

KHOA CÔNG NGHỆ THÔNG TIN

Môn học: Nhập môn mạng máy tính

Giảng viên: Lê Viết Thanh

Mã số: 7480102

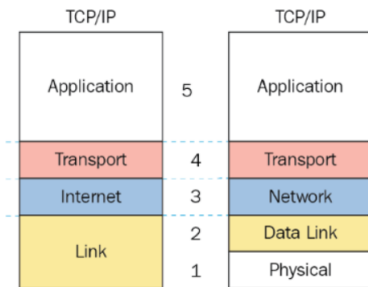
Ngày 17 tháng 9 năm 2023

NỘI DUNG TRÌNH BÀY

- 1 Tổng quan về tầng ứng dụng
- 2 Tên miền và dịch vụ DNS
- 3 HTTP và WWW
- 4 Ứng dụng truyền tệp

Kiến trúc phân tầng

- Cung cấp các dịch vụ trên mạng.
- Trong mô hình TCP/IP không có 2 tầng trình diễn và tầng phiên, nhưng các giao thức tầng ứng dụng phải cung cấp các chức năng của 2 tầng này (biểu diễn dữ liệu, điều khiển phiên...)

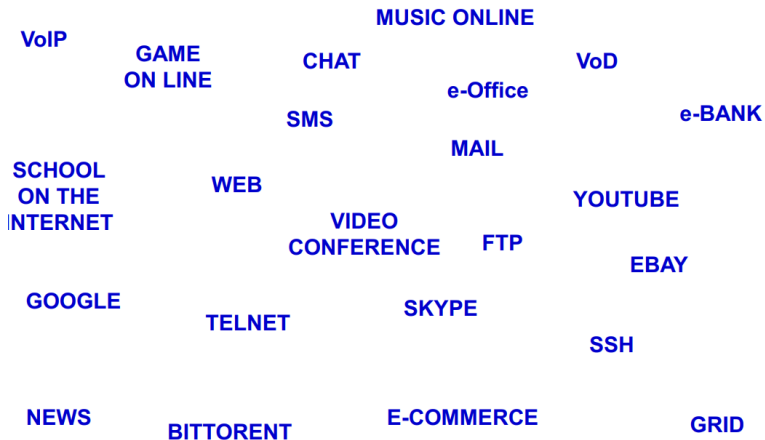


Hình: Mô hình TCP/IP

Link reference TCP:

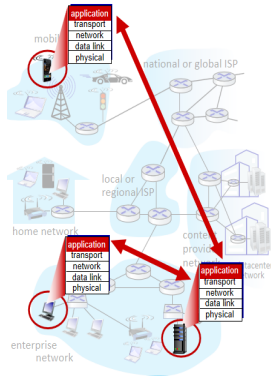
<https://www.youtube.com/watch?v=2QGgEk20RXM>

Ứng dụng và dịch vụ trên mạng?



Hình: Các ứng dụng phổ biến trên mạng

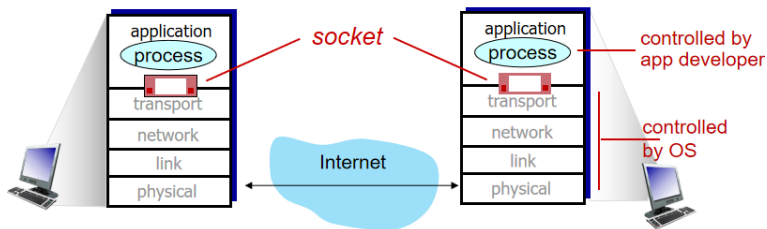
Ứng dụng mạng



- Hoạt động trên các hệ thống đầu cuối (end system)
- Cài đặt giao thức ứng dụng để cung cấp dịch vụ
- Gồm có 2 tiến trình giao tiếp với nhau qua môi trường mạng
 - ▶ **Client:** cung cấp giao diện user, gửi thông điệp yêu cầu dịch vụ
 - ▶ **Server:** cung cấp dịch vụ, trả thông điệp đáp ứng
- **Ví dụ: Web**
 - ▶ Web browser (trình duyệt Web): Chrome, Firefox. . .
 - ▶ Web server: Apache, Tomcat. . .

Giao tiếp giữa các tiến trình ứng dụng

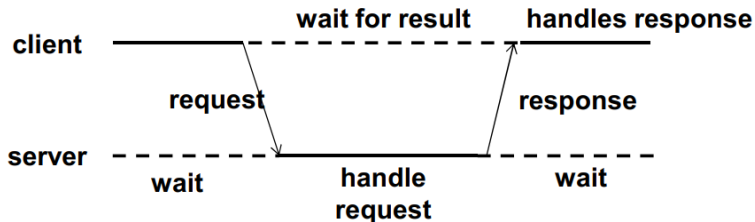
- Socket: điểm truy cập dịch vụ của tầng giao vận
- Các tiến trình ứng dụng sử dụng socket gọi dịch vụ của tầng giao vận để trao đổi thông điệp
 - ▶ Định danh cho tiến trình bởi: Địa chỉ IP, Số hiệu cổng
- Ví dụ: tiến trình web server trên máy chủ của SolICT có định danh 202.191.56.65:80



Hình: Mô hình giao tiếp giữa các tuyến trình

Giao tiếp giữa các tiến trình

- Tiến trình client: gửi yêu cầu
- Tiến trình server: trả lời
- Mô hình điển hình: 1 server – nhiều client
- Client cần biết địa chỉ của server: địa chỉ IP, số hiệu cổng



Hình: Mô hình process

Các mô hình ứng dụng

Khách-chủ (Client/Server)

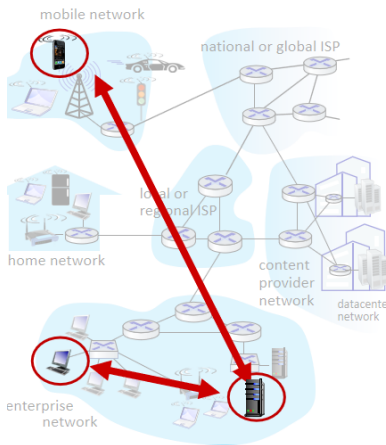
● Khách

- ▶ Khách
- ▶ Gửi yêu cầu truy cập dịch vụ đến máy chủ
- ▶ Về nguyên tắc, không liên lạc trực tiếp với các máy khách khác

● Chủ

- ▶ Thường xuyên online để chờ y/c đến từ máy trạm
- ▶ Có thể có máy chủ dự phòng để nâng cao hiệu năng, phòng sự cố

● Ví dụ: Web server, Mail

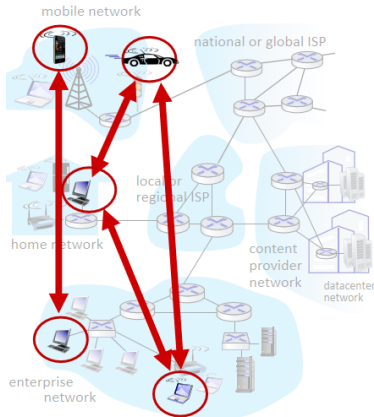


Hình: Mô hình Client-Server

Các mô hình ứng dụng

Mô hình ngang hàng thuần túy (P2P: Peer to Peer)

- Không có máy chủ trung tâm
- Các máy có vai trò ngang nhau
- Hai máy bất kỳ có thể liên lạc trực tiếp với nhau
- Không cần vào mạng thường xuyên

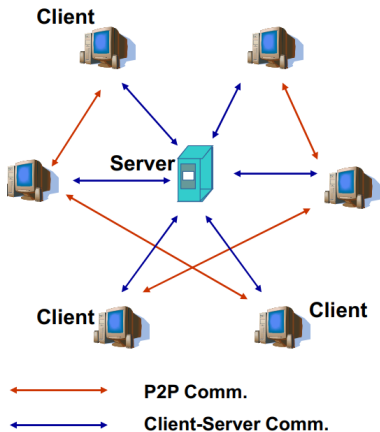


Hình: Mô hình Peer to Peer

Các mô hình ứng dụng

Mô hình lai

- Một máy chủ trung tâm để quản lý user, thông tin tìm kiếm...
- Các máy khách sẽ giao tiếp trực tiếp với nhau sau khi đăng nhập
- E.g. Skype
 - ▶ Máy chủ Skype quản lý các phiên đăng nhập, mật khẩu...
 - ▶ Sau khi kết nối, các máy sẽ gọi VoIP trực tiếp cho nhau



Hình: Mô hình lai

Giới thiệu chung

- Tên miền: định danh trên tầng ứng dụng cho các nút mạng
 - ▶ Trên Internet được quản lý tập trung
 - ▶ Quốc tế: ICANN (Internet Corporation for Assigned Names and Numbers))
 - ▶ Việt Nam: VNNIC(Việt Nam Network Information Center)
- DNS(Domain Name System): hệ thống tên miền
 - ▶ Không gian thông tin tên miền
 - ▶ Gồm các máy chủ quản lý thông tin tên miền và cung cấp dịch vụ DNS
- Vấn đề phân giải tên miền sang địa chỉ IP
 - ▶ Người sử dụng dùng tên miền để truy cập dịch vụ
 - ▶ Máy tính và các thiết bị mạng không sử dụng tên miền mà dùng địa chỉ IP khi trao đổi dữ liệu
- Làm thế nào để chuyển đổi tên miền sang địa chỉ IP?

Chuyển đổi địa chỉ và ví dụ



Hình: Domain Name System



Hình: Mô hình chuyển đổi

Quy tắc đặt tên miền

- Chỉ sử dụng các ký tự chữ cái, số và dấu gạch ngang (hyphen): Tên miền có thể chứa các ký tự a-z, 0-9 và dấu gạch ngang "-". Nhưng không được bắt đầu hoặc kết thúc bằng dấu gạch ngang và không được chứa hai dấu gạch ngang liên tiếp.
- Không sử dụng các ký tự đặc biệt: Các ký tự đặc biệt như *, %, \$, @, v.v. không được phép trong tên miền.
- Không sử dụng dấu cách: Dấu cách không được phép trong tên miền.
- Kích thước tên miền: Tên miền phải có ít nhất 2 ký tự và không quá 63 ký tự.
- Không phân biệt chữ hoa/chữ thường: Tên miền không phân biệt chữ hoa và chữ thường, nghĩa là "example.com" và "EXAMPLE.com" được coi là cùng một tên miền.
- Không vi phạm quy tắc cấp cao hơn: Tên miền phải tuân thủ các quy tắc và hướng dẫn của cơ quan quản lý tên miền quốc gia hoặc tổ chức quản lý tên miền.

Quy tắc đặt tên miền

- Không vi phạm quy tắc bản quyền và thương hiệu: Tên miền không được vi phạm bản quyền hoặc quyền thương hiệu của các tổ chức hoặc cá nhân khác.
- Ví dụ
 - ▶ Hợp lệ:
 - ★ example.com
 - ★ mywebsite.net
 - ★ best-website.org
 - ★ 123website.com
 - ★ abc123.net.vn
 - ▶ Không hợp lệ:
 - ★ ex@ample.com (sử dụng ký tự đặc biệt "@")
 - ★ my website.net (chứa dấu cách)
 - ★ my-website-.org (dấu gạch ngang ở cuối)
 - ★ website_ _with_ _underscores.com (sử dụng gạch dưới "_")

Hệ thống DNS

- Không gian tên miền

- Kiến trúc : hình cây

- ▶ Root

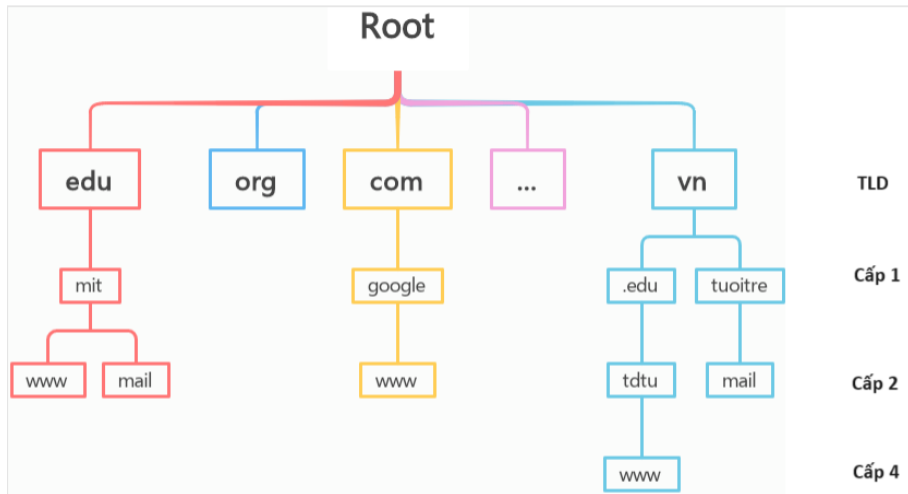
- ★ "Root" (gốc) là mức cao nhất trong cấu trúc của DNS. Nó không thể được đặt tên miền cụ thể và đại diện cho một điểm bắt đầu tuyệt đối trong cấu trúc tên miền.
- ★ Root được biểu thị bằng một dấu chấm (.) đơn giản. Ví dụ: ".".
- ★ Tại mức root, bạn sẽ tìm thấy các máy chủ gốc (root servers) mà các máy chủ DNS khắp thế giới sử dụng để thực hiện các yêu cầu DNS mà họ không thể giải quyết một cách cục bộ. Root servers chứa thông tin về tên miền cấp cao nhất (top-level domains - TLDs) như .com, .org, .net, .gov, .vn, v.v.

- ▶ Zone

- ★ Một "Zone" là một phần của không gian tên miền mà một tổ chức, nhóm hoặc cá nhân có quyền quản lý hoặc quản lý tài nguyên DNS cho nó.
- ★ Mỗi khu vực chứa thông tin về tên miền cụ thể, bao gồm các bản ghi DNS như A, CNAME, MX, v.v.
- ★ Ví dụ: Khu vực "example.com" sẽ chứa các bản ghi DNS liên quan đến tên miền "example.com" như A records, MX records, v.v.

Hệ thống DNS

Kiến trúc cây tên miền



Hình: Kiến trúc cây tên miền

Hệ thống DNS

Phân cấp của DNS (Domain Name System) là cách tổ chức và quản lý các tên miền trên Internet thành các cấp khác nhau, từ tên miền gốc cho đến các tên miền cấp cao hơn, và sau đó các tên miền cấp thấp hơn.

- Tên Miền Gốc (Root Domain):
 - ▶ Tên miền gốc được ký hiệu bằng một dấu chấm duy nhất (.) và nó không thể được chia nhỏ hơn.
 - ▶ Ví dụ: "."
- Tên Miền Cấp 1 (Top-Level Domain - TLD):
 - ▶ Tên miền cấp 1 nằm ngay dưới tên miền gốc và thường đại diện cho các quốc gia hoặc các loại tổ chức.
 - ▶ Ví dụ:
 - ★ Quốc gia: "com" (thương mại), "org" (tổ chức phi lợi nhuận), "gov" (cơ quan chính phủ), "edu" (giáo dục), "vn" (Việt Nam), "uk" (Vương quốc Anh), v.v.
 - ★ Loại tổ chức: "net" (mạng), "mil" (quân đội), v.v.
- Tên Miền Cấp 2 (Second-Level Domain - SLD):
 - ▶ Tên miền cấp 2 nằm dưới tên miền cấp 1 và thường đại diện cho tổ chức hoặc cá nhân cụ thể.
 - ▶ Ví dụ: "google.com", "microsoft.com", "example.org", "wikipedia.org"

Hệ thống DNS

- Tên Miền Cấp 3, Cấp 4, ... (Subdomains):
 - ▶ Các tên miền cấp dưới cấp 2 (subdomains) được tạo ra để chia nhỏ không gian tên miền của một tổ chức hoặc mạng.
 - ▶ Ví dụ:
 - ★ "www.google.com" (subdomain của "google.com").
 - ★ "mail.microsoft.com" (subdomain của "microsoft.com").
- Authoritative DNS Server:
 - ▶ Một máy chủ DNS chính thức (Authoritative DNS Server) là nơi lưu trữ và quản lý các bản ghi DNS của một tên miền cụ thể. Khi một máy chủ DNS được gọi là "Authoritative" cho một tên miền, nó có thẩm quyền cuối cùng trong việc cung cấp thông tin về tên miền đó.
 - ▶ Ví dụ: Một máy chủ DNS của nhà cung cấp dịch vụ hosting web có thể được coi là "Authoritative" cho các tên miền mà nó quản lý. Khi có yêu cầu về tên miền đó, máy chủ DNS này có quyền cung cấp thông tin chính thức.

Link reference:<https://www.youtube.com/watch?v=mpQZVYPuDGU>

Hệ thống DNS

- Mỗi nút là một tập hợp các bản ghi mô tả tên miền tương ứng với nút đó.
 - ▶ SOA (Start of Authority):
 - ★ Thuộc tính SOA chứa thông tin về máy chủ DNS có quyền tổ chức cho tên miền cụ thể.
 - ★ Nó bao gồm thông tin như tên miền gốc, địa chỉ email của quản trị viên, số ngày giữa các cập nhật, thời gian sống (TTL), v.v.
 - ★ Ví dụ: ns1.example.com. hostmaster.example.com.
 - ▶ NS (Name Server):
 - ★ Thuộc tính NS xác định máy chủ DNS mà tên miền cụ thể sử dụng để xác định địa chỉ IP của các máy chủ hoặc tài nguyên liên quan đến tên miền đó.
 - ★ Ví dụ: ns1.example.com, ns2.example.com.
 - ▶ A (Address Record):
 - ★ Thuộc tính A ánh xạ một địa chỉ IP (IPv4) tới một tên miền. Điều này cho phép khi bạn nhập tên miền vào trình duyệt, hệ thống biết được địa chỉ IP cụ thể của máy chủ đó.
 - ★ www.example.com, A 192.168.1.1

Hệ thống DNS

- PTR (Pointer Record):

- ▶ Thuộc tính PTR thường được sử dụng trong quá trình giải quyết ngược (reverse DNS lookup). Nó ánh xạ địa chỉ IP tới tên miền.
- ▶ Ví dụ: 1.1.168.192.in-addr.arpa. PTR www.example.com.

- CNAME (Canonical Name):

- ▶ Thuộc tính CNAME tạo ra một liên kết giữa một tên miền và một tên miền khác, cho phép một tên miền có thể được sử dụng như một biệt danh (alias) của tên miền khác.
- ▶ Ví dụ: www.example.com. IN CNAME example.com.

- MX (Mail Exchange):

- ▶ Thuộc tính MX xác định các máy chủ email mà tên miền cụ thể sử dụng để nhận thư điện tử.
- ▶ Ví dụ: example.com. IN MX 10 mail.example.com.

- TXT (Text):

- ▶ Thuộc tính TXT cho phép người quản trị tên miền đưa ra các thông tin mô tả hoặc hướng dẫn khác nhau, thường được sử dụng để cung cấp thông tin phụ trợ về tên miền.
- ▶ Ví dụ: example.com. IN TXT "This is a sample TXT record."

Phân giải tên miền

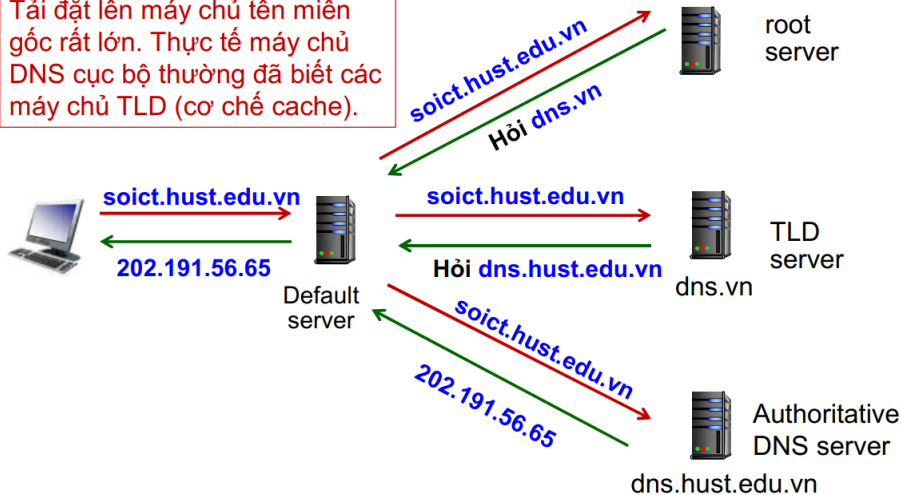
Phân giải tên miền thường được thực hiện bởi các máy chủ DNS. Khi bạn nhập một tên miền vào trình duyệt web, máy tính của bạn sẽ gửi một yêu cầu DNS đến máy chủ DNS. Máy chủ DNS sau đó sẽ trả về địa chỉ IP tương ứng cho tên miền đó, cho phép máy tính kết nối đến máy chủ cụ thể.

- UDP, Port 53.
- Local DNS Resolution (Phân giải DNS cục bộ): Máy tính đầu tiên kiểm tra xem liệu nó đã lưu trữ địa chỉ IP cho tên miền đó trong bộ nhớ đệm của nó. Nếu có, nó sẽ sử dụng địa chỉ IP này mà không cần gửi yêu cầu đến máy chủ DNS ngoài.
- Recursive DNS Resolution (Phân giải DNS đệ quy): Nếu địa chỉ IP không có trong bộ nhớ đệm hoặc nếu bộ nhớ đệm đã hết hạn, máy tính sẽ gửi yêu cầu đến máy chủ DNS cục bộ hoặc máy chủ DNS được cấu hình trong cài đặt mạng của nó. Nếu máy chủ DNS cục bộ không biết địa chỉ IP cho tên miền đó, nó sẽ gửi yêu cầu đến các máy chủ DNS khác cho đến khi nó nhận được đáp ứng.

Phân giải tương tác

Cơ chế mặc định trên các máy chủ DNS

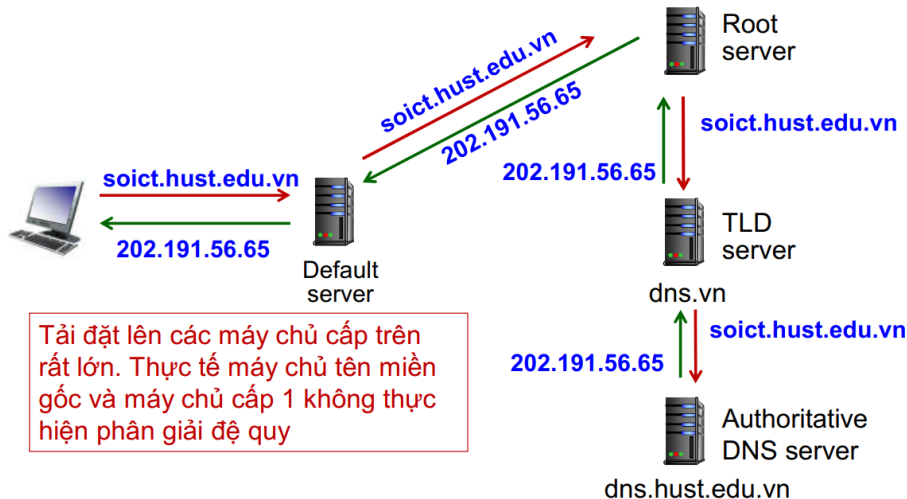
Tải đặt lên máy chủ tên miền gốc rất lớn. Thực tế máy chủ DNS cục bộ thường đã biết các máy chủ TLD (cơ chế cache).



Hình: Mô hình phân giải tương tác

Phân giải đệ quy

Cơ chế mặc định trên các máy chủ DNS



Hình: Mô hình phân giải tương tác

Thông điệp DNS

- DNS Query và DNS Reply
 - ▶ Chung khuôn dạng
- QUESTION: tên miền cần truy vấn
 - ▶ Số lượng: #Question
- ANSWER: thông tin tên miền tìm kiếm được
 - ▶ Số lượng: #Answer RRs
- AUTHORITY: địa chỉ server trả lời truy vấn
- ADDITIONAL: thông tin phân giải tên miền cho các địa chỉ khác

SRC = 53	DST = 53
checksum	length
Identification	Flags
#Question	#Answer RRs
#Authority RRs	#Additional RRs
QUESTION	
ANSWER	
AUTHORITY	
ADDITIONAL	

Hình: Khung dạng DNS

- **Internet trước thập kỷ 1990s:**

- ▶ Hầu như chỉ sử dụng hạn chế trong cơ quan chính phủ, phòng nghiên cứu...
- ▶ Các dịch vụ email, FPT không phù hợp cho chia sẻ thông tin đại chúng
- ▶ Không có cơ chế hiệu quả để liên kết các tài nguyên thông tin nằm rải rác trên Internet

- **Năm 1990, Tim Berners-Lee giới thiệu World Wide Web:**

- ▶ Trao đổi thông tin dưới dạng siêu văn bản (hypertext) sử dụng ngôn ngữ HTML (Hypertext Markup Language)
- ▶ Các đối tượng không cần đóng gói “tất cả trong một” như trên các văn bản trước đó
- ▶ Siêu văn bản chỉ chứa chứa liên kết (hypertext) tới các đối tượng khác (định vị bằng URL)

Uniform Resource Locator

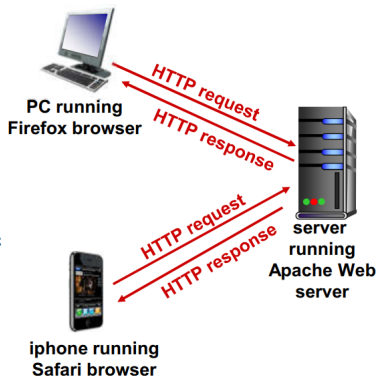
Uniform Resource Locator (URL) là một chuỗi ký tự định dạng cụ thể được sử dụng để chỉ định vị trí của một tài nguyên trên Internet. URL bao gồm các thành phần sau:

- **Giao thức (Protocol):** Phần này chỉ định giao thức mà máy tính sẽ sử dụng để truy cập tài nguyên. Ví dụ: "http", "https", "ftp", "mailto", v.v.
- **Tên miền (Domain Name):** Đây là địa chỉ của máy chủ hoặc nơi tài nguyên đó được lưu trữ. Ví dụ: "www.example.com".
- **Đường dẫn (Path):** Đây là đường dẫn cụ thể đến tài nguyên trên máy chủ. Ví dụ: "/path/to/resource".
- **Tham số (Parameters):** Đây là các thông tin bổ sung được truyền theo yêu cầu. Ví dụ: "?param1=value1¶m2=value2".
- **Fragment Identifier:** Đây là một phần nhỏ chỉ định vị trí cụ thể của tài nguyên. Nó thường được sử dụng trong các tài liệu HTML để trở đến một vị trí cụ thể trên trang. Ví dụ:
<https://www.tuoitre.vn/path/to/resource?param1=value1¶m2=value2#>

HTTP và WWW

Uniform Resource Locator (URL) là một chuỗi ký tự định dạng cụ thể được sử dụng để chỉ định vị trí của một tài nguyên trên Internet. URL bao gồm các thành phần sau:

- WWW: World Wide Web
 - ▶ Trao đổi dữ liệu siêu văn bản HTML (HyperText Markup Language) trên mạng
- HTTP: HyperText Transfer Protocol
 - ▶ Mô hình Client/Server
 - ▶ Client yêu cầu truy nhập tới các trang web (chứa các đối tượng web) và hiển thị chúng trên trình duyệt
 - ▶ Server: Nhận yêu cầu và trả lời cho client



Hình: Mô hình HTTP

Hoạt động của giao thức HTTP

● Hoạt động của giao thức HTTP

- ▶ Server mở một TCP socket chờ yêu cầu kết nối tại cổng 80 (default)
- ▶ Client khởi tạo một liên kết TCP tới server
- ▶ Server chấp nhận yêu cầu, tạo liên kết
- ▶ Trao đổi thông điệp HTTP (giao thức ứng dụng)
 - ★ HTTP Request
 - ★ HTTP Response
- ▶ Đóng liên kết TCP

● **Khuôn dạng HTTP request** Là cách mà thông điệp được gửi từ một client đến một server trên Internet. Nó chứa các thông tin cần thiết để yêu cầu một tài nguyên cụ thể từ server.

- ▶ Method (Phương thức) Xác định loại yêu cầu mà client muốn thực hiện. Các phương thức phổ biến bao gồm:
 - ★ GET: Yêu cầu lấy dữ liệu từ server.
 - ★ POST: Yêu cầu gửi dữ liệu đến server tạo mới tài nguyên Không yêu cầu xác định trước ID và Không idempotent.
 - ★ PUT: Yêu cầu cập nhật tài nguyên trên server hoặc tạo mới yêu cầu ID và dempotent .
 - ★ DELETE: Yêu cầu xóa tài nguyên trên server, và nhiều phương thức khác.

Khuôn dạng HTTP request

- URL (Uniform Resource Locator):
 - ▶ Xác định địa chỉ của tài nguyên mà client muốn truy cập.
- HTTP Version (Phiên bản HTTP):
 - ▶ Xác định phiên bản của giao thức HTTP đang được sử dụng (ví dụ: HTTP/1.1).
- Headers (Tiêu đề):
 - ▶ Chứa các thông tin bổ sung về yêu cầu như user-agent, cookies, và các thông số khác.
- Body (Nội dung):
 - ▶ Chứa dữ liệu gửi kèm theo yêu cầu (thường được sử dụng trong các phương thức như POST).
- Ví dụ: Ví dụ về một HTTP request cơ bản sử dụng phương thức GET

GET /index.html HTTP/1.1

Host: www.example.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.82

Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
image/avif,image/webp,*/*;q=0.8

Ví dụ khuôn dạng HTTP request

Trong ví dụ trên:

- Phương thức là GET.
- URL là `"/index.html"`.
- Phiên bản HTTP là HTTP/1.1.
- Tiêu đề bao gồm Host, User-Agent, và Accept.
- Ví dụ: Tạo một người dùng mới trên máy chủ:

POST /api/users HTTP/1.1

Host: example.com

Content-Type: application/json

`"username": "john_doe",`

`"email": "john@example.com"`

Các phương thức trong thông điệp yêu cầu

• HTTP/1.0

- ▶ GET
- ▶ POST
- ▶ HEAD

★ Yêu cầu máy chủ loại một số đối tượng ra khỏi thông điệp trả lời

• HTTP/1.1

- ▶ GET
- ▶ POST
- ▶ HEAD
- ▶ PUT

★ Tải file lên máy chủ, đường dẫn chỉ ra trong URL, file để trong body

- ▶ DELETE

★ Xóa file chỉ ra bởi đường dẫn

• Ví

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

Mã trạng thái trả lời

Trong dòng đầu tiên của thông điệp trả lời, ví dụ

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

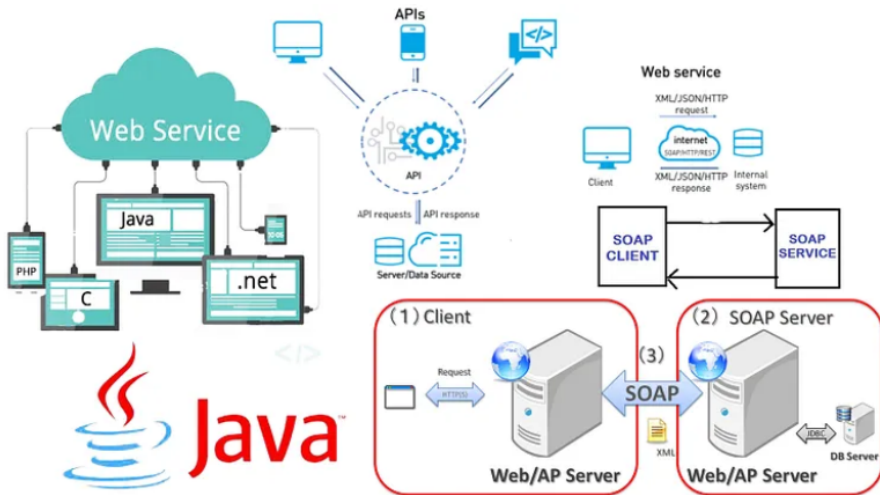
- ▶ request message not understood by server

404 Not Found

- ▶ requested document not found on this server 505 HTTP Version Not Supported

Link reference:<https://www.youtube.com/watch?v=hExRDVZHhig>

Kiến trúc chung của các dịch vụ web



Hình: Dịch vụ web

Hiển thị nội dung trang web

- Mô hình xử lý cơ bản tại trình duyệt: mỗi cửa sổ hoặc 1 frame:
 - ▶ Nhận thông điệp HTTP Response
 - ★ Hiển thị:
 - ★ Xử lý mã HTML, CSS, Javascripts
 - ★ Gửi thông điệp HTTP Request yêu cầu các đối tượng khác(nếu có)
 - ★ Bắt và xử lý sự kiện
- Các sự kiện có thể xảy ra:
 - ▶ Sự kiện của người dùng: OnClick, OnMouseOver...
 - ▶ Sự kiện khi hiển thị: OnLoad, OnBeforeUnload...
 - ▶ Theo thời gian: setTimeout(), clearTimeout()...

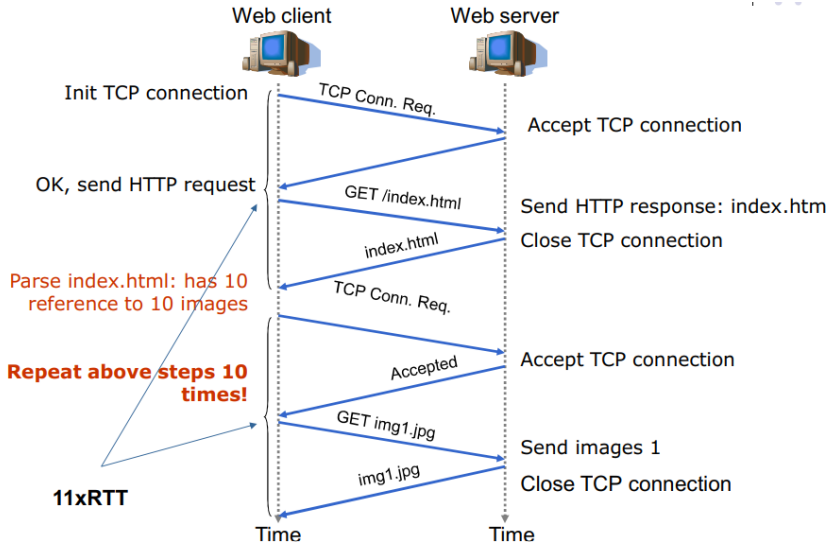
Các liên kết HTTP HTTP không duy trì

- Chỉ một đối tượng web được gửi qua liên kết TCP
- Sử dụng mặc định trong HTTP/1.0
- HTTP 1.0: RFC(Request for Comments) 1945

HTTP có duy trì

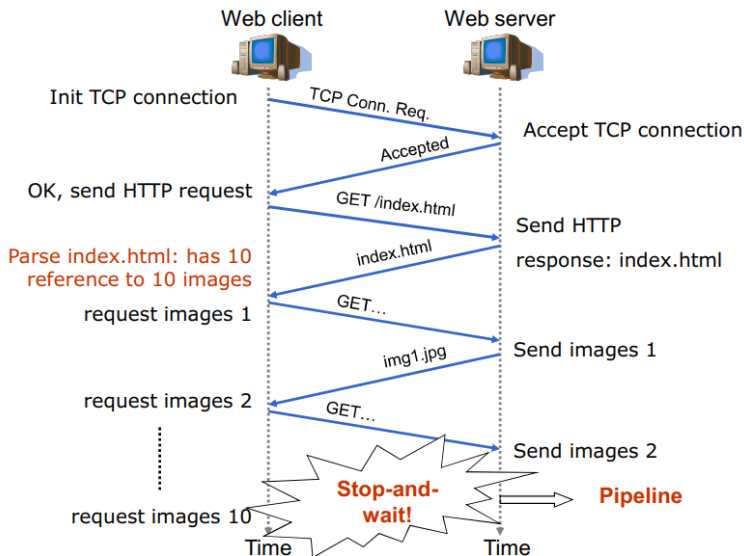
- Nhiều đối tượng có thể được gửi qua một liên kết TCP.
- Sử dụng mặc định trong HTTP/1.1
- HTTP 1.1: RFC 2068

Hoạt động của HTTP/1.0



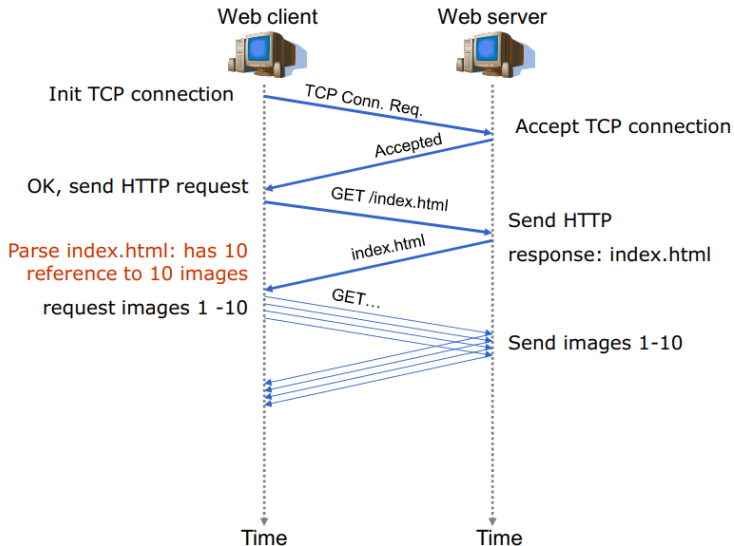
Hình: Hoạt động của HTTP/1.0

Hoạt động của HTTP/1.1



Hình: Hoạt động của HTTP/1.1

HTTP/1.1 với pipeline



Hình: Hoạt động Pipeline

HTTP là giao thức stateless

Giao thức HTTP (Hypertext Transfer Protocol) được gọi là "stateless" (không lưu trạng thái) có nghĩa là mỗi yêu cầu HTTP được gửi từ máy khách (trình duyệt) tới máy chủ không lưu trữ thông tin về trạng thái trước đó của các yêu cầu. Điều này có nghĩa là mỗi yêu cầu được xử lý độc lập và không có thông tin về các yêu cầu trước đó.

- Một phiên hoạt động của HTTP:
 - ▶ Trình duyệt kết nối với Web server
 - ▶ Trình duyệt gửi thông điệp yêu cầu HTTP Request
 - ▶ Web server đáp ứng với một thông điệp HTTP Response
 - ▶ ...lặp lại...
 - ▶ Trình duyệt ngắt kết nối
- Các thông điệp HTTP Request được xử lý độc lập
- Web server không ghi nhớ trạng thái của phiên HTTP
 - ▶ Nếu dịch vụ Web cần xác thực người dùng thì người dùng sẽ phải đăng nhập lại cho mỗi thông điệp HTTP Request gửi đi

HTTP Cookie

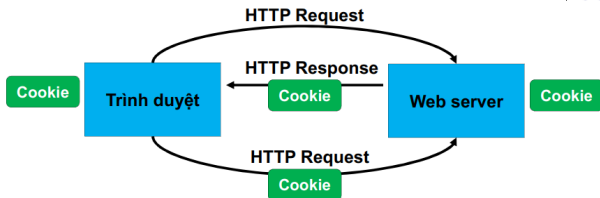
HTTP Cookie (hay đơn giản là Cookie) là một mẫu thông tin mà máy chủ web có thể lưu trữ trên máy tính của người dùng và sau đó sử dụng thông tin đó khi người dùng truy cập lại trang web. Cookie được sử dụng để theo dõi và duy trì trạng thái của người dùng trong suốt các phiên truy cập liên tiếp vào trang web.

- Cookie: dữ liệu do ứng dụng Web tạo ra, chứa thông tin trạng thái của phiên làm việc
 - ▶ Server có thể lưu lại cookie (một phần hoặc toàn bộ)
- Sau khi xử lý yêu cầu, Web server trả lại thông điệp HTTP Response với cookie đính kèm
 - ▶ Set-Cookie: key = value; options;
- Trình duyệt lưu cookie
- Trình duyệt gửi HTTP Request tiếp theo với cookie được đính kèm

Ví dụ: user_id=12345; session_token=abcdef

Cookie này có thể được gửi từ máy chủ đến trình duyệt của người dùng và sau đó trình duyệt sẽ gửi lại cookie này mỗi khi người dùng truy cập trang web đó.

HTTP Cookie



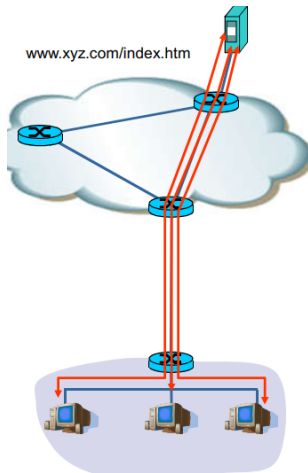
Hình: HTTP Cookie

Các Cookie thường được sử dụng để:

- Bảo trì trạng thái người dùng: Để nhớ thông tin như đăng nhập, giỏ hàng mua hàng, v.v.
- Theo dõi hoạt động người dùng: Để phân tích hành vi trực tuyến và tùy chỉnh trải nghiệm người dùng.
- Cung cấp quảng cáo có mục tiêu: Cho phép quảng cáo được tùy chỉnh dựa trên sở thích cá nhân.

Bộ đệm- Caching

- “Cache”: Bộ nhớ đệm
- Khái niệm bộ nhớ cache trong máy tính
 - ▶ L1 cache, L2 cache
 - ▶ “cache miss”, “cache hit”
- Xem xét trường hợp sau:
 - ▶ Một tổ chức có một đường nối tới Internet
 - ▶ Tất cả lưu lượng truy cập web đều đi qua liên kết này
 - ▶ Nhiều users web có thể cùng truy nhập tới cùng một nội dung
 - ▶ Giải pháp cải tiến?



Hình: Caching

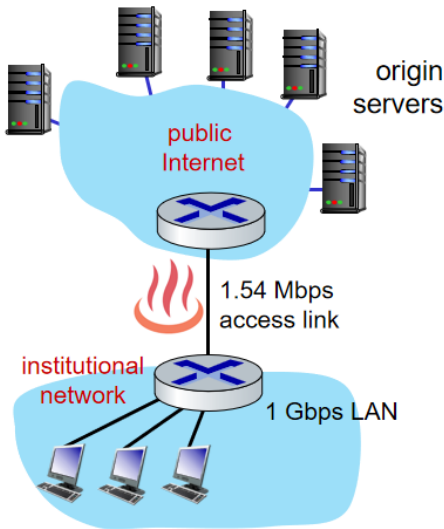
Caching example

1 Scenario:

- ▶ access link rate: 1.54 Mbps
- ▶ RTT from institutional router to server: 2 sec
- ▶ web object size: 100K bits
- ▶ average request rate from browsers to origin servers: 15/sec
 - ★ avg data rate to browsers: 1.50 Mbps

2 Performance:

- ▶ access link utilization = .97
- ▶ LAN utilization: .0015
- ▶ end-end delay = Internet delay + access link delay + LAN delay = 2 sec + minutes + usecs



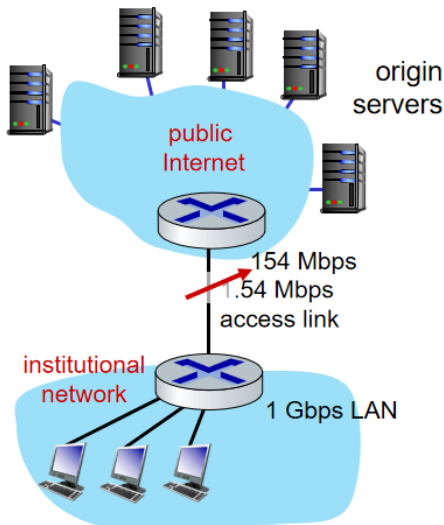
Option 1: buy a faster access link

1 Scenario:

- ▶ access link rate: 1.54 → **154 Mbps**
- ▶ RTT from institutional router to server: 2 sec
- ▶ web object size: 100K bits
- ▶ average request rate from browsers to origin servers: 15/sec
 - ★ avg data rate to browsers: 1.50 Mbps

2 Performance:

- ▶ access link utilization = .97
→ **.0097**
- ▶ LAN utilization: .0015
- ▶ end-end delay = Internet delay + access link delay + LAN delay = 2 sec + minutes → **msecs** + usecs



Hình: Example Caching

Option 2: install a web cache

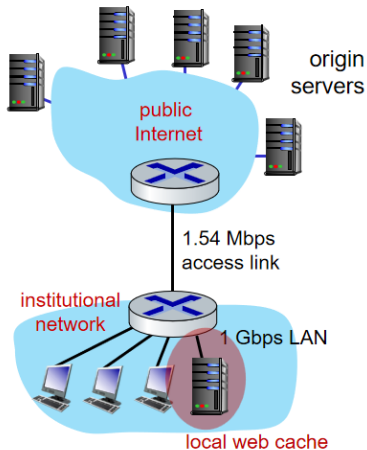
1 Scenario:

- ▶ access link rate: 1.54 Mbps
- ▶ RTT from institutional router to server: 2 sec
- ▶ web object size: 100K bits
- ▶ average request rate from browsers to origin servers: 15/sec
 - ★ avg data rate to browsers: 1.50 Mbps

2 Cost: web cache (cheap!)

3 Performance:

- ▶ LAN utilization: .?
- ▶ access link utilization = ?
- ▶ average end-end delay = ?



Hình: Web Cache

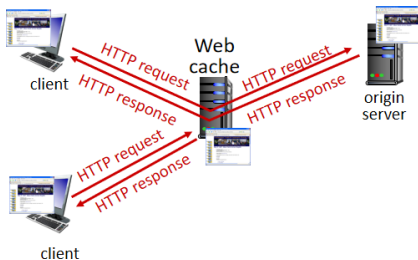
How to compute link utilization, delay?

- Tỷ lệ sử dụng mạng LAN:
 - ▶ Tỷ lệ sử dụng mạng LAN (LAN utilization) là tỷ lệ thời gian mạng LAN thực sự được sử dụng so với thời gian mà nó có thể được sử dụng.
 - ▶ $\text{LAN utilization} = (\text{Tốc độ dữ liệu đến trình duyệt}) / (\text{Tốc độ truy vấn trung bình từ trình duyệt đến máy chủ gốc}) = 1.50 \text{ Mbps} / 15 \text{ truy vấn/giây} = 0.10 \text{ Mbps/truy vấn/giây}$
- Tỷ lệ sử dụng liên kết truy cập:
 - ▶ Tỷ lệ sử dụng liên kết truy cập (Access link utilization) là tỷ lệ thời gian liên kết truy cập thực sự được sử dụng so với thời gian mà nó có thể được sử dụng.
 - ▶ $\text{Access link utilization} = (\text{Tốc độ dữ liệu đến trình duyệt}) / (\text{Access link rate}) = 1.50 \text{ Mbps} / 1.54 \text{ Mbps} = 0.974$
- Độ trễ trung bình từ nguồn đến đích:
 - ▶ Độ trễ trung bình (Average end-to-end delay) bao gồm độ trễ trong quá trình truyền dữ liệu từ nguồn đến đích, bao gồm cả RTT và thời gian truyền dữ liệu.
 - ▶ Thời gian truyền dữ liệu = Kích thước đối tượng / Tốc độ dữ liệu đến trình duyệt = $(100\text{K bits}) / (1.50 \text{ Mbps}) = 0.53 \text{ giây}$
 - ▶ Độ trễ trung bình = RTT + Thời gian truyền dữ liệu = $2 \text{ giây} + 0.53 \text{ giây} = 2.53 \text{ giây}$

Sử dụng bộ đệm - web proxy

Web proxy là một máy chủ đứng giữa máy người dùng và máy chủ mục tiêu trên Internet. Chức năng chính của web proxy là đại diện cho người dùng khi truy cập các tài nguyên trên Internet.

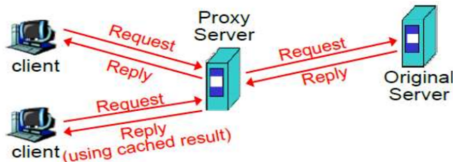
- User đặt tham số kết nối truy cập web của trình duyệt qua một máy chủ proxy
- Trình duyệt gửi yêu cầu đến proxy web cho trình duyệt
 - ▶ Miss: Proxy gửi yêu cầu tới máy chủ web, trả lời trình duyệt và lưu đệm đối tượng web
 - ▶ Hit: Proxy trả đối tượng



Hình: Caching

Web proxy

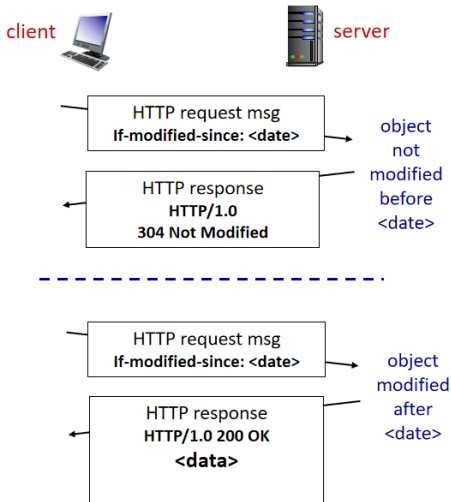
- Proxy: Vừa là client, vừa là server
- Sử dụng bởi các ISP nhỏ, các tổ chức như trường học, công ty...
- Ảnh hưởng của proxy
 - ▶ Làm giảm lưu lượng web trên đường ra Internet
 - ▶ Có thể làm giảm thời gian đáp ứng
 - ▶ Thử phân tích vài trường hợp
 - ★ cache hit
 - ★ cache miss
 - ★ proxy bị quá tải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



Hình: Web Proxy

Conditional GET

- 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



Hình: Điều kiện Get

Hạn chế của HTTP

- Không bảo mật dữ liệu: HTTP truyền dữ liệu dưới dạng văn bản thô, không được mã hóa.
- Không xác thực đáng tin cậy: HTTP không cung cấp các cơ chế xác thực mạnh mẽ. Điều này có nghĩa là không có cách chắc chắn rằng máy chủ mà trình duyệt kết nối đến là máy chủ thực sự của trang web đó.
- Không quản lý trạng thái: HTTP là một giao thức không lưu trạng thái, điều này có nghĩa là mỗi yêu cầu được xử lý độc lập và không có thông tin về các yêu cầu trước đó từ cùng một người dùng.
- Hiệu năng không tối ưu: HTTP không nén dữ liệu trên đường truyền. Điều này có thể dẫn đến việc tiêu tốn nhiều băng thông hơn và tốn thời gian hơn đối với việc tải các trang web lớn.
- Khả năng bị tấn công: HTTP không cung cấp các cơ chế bảo mật mạnh mẽ, dễ dàng trở thành mục tiêu của các cuộc tấn công như tấn công Man-in-the-Middle, tấn công tăng cường và nhiều hình thức tấn công khác.

Hypertext Transfer Protocol Secure (HTTPS)

- Bảo mật dữ liệu: mã hóa dữ liệu truyền tải giữa máy khách và máy chủ. Điều này đảm bảo rằng thông tin nhạy cảm như thông tin đăng nhập, thông tin thẻ tín dụng, và dữ liệu cá nhân không thể bị đánh cắp hoặc nhìn thấy bởi bên thứ ba không có quyền truy cập.
- Xác thực máy chủ: HTTPS sử dụng chứng chỉ SSL/TLS để xác thực máy chủ. Điều này đảm bảo rằng trang web mà người dùng đang truy cập thực sự là trang web mà họ mong muốn, và không phải là một trang web giả mạo.
- Cung cấp uy tín và độ tin cậy: Các trang web sử dụng HTTPS thường được coi là đáng tin cậy hơn và đáng giá hơn đối với người dùng. Điều này đặc biệt quan trọng đối với các trang web chứa thông tin cá nhân và thanh toán trực tuyến.
- Cải thiện SEO(Search Engine Optimization): Các công cụ tìm kiếm như Google ưu tiên các trang web sử dụng HTTPS trong kết quả tìm kiếm của mình. Do đó, việc sử dụng HTTPS có thể cải thiện vị trí của trang web trong kết quả tìm kiếm.

Hypertext Transfer Protocol



Hình: HTTP

Hypertext Transfer Protocol Secure



Hình: HTTPS

HTTPS

- Tuy nhiên, HTTPS có thể gây hiểu nhầm cho người dùng rằng trang web là an toàn:
 - ▶ Người dùng bất cẩn vì chỉ chú ý biểu tượng
 - ▶ HTTPS không chống lại được các dạng tấn công khai thác điểm yếu của Website
- Số hiệu cổng: 443

Electronic Mail (Email)

- MUA (Mail User Agent)

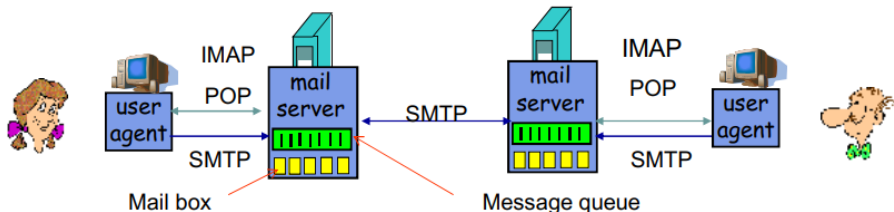
- ▶ Lấy thư từ máy chủ, gửi thư đến máy chủ
- ▶ e.g. Outlook, Thunderbird...

- MTA (Mail Transfer Agent):

- ▶ Chứa hộp thư đến của user (mail box)
- ▶ Hàng đợi để gửi thư đi
- ▶ e.g. Sendmail, MS Exchange

- Giao thức:

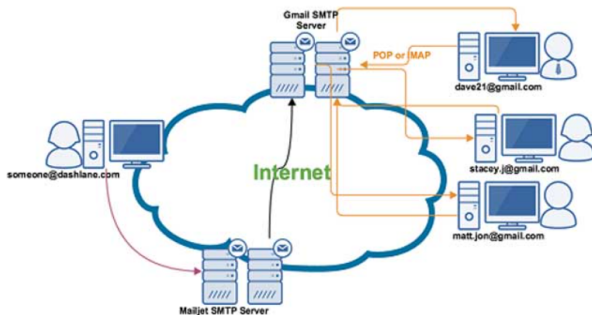
- ▶ Chuyển thư: SMTP-Simple Mail Transfer Protocol
- ▶ nhận thư
- ▶ POP – Post Office Protocol
- ▶ IMAP – Internet Mail Access Protocol



Hình: E-Mail

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



Hình: E-Mail

Các giao thức nhận thư

POP và IMAP là hai giao thức phổ biến được sử dụng để truy cập và quản lý email trên máy chủ email từ các ứng dụng email hoặc trình duyệt web.

- POP (Post Office Protocol):

- ▶ POP là một giao thức truyền tải email từ máy chủ email đến máy tính cá nhân của người dùng.
- ▶ Khi một email được tải về từ máy chủ POP, nó thường bị xóa khỏi máy chủ, nghĩa là email chỉ tồn tại trên thiết bị mà nó đã được tải về.
- ▶ POP thường được sử dụng khi người dùng muốn lưu trữ email trên máy tính cá nhân và không quan trọng việc truy cập email từ nhiều thiết bị.

- IMAP (Internet Message Access Protocol):

- ▶ IMAP cũng là một giao thức truyền tải email từ máy chủ đến thiết bị người dùng.
- ▶ Tuy nhiên, IMAP giữ email trên máy chủ nguyên vẹn và không xóa email khi được tải về.
- ▶ Điều này cho phép người dùng truy cập vào email từ nhiều thiết bị và đồng bộ hóa trạng thái email trên tất cả các thiết bị.

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

Link reference SMTP:

<https://www.youtube.com/watch?v=PJo5yOtu7o8>

Link reference POP and IMAP:

<https://www.youtube.com/watch?v=SBaARws0hy4&t=30s>

Khuôn dạng thông điệp thư điện tử

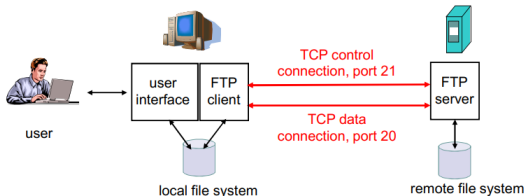
- SMTP: Giao thức để truyền thư
- RFC 822: Định nghĩa khuôn dạng
 - ▶ Phần đầu
 - ★ To:
 - ★ From:
 - ★ Subject:
 - ▶ Phần thân
 - ★ mã hóa dưới dạng mã ASCII

The diagram illustrates the structure of an email. It is divided into two main sections: the header and the body. The header section, titled 'New Message', contains fields for 'To', 'Cc', 'Bcc', and 'Subject'. The body section is the main content area. Below the body section is a toolbar with various icons for editing, including undo, redo, font face (Sans Serif), font size (12), bold (B), italic (I), underline (U), text color (A), background color, bulleted list, numbered list, and a dropdown menu.

Hình: Cấu trúc E-Mail

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 :
 - ▶ TCP control connection: port 21
 - ▶ TCP data connection: port 20
- User phải đăng nhập trước khi truyền file
- Một số server cho phép user với tên là anonymous



Hình: File Transfer Protocol

Link reference FTP: <https://www.youtube.com/watch?v=tOj8MSEIbfA>

Lệnh và mã trả lời

Các lệnh FTP phổ biến:

- USER - Để xác định tên người dùng.
- PASS - Để xác định mật khẩu.
- PWD - Để hiển thị thư mục làm việc hiện tại trên máy chủ.
- CWD - Để thay đổi thư mục làm việc hiện tại.
- LIST - Để liệt kê các tệp tin và thư mục trong thư mục hiện tại.
- RETR - Để tải tệp tin từ máy chủ xuống máy người dùng.
- STOR - Để tải tệp tin từ máy người dùng lên máy chủ.
- DELE - Để xóa một tệp tin trên máy chủ.
- 200 - Lệnh được thực thi thành công. 331 - Yêu cầu xác thực, yêu cầu tên người dùng và mật khẩu.
- 530 - Lỗi xác thực, tên người dùng hoặc mật khẩu không chính xác. 226 - Thành công, dữ liệu đã được truyền tải hoặc thực hiện thành công.
- 550 - Lỗi, tệp tin không tồn tại hoặc quyền truy cập bị từ chối. 425 - Lỗi, không thể mở cổng dữ liệu. 250 - Lệnh thành công, thay đổi thư mục làm việc hiện tại. 257 - Đường dẫn thư mục mới được tạo. 221 - Kết nối được đóng.