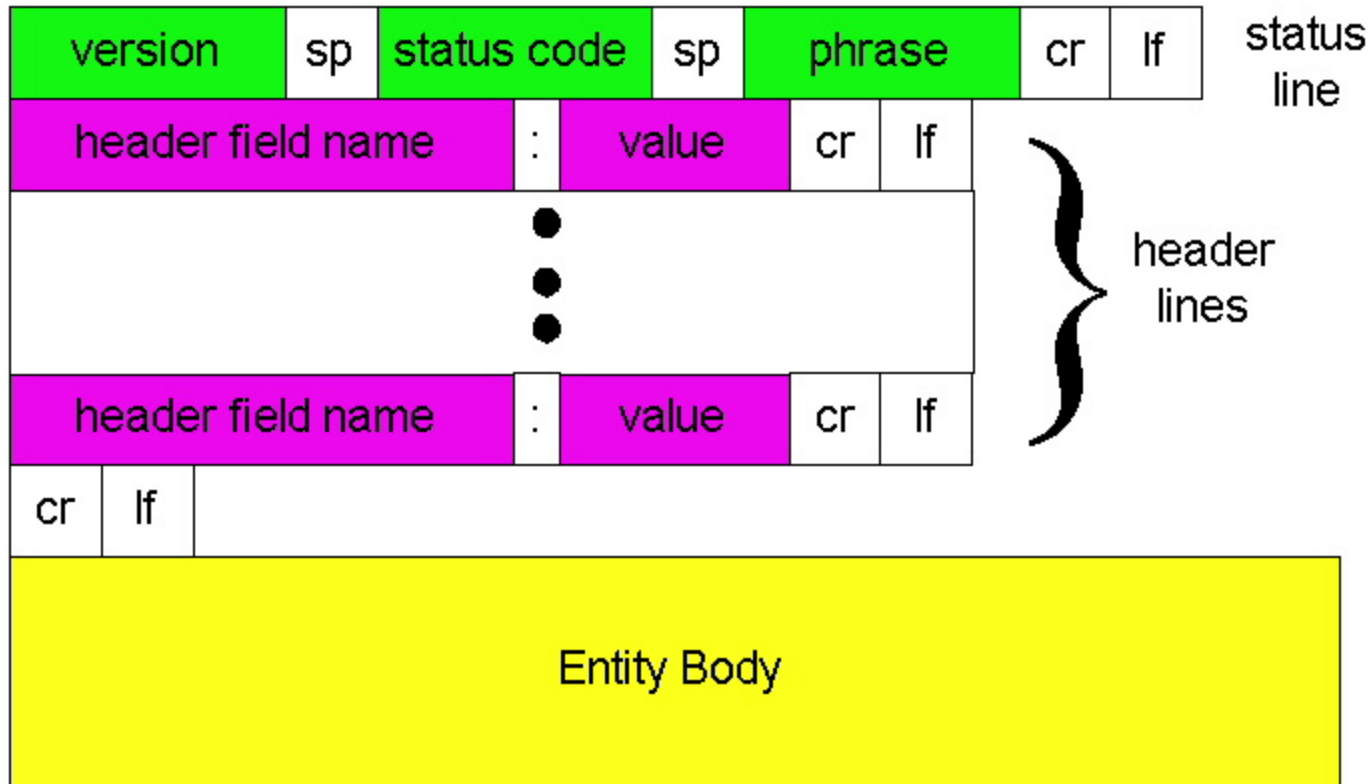
header
lines

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

(extra carriage return, line feed)

Carriage return,
line feed
indicates end
of message

http response message



Status codes

Mã	Loại	Lý do
1xx	Thông tin	Đã nhận được yêu cầu, đang tiếp tục xử lý
2xx	Thành công	Thao tác đã được tiếp nhận, hiểu được và chấp nhận được
3xx	Chuyển hướng	Cần thực hiện thêm thao tác để hoàn tất yêu cầu được đặt ra
4xx	Lỗi client	Yêu cầu có cú pháp sai hoặc không thể được đáp ứng
5xx	Lỗi server	Server thất bại trong việc đáp ứng một yêu cầu hợp lệ

Status codes (tt)

HTTP status code	Description
200	OK. Document returned correctly.
302	Redirect. Go someplace else to get the resource.
404	Not Found. Can't find this resource.

- **More status codes:**

http://www.tutorialspoint.com/http/http_status_codes.htm

http response message

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 ...
```

The diagram illustrates the structure of an HTTP response message. It consists of three main parts: a status line, header lines, and data. The status line is labeled 'status line (protocol status code status phrase)' and points to 'HTTP/1.0 200 OK'. The header lines are labeled 'header lines' and point to a block of five lines: 'Date: Thu, 06 Aug 1998 12:00:15 GMT', 'Server: Apache/1.3.0 (Unix)', 'Last-Modified: Mon, 22 Jun 1998 ...', 'Content-Length: 6821', and 'Content-Type: text/html'. The data is labeled 'data, e.g., requested html file' and points to 'data data data data data ...'.

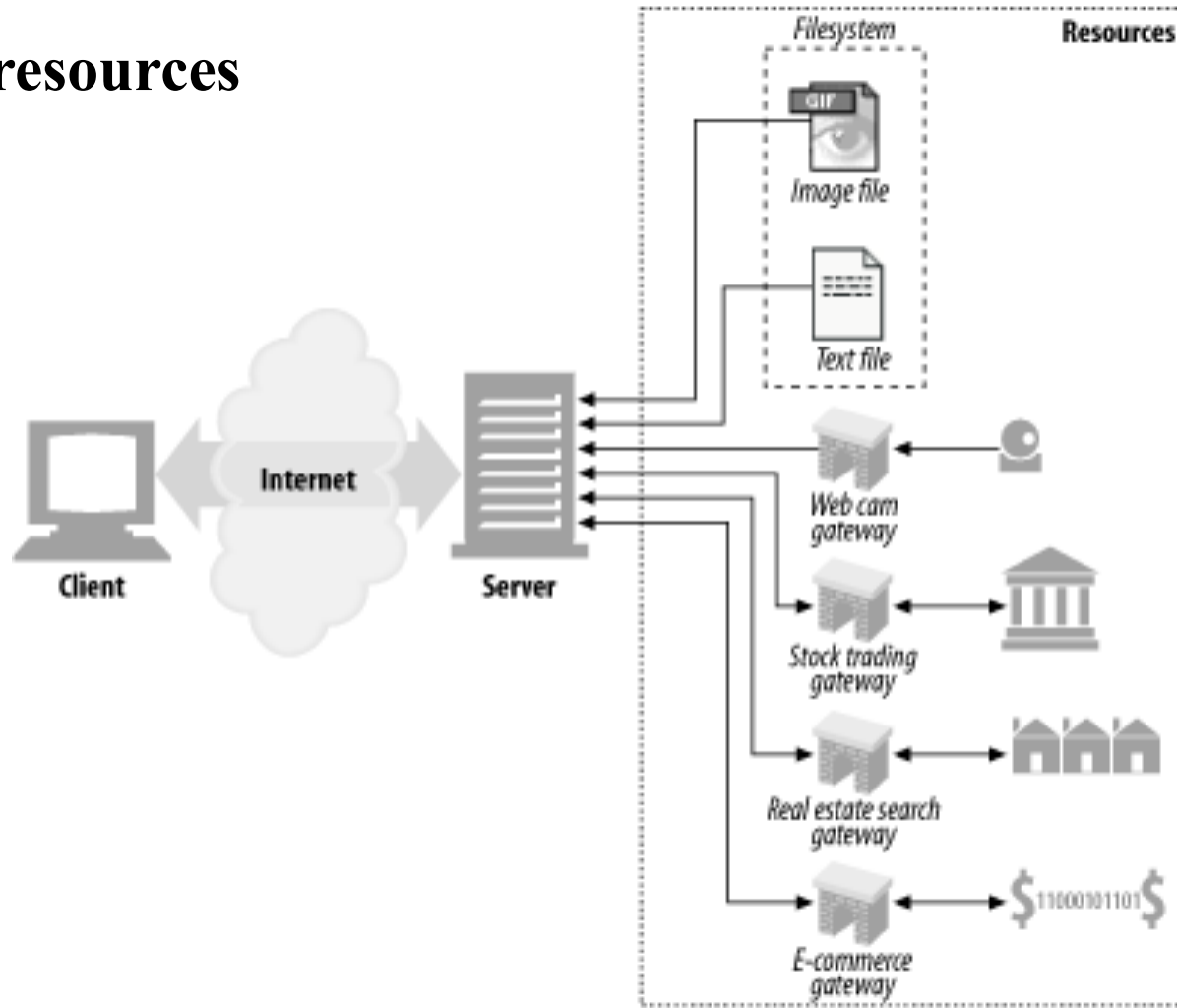


NỘI DUNG

1. Tổng quan
2. Các thành phần của dịch vụ web
3. Giao thức HTTP
- 4. URL**
5. HyperText & HyperLink
6. Web Cache

4. URL

- Web resources



URL

- **URL (Uniform Resource Locator)** hay bộ định vị tài nguyên đồng dạng là địa chỉ dùng để định vị các nguồn tài nguyên trên Internet
- **Cú pháp tổng quát:**
`<protocol>://<user>:<password>@<host>:<port>/<path>
;<params>?<query>`
- **Trong đó:**
 - **protocol:** là giao thức được dùng (http, file, ftp, mailto,...)
 - **host:** là tên máy chủ cung cấp dịch vụ Web, FTP,...
 - **path:** là đường dẫn cục bộ chỉ đến nguồn tài nguyên trên server
 - **params:** gồm các cặp name/value dùng để cung cấp thêm bất kỳ thông tin bổ xung cần thiết để truy cập nguồn tài nguyên
 - **query:** dùng để gửi các tham số đến ứng dụng trên server, có định dạng phổ biến là: `name1=value1[&name2=value2][&...]`

Ví dụ về URL



Một số URL phổ biến

- **HTTP:**

`http://<host>:<port>/<path>?<query>`

Ví dụ:

`http://www.microsoft.com`

`http://www.ctu.edu.vn:8080/cong/home.htm`

`http://www.joes-hardware.com/inventory-check.cgi?item=12731`

- **FTP:**

`ftp://<user>:<password>@<host>:<port>/<path>;<params>`

Ví dụ:

`ftp://ftp.cit.ctu.edu.vn/giaotrinh/`

`ftp://prep.ai.mit.edu/pub/gnu;type=d`

`ftp://ttinternet:ttinternet@172.18.211.19/thuctap/file1.txt`

Một số URL phổ biến (tt)

- **Email:**

`mailto:email_address`

Ví dụ:

`mailto:tttgiang@cit.ctu.edu.vn`

- **File trên đĩa:**

`file://<host>/<path>`

Ví dụ:

`file://internet_server/course/index.htm`

`file:///c:/course/test.html`

URL tuyệt đối & URL tương đối

- **URL tuyệt đối:**

- Là địa chỉ đầy đủ của một tài nguyên.
- Bao gồm giao thức, tên máy chủ, đường dẫn và tên tập tin.
- Ví dụ: *http://www.cit.ctu.edu.vn/student/index.html*

- **URL tương đối:**

- Là một địa chỉ không đầy đủ của một tài nguyên.
- Bao gồm đường dẫn (có thể không có) và tên tập tin.
- Ví dụ 1:

- Người dùng đang đọc trang web:

`http://www.cit.ctu.edu.vn/student/homepage.php`

- Địa chỉ URL tương đối URL= `chuyenmuc.php`

- Phần thông tin bị mất `http://www.cit.ctu.edu.vn/student/`

- Trình duyệt tự xác định URL tuyệt đối:

URL= `http://www.cit.ctu.edu.vn/student/chuyenmuc.php`

URL tuyệt đối & URL tương đối (tt)

- **URL tương đối (tt):**

- Ví dụ 2:

- Người dùng đang đọc trang web:

`http://www.cit.ctu.edu.vn/student/homepage.php`

- Địa chỉ URL tương đối URL= ../index.php

- Phần thông tin bị mất `http://www.cit.ctu.edu.vn/student/`

- Trình duyệt tự xác định URL tuyệt đối:

URL= <http://www.cit.ctu.edu.vn/index.php>

- Ký hiệu:

- "/" thư mục gốc của web server.

- "../" được trình duyệt hiểu như trở về thư mục cấp trên.

URL Encoding

- URLs khi được gửi qua internet các ký tự phải thuộc bộ ký tự ASCII
- URLs có chứa các ký tự bên ngoài bộ ký tự ASCII thì phải được chuyển đổi sang định dạng ASCII trước khi gửi => URL Encoding
- URL Encoding: thay thế các ký tự không thuộc bộ mã ASCII bằng một ký hiệu "%" + hai số thập lục phân theo sau (VD: "*" => %2A)
- URLs không thể chứa khoảng trắng => thay thế bằng "+" hay %20
- VD: `http://www.example.com/new%20pricing.html`
- *URL Encoding Reference:*
http://w3schools.com/tags/ref_urlencode.asp

NỘI DUNG

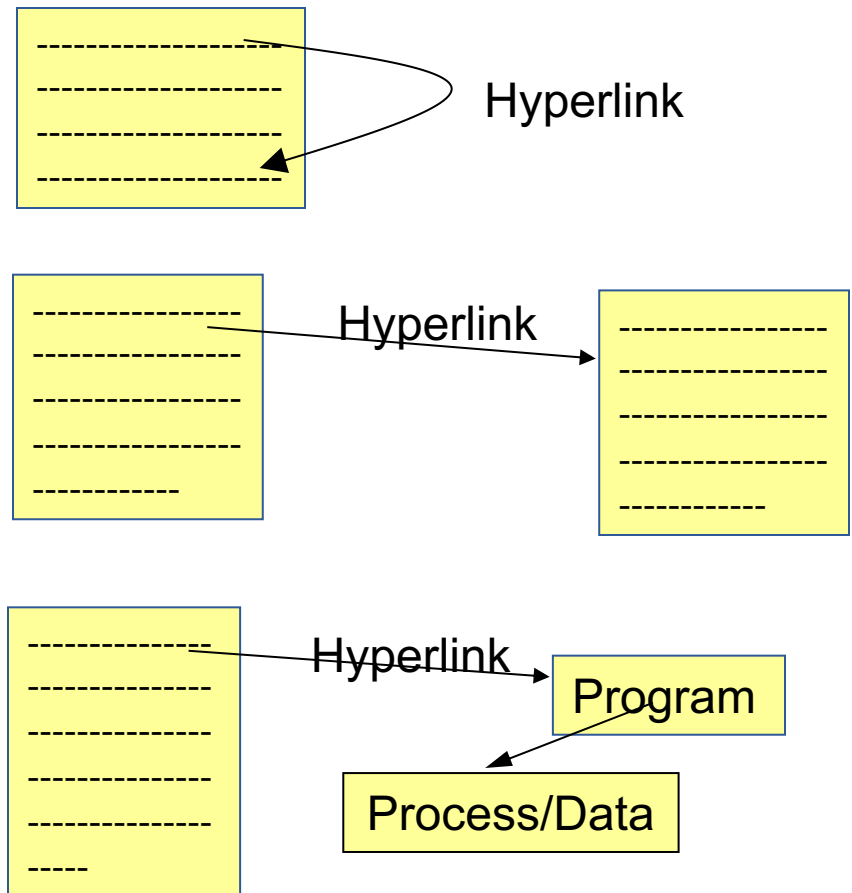
1. Tổng quan
2. Các thành phần của dịch vụ web
3. Giao thức HTTP
4. URL
- 5. HyperText & HyperLink**
6. Web Cache

5. HYPERLINK & HYPERTEXT

- **HyperText:** là hệ thống liên kết các phần tử thông tin nhờ vào các liên kết bằng văn bản có thể kích hoạt hay còn gọi là các siêu liên kết.
- **HyperLink (Siêu liên kết):** Là mối nối kết giữa phần tử thông tin này với phần tử thông tin khác. Phần tử thông tin có thể là:
 - Văn bản, siêu văn bản, website
 - Âm thanh, hình ảnh
 - Tập tin, các đối tượng ActiveX (Word, Excel,...)
 - Những chương trình có thể thực thi viết bằng các ngôn ngữ như Java, Java Applet, ASP, ASP.NET, PHP,...

Các loại siêu liên kết

- **Liên kết trong:** liên kết trong một tài liệu chỉ đến một phần tử thông tin ngay trong chính tài liệu đó.
- **Liên kết ngoài:** liên kết đến một tài liệu khác bên ngoài tài liệu đang tra cứu.
- **Liên kết có thể thực thi được:** liên kết ngoài, thực thi một chương trình xử lý dữ liệu theo yêu cầu người dùng Web, và cho ra thông tin kết quả.



NỘI DUNG

1. Tổng quan
2. Các thành phần của dịch vụ web
3. Giao thức HTTP
4. URL
5. HyperText & HyperLink
- 6. Web Cache**

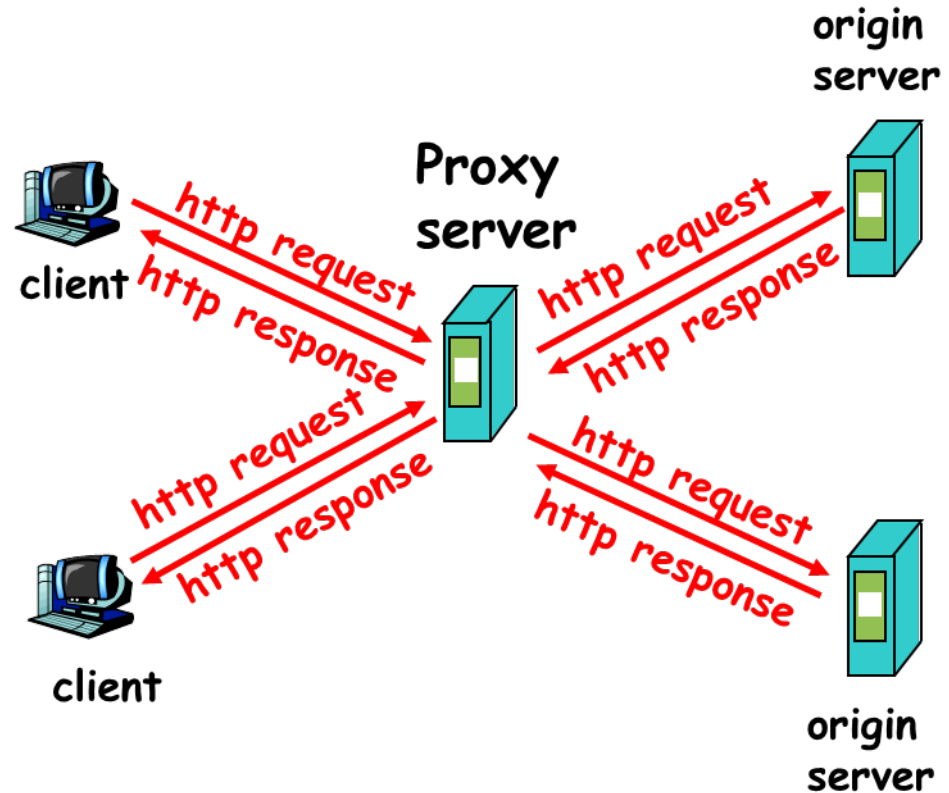
6. WEB CACHE

- **Web cache là một cơ chế lưu trữ tạm bản sao của các tài nguyên web (các trang html, hình ảnh, ...) nhằm đáp ứng cho client mà không cần truy xuất đến server gốc.**
- **Lợi ích của web cache:**
 - Giảm thời gian đáp ứng cho client vì không cần truy xuất đến server gốc.
 - Giảm tải cho web server.
 - Giảm lưu thông trên mạng, tiết kiệm băng thông => hạn chế tình trạng nghẽn (bottleneck) đường truyền.
- **Có 2 loại web cache chính:**
 - Private cache (Browser cache), Public cache (Proxy cache).

Hoạt động của web cache

- **Tất cả yêu cầu http của client được gửi đến web cache**

- Nếu đối tượng yêu cầu đã được lưu trữ tại web cache, web cache sẽ lập tức đáp ứng cho client
- Ngược lại, web cache sẽ truy xuất đến server gốc, nhận đáp ứng, chuyển đến client, đồng thời cũng lưu 1 bản sao trên web cache



Ngăn chặn web cache

- **HTML**

- `<meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidate">`

- **PHP**

- `header("Cache-Control: no-store, no-cache, must-revalidate, max-age=0");`

Question ?