**Chương 1: TỔNG QUAN VỀ LẬP TRÌNH MẠNG**

**Contents**

[Các thành phần 2](#_Toc179359504)

[Communication links 2](#_Toc179359505)

[Host 2](#_Toc179359506)

[Router 2](#_Toc179359507)

[Gateway (Proxy - cổng nối) 3](#_Toc179359508)

[Proxy server (Máy chủ proxy) 3](#_Toc179359509)

[Switch (Bộ chuyển mạch) 4](#_Toc179359510)

[Modem 4](#_Toc179359511)

[Firewall (Tường lửa) 4](#_Toc179359512)

[Access Point (Điểm truy cập không dây) 4](#_Toc179359513)

[Network Interface Card (NIC - Card giao tiếp mạng) 4](#_Toc179359514)

[Server (Máy chủ) 5](#_Toc179359515)

[Repeater (Bộ lặp/ bộ chuyển tiếp) 5](#_Toc179359516)

[Hub (bộ tập trung) 5](#_Toc179359517)

[Bridge (Cầu nối mạng) 6](#_Toc179359518)

[Truyền dữ liệu và mạng máy tính 6](#_Toc179359519)

[Truyền dữ liệu (Data communications) 6](#_Toc179359520)

[Kết nối mạng (Networking) 7](#_Toc179359521)

[Mạng LAN 7](#_Toc179359522)

[Mạng MAN 7](#_Toc179359523)

[Mạng WAN 8](#_Toc179359524)

[Các khái niệm cơ bản 8](#_Toc179359525)

[Giao thức (protocol) 8](#_Toc179359526)

[Kiến trúc giao thức 8](#_Toc179359527)

[a) Tầng Vật lý (Physical Layer) 9](#_Toc179359528)

[b) Tầng Liên kết dữ liệu (Data Link Layer) 9](#_Toc179359529)

[c) Tầng Mạng (Network Layer) 10](#_Toc179359530)

[d) Tầng Giao vận (Transport Layer) 11](#_Toc179359531)

[e) Tầng Phiên (Session Layer) 12](#_Toc179359532)

[f) Tầng Trình bày (Presentation Layer) 13](#_Toc179359533)

[g) Tầng Ứng dụng (Application Layer) 13](#_Toc179359534)

[Một số bộ giao thức kết nối mạng & Các cổng phổ biến 14](#_Toc179359535)

[Cổng (Port) 15](#_Toc179359536)

[Interface 16](#_Toc179359537)

[Socket 16](#_Toc179359538)

[Ứng dụng mạng 17](#_Toc179359539)

[Lập trình mạng 17](#_Toc179359540)

[Mô hình Peer-to-Peer và mô hình Client/Server 18](#_Toc179359541)

[Mô hình Client/Server 18](#_Toc179359542)

[WWW - Kết nối đến Web Server 19](#_Toc179359543)

[File Transfer Protocol (FTP) - FTP client và FTP Server 20](#_Toc179359544)

[Email 20](#_Toc179359545)

[Webmail 20](#_Toc179359546)

[Chat 20](#_Toc179359547)

# Các thành phần

## Communication links

Liên kết truyền thông là các phương tiện vật lý hoặc không dây dùng để truyền tín hiệu giữa các thiết bị mạng, như máy tính, router, và các thành phần mạng khác. Các liên kết truyền thông này là con đường để dữ liệu đi từ một thiết bị đến thiết bị khác. Có nhiều loại liên kết truyền thông phổ biến:

* **Liên kết có dây (Wired Communication)**: Sử dụng cáp như cáp Ethernet, cáp đồng trục, hoặc cáp quang. Các liên kết này thường ổn định và ít bị nhiễu hơn.
  + **Cáp đồng (Copper cables)**: Sử dụng dây dẫn bằng đồng để truyền tín hiệu. Ví dụ phổ biến là cáp Ethernet (Cat5, Cat6).
  + **Cáp quang (Optical fiber)**: Sử dụng ánh sáng để truyền tín hiệu, cho phép tốc độ truyền cao và phạm vi xa hơn so với cáp đồng.
* **Liên kết không dây (Wireless Communication)**: Sử dụng sóng radio hoặc vi sóng để truyền tín hiệu. Ví dụ như Wi-Fi, Bluetooth, hoặc các liên kết di động 4G/5G.
  + **Wi-Fi**: Liên kết không dây phổ biến nhất cho các thiết bị mạng trong nhà hoặc văn phòng.
  + **Sóng vệ tinh (Satellite communication)**: Sử dụng sóng radio để liên lạc giữa các thiết bị và vệ tinh.

## Host

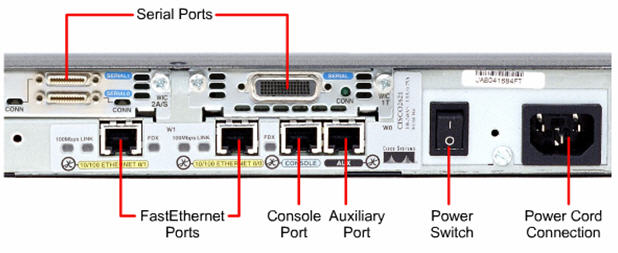
Host là bất kỳ thiết bị nào trên mạng có khả năng truyền và nhận dữ liệu, như máy tính, máy chủ, điện thoại di động, hoặc máy in. Một host có thể là người khởi xướng yêu cầu truyền dữ liệu (client) hoặc cung cấp dịch vụ dữ liệu (server). Host có địa chỉ IP (Internet Protocol) riêng để định danh và giao tiếp trong mạng.

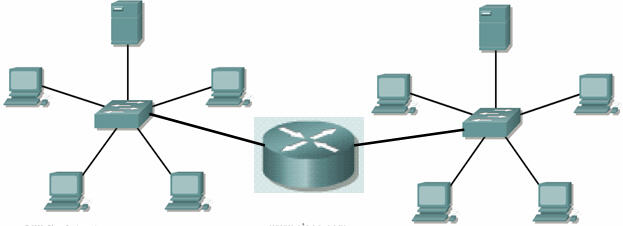
## Router

* Dùng để ghép nối các mạng cục bộ lại với nhau thành mạng rộng.
* Hoạt động chủ yếu ở lớp Network.
* Có 2 phương thức định tuyến chính:
  + Định tuyến tĩnh: cấu hình các đường cố định và cài đặt các đường đi này vào bảng định tuyến.
  + Định tuyến động:
    - Vectơ khoảng cách: RIP, IGRP, EIGRP, BGP
    - Trạng thái đường liên kết: OSPF

Router là thiết bị mạng có nhiệm vụ định tuyến dữ liệu giữa các mạng khác nhau, dựa trên địa chỉ IP. Router giúp xác định con đường tốt nhất để gói tin (data packet) đi từ nguồn đến đích, qua nhiều mạng trung gian. Router có khả năng:

* **Chuyển tiếp dữ liệu** giữa các mạng khác nhau, giúp dữ liệu đi từ mạng nguồn đến mạng đích.
* **Xác định đường đi tối ưu**: Sử dụng các giao thức định tuyến (routing protocols) như OSPF, BGP để tìm ra con đường nhanh nhất và hiệu quả nhất cho gói tin.
* **Kết nối các mạng LAN với Internet**: Trong mạng gia đình hoặc văn phòng, router thường kết nối mạng cục bộ (LAN) với Internet, cho phép các thiết bị trong mạng cục bộ truy cập Internet.





## Gateway (Proxy - cổng nối)

* Thường dùng để kết nối các mạng không thuần nhất, chủ yếu là mạng LAN với mạng lớn bên ngoài chứ **không dùng kết nối LAN – LAN.**
* Kiểm soát luồng dữ liệu ra vào mạng.
* Hoạt động phức tạp và chậm hơn Router.
* Hoạt động từ tầng thứ 4🡪7

Gateway là một thiết bị hoặc một điểm kết nối giữa hai mạng khác nhau, thường với các kiến trúc hoặc giao thức khác nhau. Một gateway hoạt động như một "cầu nối" giữa hai môi trường mạng không tương thích, chẳng hạn giữa mạng IP và mạng không sử dụng IP, hoặc giữa mạng công ty và Internet.

* **Chức năng dịch giao thức**: Gateway có thể chuyển đổi dữ liệu từ giao thức này sang giao thức khác để đảm bảo việc giao tiếp giữa các hệ thống không tương thích.
* **Ví dụ về Gateway**: Một router có thể hoạt động như một gateway khi nó kết nối mạng cục bộ (LAN) với mạng diện rộng (WAN) hoặc Internet. Gateway cũng thường được sử dụng trong các hệ thống VoIP để chuyển đổi giữa tín hiệu thoại số và thoại tương tự.

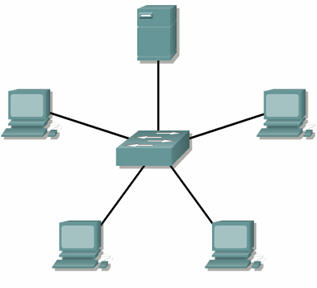
## Proxy server (Máy chủ proxy)

Proxy server là một máy chủ trung gian đứng giữa người dùng và Internet. Nó thực hiện việc gửi các yêu cầu của người dùng đến Internet và sau đó trả kết quả về cho người dùng. Một số chức năng chính của proxy server bao gồm:

* **Cải thiện bảo mật**: Ẩn địa chỉ IP của người dùng và bảo vệ danh tính trực tuyến.
* **Lọc nội dung**: Kiểm soát và chặn các trang web hoặc dịch vụ không phù hợp.
* **Tăng tốc độ truy cập**: Bằng cách lưu trữ các trang web đã truy cập trước đó (caching), proxy có thể trả về kết quả nhanh hơn trong những lần truy cập sau.

## Switch (Bộ chuyển mạch)

* Có khả năng chỉ chuyển dữ liệu đến đúng kết nối thực sự cần dữ liệu này làm giảm đụng độ trên mạng.
* Switch là thiết bị mạng dùng để kết nối nhiều thiết bị trong một mạng cục bộ (LAN) với nhau, chẳng hạn như máy tính, máy in, hoặc các thiết bị mạng khác.
* Switch hoạt động ở tầng liên kết dữ liệu (Layer 2 của mô hình OSI) và đôi khi cũng có thể hoạt động ở tầng mạng (Layer 3). Chức năng của switch bao gồm:
* **Chuyển tiếp dữ liệu trong mạng LAN**: Switch nhận các khung dữ liệu (data frame) từ các thiết bị trong mạng và chuyển tiếp chúng đến đúng đích (một thiết bị khác trong cùng mạng).
* **Tăng hiệu suất mạng**: Switch tạo các đường truyền dữ liệu riêng biệt giữa các thiết bị, giảm tắc nghẽn mạng bằng cách chỉ gửi dữ liệu đến thiết bị cần thiết thay vì phát dữ liệu cho tất cả các thiết bị.
* **VLAN (Virtual LAN)**: Switch có khả năng tạo ra các VLAN để chia mạng cục bộ thành các phân đoạn nhỏ hơn nhằm tăng cường bảo mật và quản lý lưu lượng mạng.



## Modem

Là tên viết tắt của hai từ điều chế (MOdulation) và giải điều chế (DEModulation).

Modem là thiết bị dùng để kết nối mạng cục bộ với mạng diện rộng (WAN), thường là Internet. Modem chuyển đổi tín hiệu số từ các thiết bị mạng thành tín hiệu tương tự (đối với truyền dẫn qua đường dây điện thoại hoặc cáp quang) và ngược lại. Có nhiều loại modem như:

* **Modem DSL (Digital Subscriber Line)**: Sử dụng đường dây điện thoại để kết nối mạng.
* **Modem cáp (Cable Modem)**: Sử dụng cáp truyền hình để kết nối với nhà cung cấp dịch vụ Internet.
* **Modem quang (Fiber Modem)**: Sử dụng cáp quang để kết nối với mạng tốc độ cao.

## Firewall (Tường lửa)

Firewall là một thiết bị hoặc phần mềm bảo mật mạng dùng để kiểm soát lưu lượng dữ liệu ra vào mạng. Firewall có nhiệm vụ:

* **Lọc lưu lượng mạng**: Firewall giám sát các gói tin dữ liệu và quyết định cho phép hoặc chặn chúng dựa trên các quy tắc bảo mật được cấu hình.
* **Bảo vệ mạng**: Bảo vệ mạng nội bộ khỏi các cuộc tấn công từ bên ngoài như hacker, virus, hoặc malware.

## Access Point (Điểm truy cập không dây)

Access Point (AP) là thiết bị cho phép các thiết bị không dây kết nối với mạng có dây thông qua sóng Wi-Fi. Một access point thường *kết nối với switch hoặc router qua cáp Ethernet và phát sóng Wi-Fi cho các thiết bị di động* (như điện thoại, laptop) trong mạng không dây (WLAN). Một số tính năng của AP:

* **Mở rộng phạm vi mạng**: Access Point có thể mở rộng phạm vi mạng không dây, cho phép các thiết bị kết nối ở xa hơn.
* **Hỗ trợ nhiều thiết bị**: AP có thể kết nối hàng chục đến hàng trăm thiết bị không dây.

## Network Interface Card (NIC - Card giao tiếp mạng)

Kết nối giữa máy tính và cáp mạng để phát hoặc nhận dữ liệu với các máy tính khác thông qua mạng.

NIC là thành phần phần cứng trong các thiết bị, cho phép chúng kết nối với mạng. Mỗi thiết bị có thể có một NIC riêng (có dây hoặc không dây) để tương tác với mạng:

* **NIC có dây (Ethernet card)**: Kết nối thiết bị với mạng qua cáp Ethernet.
* **NIC không dây (Wireless NIC)**: Cho phép thiết bị kết nối với mạng không dây qua Wi-Fi.

NIC cũng có địa chỉ MAC (Media Access Control) riêng, giúp nhận diện duy nhất thiết bị trong mạng.

A close-up of a sign

Description automatically generated

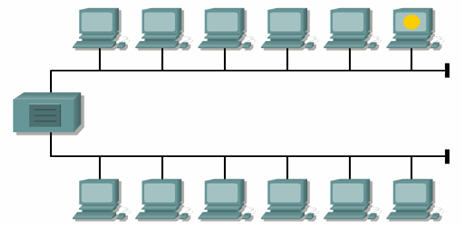
## Server (Máy chủ)

Máy chủ là các thiết bị chuyên cung cấp dịch vụ hoặc dữ liệu cho các thiết bị khác trong mạng (client). Có nhiều loại server khác nhau:

* **Web server**: Lưu trữ và cung cấp các trang web.
* **File server**: Cung cấp dịch vụ lưu trữ và quản lý các tệp dữ liệu.
* **Mail server**: Quản lý và cung cấp dịch vụ gửi và nhận email.

## Repeater (Bộ lặp/ bộ chuyển tiếp)

* Repeater là thiết bị mạng dùng để khuếch đại hoặc tái tạo tín hiệu khi nó truyền qua khoảng cách xa. Điều này giúp tín hiệu không bị suy giảm hoặc mất mát khi truyền qua các đường truyền dài.
* Repeater thường được sử dụng trong các mạng lớn hoặc khi tín hiệu phải truyền qua các địa điểm có khoảng cách xa.
* Chỉ được dùng nối hai mạng có cùng giao thức truyền thông.
* Hoạt động ở lớp Physical.

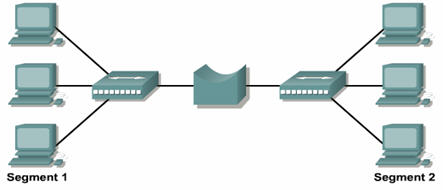
## Hub (bộ tập trung)

* Chức năng **như Repeater** nhưng mở rộng hơn với nhiều đầu cắm các đầu cáp mạng.
* Tạo ra điểm kết nối tập trung để nối mạng theo **kiểu hình sao**.
* Tín hiệu được phân phối đến tất cả các kết nối.
* Có 3 loại Hub: **thụ động, chủ động, thông minh.**
  + **Hub thụ động (Passive Hub):** chỉ đảm bảo chức năng kết nối, không xử lý lại tín hiệu.
  + **Hub chủ động (Active Hub):** có khả năng khuếch đại tín hiệu để chống suy hao.
  + **Hub thông minh (Intelligent Hub):** **là Hub chủ động** nhưng có thêm khả năng tạo ra các gói tin thông báo hoạt động của mình giúp cho việc quản trị mạng dễ dàng hơn.



## Bridge (Cầu nối mạng)

* Dùng để **nối 2 mạng có giao thức giống hoặc khác nhau.**
* Bridge là thiết bị mạng dùng để kết nối và phân đoạn hai hoặc nhiều mạng cục bộ (LAN) với nhau, nhằm giảm lưu lượng trên mạng.
* Bridge hoạt động ở tầng liên kết dữ liệu (Layer 2 của mô hình OSI) và giúp quản lý lưu lượng dữ liệu giữa các phân đoạn mạng khác nhau.
* Dựa trên bảng địa chỉ MAC lưu trữ, Brigde kiểm tra các gói tin và xử lý chúng trước khi có quyết định chuyển đi hay không.

# Truyền dữ liệu và mạng máy tính

## Truyền dữ liệu (Data communications)

Truyền dữ liệu là quá trình trao đổi thông tin giữa các thiết bị điện tử, như máy tính, điện thoại, hoặc các thiết bị mạng. Thông tin có thể là văn bản, hình ảnh, âm thanh hoặc video và được truyền qua các phương tiện truyền dẫn (như cáp, sóng radio, sóng quang). Để thực hiện quá trình này, các thiết bị cần chuyển đổi thông tin thành tín hiệu phù hợp để truyền qua kênh truyền dẫn.

Có hai kiểu truyền dữ liệu chính:

* **Truyền dữ liệu tuần tự (Serial Communication)**: Dữ liệu được gửi từng bit một theo thứ tự từ đầu đến cuối. Các đường truyền dữ liệu hiện đại thường sử dụng phương pháp này vì nó dễ triển khai và có hiệu quả tốt trên khoảng cách xa.
* **Truyền dữ liệu song song (Parallel Communication)**: Nhiều bit dữ liệu được gửi đồng thời trên nhiều kênh truyền dẫn khác nhau. Cách này thường được sử dụng trong các hệ thống ngắn hạn, như kết nối giữa các thành phần trong máy tính.

**Tín hiệu (Signal)**

Trong truyền dữ liệu, thông tin cần được chuyển thành dạng tín hiệu để có thể di chuyển qua các phương tiện truyền dẫn. Có hai loại tín hiệu chính:

* **Tín hiệu tương tự (Analog Signal)**: Là các sóng liên tục biến đổi theo thời gian, có thể có vô hạn giá trị trong khoảng thời gian xác định. Ví dụ như sóng âm thanh hoặc sóng ánh sáng.
* **Tín hiệu số (Digital Signal)**: Là các tín hiệu rời rạc, chỉ có hai trạng thái rõ ràng: 0 hoặc 1 (tương ứng với hai mức điện áp khác nhau). Đây là dạng tín hiệu được sử dụng trong các hệ thống kỹ thuật số như máy tính.

**Mã hoá tín hiệu (Signal Encoding)**

Mã hóa tín hiệu là quá trình chuyển đổi dữ liệu từ dạng này sang dạng khác để truyền qua các kênh truyền dẫn. Quá trình này quan trọng để đảm bảo tín hiệu có thể được truyền hiệu quả và ít bị nhiễu.

Có nhiều phương pháp mã hóa tín hiệu, tùy thuộc vào loại tín hiệu và kênh truyền dẫn:

* **Mã hóa số (Digital Encoding)**: Dữ liệu số được mã hóa thành tín hiệu số. Ví dụ:
  + **NRZ (Non-Return-to-Zero)**: Một dạng mã hóa mà bit "1" được biểu thị bằng mức điện áp cao, và bit "0" là mức điện áp thấp.
  + **Manchester Encoding**: Sử dụng các thay đổi giữa các mức điện áp để biểu diễn dữ liệu (một sự thay đổi từ cao xuống thấp biểu diễn một bit "1", và từ thấp lên cao biểu diễn bit "0").
* **Mã hóa tương tự (Analog Encoding)**: Dữ liệu số có thể được chuyển thành tín hiệu tương tự để truyền qua các phương tiện như cáp quang hoặc sóng radio. Ví dụ:
  + **Amplitude Modulation (AM)**: Biến đổi biên độ của sóng mang tương tự theo dữ liệu.
  + **Frequency Modulation (FM)**: Biến đổi tần số của sóng mang tương tự theo dữ liệu.

## Kết nối mạng (Networking)

* Mạng cục bộ (LAN) 
  + **IEEE 802.x** (***Ethernet***, Token Ring,…)
* Mạng diện rộng (WAN) 
  + POTS, Frame Relay, ISDN, xDSL, ...
* Internet : TCP/IP

### Mạng LAN

**LANs (Local Area Networks)**

* Có giới hạn về địa lý.
* Tốc độ truyền dữ liệu khá cao.
* **Một tổ chức quản lý.**
* Thường dùng **kênh truyền đa truy cập.**
* Những kỹ thuật thường dùng :
  + **Ethernet :** 10/100/1000 Mbps,
  + **Token Ring :** 16 Mbps
  + **FDDI :** 100 Mbps

A diagram of a network

Description automatically generated

### Mạng MAN

**MANs (Metropolitan Area Networks)**

* Có kích thước vùng địa lý **lớn hơn LAN** tuy nhiên **nhỏ hơn WAN.**
* **Một tổ chức** quản lý.
* Thường dùng cáp đồng trục, cáp quang hay sóng ngắn.

A diagram of a network

Description automatically generated

### Mạng WAN

* **WANs (Wide Area Networks)**
* Chúng thường là **sự kết nối nhiều LANs.**
* **Không có giới hạn về địa lý.**
* **Tốc độ** truyền dữ liệu khá **thấp.**
* **Nhiều tổ chức** quản lý.
* Trục chính thường dùng kênh truyền **điểm điểm**
* Những kỹ thuật thường dùng :
  + Các đường dây **điện thoại.**
  + Truyền thông bằng **vệ tinh.**

# Các khái niệm cơ bản

## Giao thức (protocol)

* **cách thức/ các quy tắc** để cho phép các đối tượng (máy tính…) **giao tiếp** với nhau
  + Là một bộ các luật về ngữ nghĩa và cú pháp xác định hành vi của các đơn vị chức năng trong các liên kết trên mạng.
  + Tương tự với ngôn ngữ, ám hiệu,...
  + Định nghĩa giao thức bao gồm định nghĩa **định dạng dữ liệu, nội dung dữ liệu, định thời, và việc kiểm soát lỗi.**
  + Ví dụ các giao thức thuộc TCP/IP:
    - SMTP (Simple Mail Transfer Protocol)
    - POP3 (Post Office Protocol v.3)
    - HTTP (HyperText Transfer Protocol)
    - FTP (File Transfer Protocol)
    - IP (Internetwork Protocol)



## Kiến trúc giao thức

* **Kiến trúc giao thức (Protocol Architecture):** 
  + một **tập hợp các quy tắc, chuẩn mực và phương pháp** cho phép các thiết bị và hệ thống mạng giao tiếp và truyền dữ liệu với nhau. Kiến trúc giao thức quy định cách thức dữ liệu được đóng gói, truyền, xử lý, và giải mã giữa các thiết bị trong một mạng máy tính.
  + *cấu trúc phân lớp của phần cứng và phần mềm* nhằm hỗ trợ việc truyền dữ liệu giữa các hệ thống, còn gọi là *mô hình*
* Tại **mỗi lớp (layer), một hoặc nhiều giao thức** (protocol) được hiện thực cho việc giao tiếp giữa các hệ thống
* **OSI (Open Systems Interconnection):** mô hình được chuẩn hoá, gồm 7 lớp **(mô hình thí nghiệm)**.
* **TCP/IP Protocol:** mô hình được sử dụng rộng rãi nhất **(mô hình thực tế).**

A diagram of different types of internet services

Description automatically generated with medium confidence

|  |  |
| --- | --- |
| **Application Layer** | DHCP, **DNS**, **FTP**, **HTTP**, **IMAP**, LDAP, **POP**, **SMTP**, **SNMP**, SSH, Telnet, TLS/SSL, ... |
| **Transport Layer** | **TCP**, **UDP**, DCCP, SCTP, RSVP, ... |
| **Internet Layer** | **IP (IPv4 - IPv6)**, **ICMP**, **ICMPv6**, ECN, IGMP, IPsec,... |
| **Network Access Layer** | **ARP**, NDP, OSPF, Tunnels, L2TP, PPP, MAC, Ethernet, DSL, ISDN, FDDI, ... |

A diagram of different types of internet

Description automatically generated

Mô hình OSI là một mô hình tham chiếu chuẩn quốc tế, được phát triển bởi ISO (International Organization for Standardization), giúp chuẩn hóa cách thức các hệ thống mạng giao tiếp với nhau. Nó được chia thành **7 tầng**, mỗi tầng đảm nhận một nhiệm vụ riêng biệt trong việc truyền dữ liệu.

**Các tầng trong mô hình OSI:**

### Tầng Vật lý (Physical Layer)

* + Đề cập đến các tiêu chuẩn từ các tổ chức khác, liên quan đến các **đặc điểm vật lý** của phương tiện truyền dẫn, bao gồm đầu nối, chân, cách sử dụng chân, dòng điện, mã hóa, điều chế ánh sáng và các quy tắc về **cách kích hoạt và hủy kích hoạt** việc sử dụng phương tiện vật lý
  + Chịu trách nhiệm truyền các bit dữ liệu thô qua môi trường truyền dẫn vật lý, chẳng hạn như dây cáp đồng, cáp quang, hoặc không dây.

### Tầng Liên kết dữ liệu (Data Link Layer)

* + Định nghĩa các quy tắc xác định thời điểm một thiết bị có thể gửi dữ liệu qua một phương tiện cụ thể.
  + Đảm bảo việc truyền dữ liệu chính xác qua môi trường vật lý.
  + Chịu trách nhiệm phát hiện và sửa lỗi ở mức khung dữ liệu (frame), kiểm soát luồng và tạo địa chỉ MAC cho các thiết bị.
  + Giao thức liên kết dữ liệu cũng định nghĩa **định dạng của tiêu đề (header) và phần cuối (trailer)** cho phép các thiết bị được gắn vào phương tiện gửi và nhận dữ liệu thành công
  + Bao gồm các giao thức như **Ethernet và PPP (Point-to-Point Protocol).**

### Tầng Mạng (Network Layer)

* + Có nhiệm vụ cung cấp sự liên lạc giữa các hosts trên các loại networks khác nhau
  + Lớp này định nghĩa ba tính năng chính: **định địa chỉ logic, định tuyến (chuyển tiếp) và xác định đường dẫn.**
    - Định địa chỉ logic định nghĩa cách mỗi thiết bị có thể có một địa chỉ có thể *được quy trình định tuyến sử dụng*.
  + Xử lý định tuyến dữ liệu giữa các mạng khác nhau, đảm bảo rằng các gói tin (packets) được gửi đến đúng đích.
  + Quy định về địa chỉ IP, định tuyến (routing) và quản lý lưu lượng.
  + Giao thức chính: **IP (Internet Protocol).**
    - Dịch vụ **chuyển phát các gói dữ liệu** (IP datagrams)
    - Cơ chế **host-to-host**
    - IP hỗ trợ cơ chế chuyển phát các gói dữ liệu với hai đặc tính: không kết nối (connectionless) và không tin cậy (unreliable).
      * Connectionless: **mỗi gói dữ liệu là độc lập** với các gói dữ liệu khác
      * Unreliable: **không có sự đảm bảo** rằng các gói dữ liệu được chuyển phát đúng đắn

|  |  |  |
| --- | --- | --- |
|  | **IPv4** | **IPv6** |
| Năm triển khai | 1981 | 1999 |
| Chiều dài địa chỉ | 32-bits | 128-bits |
| Định dạng | Dotted Decimal Notation:192.149.252.76 | Hexadecimal Notation: 3FFE:F200:0234:AB00:0123: 4567:8901:ABCD |
| Prefix Notation | 192.149.0.0/24 | 3FFE:F200:0234::/48 |
| Số lượng địa chỉ | 232 = 4,294,967,296 | 2128 = 340,282,366,920,938,463,463,374,607,431,768,211,456 |

Trong địa chỉ IPv4, một vài khối địa chỉ và khuôn mẫu có dạng đặc biệt.

* Tất cả các khối địa chỉ **bắt đầu bằng 10., bằng 172.16. đến 172.31. và bằng 192.168. là chưa được cấp cho bất kỳ máy tính nào trên mạng.** Các khối địa chỉ này được gọi là ***các khối địa chỉ không thể định tuyến được (non-routable) và có thể được sử dụng trên các mạng cục bộ.*** **Không một máy tính nào sử dụng địa chỉ trong các khối trên có thể truy nhập được vào Internet.** Các địa chỉ không thể định tuyến được rất thích hợp cho việc xây dựng các mạng riêng, các mạng này sẽ không xuất hiện trên mạng Internet.
* Các địa chỉ **IPv4 bắt đầu bằng 127** luôn có nghĩa là các địa chỉ **loopback** cục bộ (local loopback address), nghĩa là **các địa chỉ này luôn chỉ tới máy tính cục bộ**. Hostname cho địa chỉ này thường là localhost. (**Localhost là tên mặc định mô tả địa chỉ máy tính cục bộ)**
  + **sử dụng "localhost" để khai báo kết nối *khi kết nối cục bộ***
* Trong IPv6, địa chỉ 0:0:0:0:0:0:0:1 (cũng được viết tắt là :: 1) là địa chỉ loopback.
* Địa chỉ 0.0.0.0 luôn được chỉ đến máy gửi nhưng cũng có thể được dùng như là địa chỉ nguồn mà không phải địa chỉ đích. Tương tự, trong IPv4, bất kỳ địa chỉ nào bắt đầu bằng 0. (tám bit không) là chỉ đến một máy tính trong cùng một mạng cục bộ.

### Tầng Giao vận (Transport Layer)

* + Chịu trách nhiệm cho giao tiếp giữa các máy (host) với nhau và tạo **kết nối logic** giữa hai thiết bị.
  + Bao gồm **khởi tạo kết nối** giữa các thiết bị, **kiểm soát luồng** giữa các thiết bị, **phân phối cùng thứ tự và giao tiếp đa kênh**
  + Đảm bảo việc truyền dữ liệu đáng tin cậy và kiểm soát lưu lượng giữa hai hệ thống đầu cuối.
  + Cung cấp dịch vụ **phân mảnh và lắp ráp lại các gói tin**, đồng thời **xử lý các yêu cầu truyền dữ liệu liên tiếp (end-to-end).**
  + Các giao thức của lớp này cung cấp một số lượng lớn các dịch vụ, tập trung vào các vấn đề liên quan đến việc **phân phối dữ liệu đến máy tính khác** (ví dụ: phục hồi lỗi và kiểm soát luồng)
  + Giao thức chính: **TCP** (Transmission Control Protocol), **UDP** (User Datagram Protocol).
    - **UDP:** transport protocol cho việc giao tiếp giữa các processes
      * **Sử dụng IP** để vận chuyển các gói dữ liệu (datagrams) đến host nhận.
      * **Sử dụng cổng (ports) để tạo các giao tiếp** giữa các processes.
      * **Đặc điểm:** Chuyển phát các gói dữ liệu (datagram); **Không kết nối (connectionless); Không tin cậy (unreliable);** Gọn nhẹ (lightweight)

A diagram of a data diagram

Description automatically generated

A white clouds with green arrows

Description automatically generated with medium confidence

**Tại sao lại sử dụng UDP?**

* + - Có thể tinh chỉnh những gì sẽ truyền đi và truyền đi lúc nào
      * Có cơ chế đóng gói dữ liệu
      * Gói dữ liệu được truyền đi ngay khi được gửi ra socket
    - **Độ trễ thấp:** Không có các cơ chế thiết lập kết nối và xác nhận dữ liệu
    - Không cần duy trì trạng thái kết nối
      * Không cần bộ đệm, tham số, số thứ tự của gói dữ liệu
      * Dễ dàng duy trì nhiều clients hoạt động cùng lúc
    - **Packet Header gọn nhẹ**
  + **TCP - Transmission Control Protocol**
    - Hướng kết nối (connection-oriented)
      * Một kết nối ảo được thiết lập trước khi dữ liệu được truyền
      * Nếu có vấn đề với kết nối (không thiết lập được hoặc bị ngắt) thì chương trình sẽ được cảnh báo
    - Có độ tin cậy: Mọi thao tác truyền dữ liệu đều được xác nhận từ phía thu
    - Byte-Stream
      * Dữ liệu được truyền theo dòng bytes
      * **Không cần đóng gói dữ liệu** thành các datagrams như UDP
    - **Full-duplex:** Dữ liệu có thể truyền theo hai hướng trên cùng một kết nối ảo
    - Không gọn nhẹ như UDP

A group of arrows pointing to a few white clouds

Description automatically generated with medium confidence

* + Transport Layer: Dùng cặp *<***Địa chỉ IP, Số hiệu cổng***>* nhận dạng dữ liệu gửi cho ứng dụng (process) nào đang chạy ở host



A diagram of a diagram of a number

Description automatically generated with medium confidence

* Ứng dụng đặt nặng vấn đề truyền dữ liệu với độ tin cậy, cần cơ chế xác nhận: **TCP**
* Ứng dụng cần tốc độ truyền nhận dữ liệu theo thời gian thực, cho phép một số gói tin có thể không đến đích (ví dụ các ứng dụng multimedia): **UDP**

### Tầng Phiên (Session Layer)

* + Quản lý các phiên kết nối giữa các hệ thống mạng, thiết lập, duy trì và kết thúc phiên.
  + Các thiết bị sẽ thống nhất về **loại giao tiếp sẽ diễn ra bao gồm: Simplex, Half duplex, Full-duplex**
  + Đảm bảo rằng dữ liệu được đồng bộ hóa và bảo vệ trong quá trình truyền.
  + Các giao thức được hỗ trợ trên lớp này bao gồm: RPC, **SQL**, NetBIOS, Point-to Point Tunneling Protocol (PPTP)

### Tầng Trình bày (Presentation Layer)

* + Chịu trách nhiệm về định dạng dữ liệu (văn bản ASCII, văn bản EBCDIC, nhị phân, BCD và JPEG), mã hóa và giải mã dữ liệu, cũng như nén dữ liệu để tối ưu hóa truyền tải.
  + Đảm bảo rằng **dữ liệu được chuyển đổi thành định dạng mà cả hai đầu giao tiếp hiểu được.**

### Tầng Ứng dụng (Application Layer)

* + Cung cấp các giao diện và dịch vụ để người dùng hoặc các ứng dụng có thể truy cập vào mạng. (giữa phần mềm truyền thông và bất kỳ ứng dụng nào cần giao tiếp bên ngoài máy tính mà ứng dụng đó lưu trú.)
  + Nó cũng xác định các quy trình để xác thực người dùng.
  + Ví dụ: giao thức HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol).

**Minh hoạ mô hình TCP/IP**

A diagram of a network connection

Description automatically generated

**1. Host A và Host B**

* **Host A** và **Host B** đại diện cho hai thiết bị hoặc hệ thống mạng khác nhau (ví dụ: máy tính, server) tham gia vào quá trình giao tiếp.
* Mỗi **Host** chứa các lớp trong mô hình TCP/IP, bao gồm:
  + **Application** (ứng dụng): Ở đây có các ứng dụng như **App X** và **App Y** (ứng dụng số 1, 2, 3, 4, 6).
  + **Transport** (giao vận): **TCP** là giao thức tầng giao vận, chịu trách nhiệm tạo kết nối logic (logical connection) giữa hai thiết bị và đảm bảo việc truyền dữ liệu đáng tin cậy.
  + **Network** (mạng): **IP** là giao thức tầng mạng, chịu trách nhiệm định địa chỉ và định tuyến gói tin (packets) giữa các mạng khác nhau thông qua địa chỉ IP toàn cầu.
  + **Network Access Protocol**: Tầng truy cập mạng, nơi các gói tin IP được truyền qua mạng vật lý (Physical), với các giao thức kết nối với mạng như Ethernet hoặc Wi-Fi.

**2. TCP (Transmission Control Protocol)**

* **TCP** là một phần quan trọng trong quá trình truyền dữ liệu, tạo ra kết nối logic giữa Host A và Host B.
* Kết nối này được biểu diễn qua các **port** (cổng), như App X ở Host A liên kết với App X ở Host B thông qua cổng 2, App Y sử dụng cổng 1 và 4, v.v. **Mỗi cổng này là một điểm truy cập dịch vụ (Service Access Point - SAP).**
* **TCP** đảm bảo rằng dữ liệu từ các ứng dụng khác nhau sẽ được truyền đi một cách đáng tin cậy và tuần tự.

**3. IP (Internet Protocol)**

* **IP** là giao thức chịu trách nhiệm định tuyến các gói tin giữa các mạng khác nhau.
* Địa chỉ IP toàn cầu của thiết bị giúp xác định rõ nguồn và đích của dữ liệu.

**4. Router J**

* **Router J** là thiết bị định tuyến, giúp chuyển tiếp các gói tin giữa hai mạng khác nhau (Network 1 và Network 2).
* Router J chứa tầng **IP** và hai giao thức truy cập mạng (NAP 1 và NAP 2) để kết nối với hai mạng vật lý khác nhau.
* **NAP** (Network Access Protocol) chịu trách nhiệm **liên kết các gói tin vào môi trường mạng vật lý.**

**5. Mạng vật lý (Physical Layer)**

* Mạng vật lý là tầng thấp nhất, nơi các tín hiệu thực tế được truyền qua các phương tiện truyền dẫn như cáp đồng, cáp quang hoặc không dây.
* Trong hình, hai mạng vật lý **Network 1** và **Network 2** đại diện cho hai mạng khác nhau mà Host A và Host B đang kết nối.

**Quá trình truyền dữ liệu**

* **Từ Host A đến Host B**: Dữ liệu từ các ứng dụng tại Host A (như App X, App Y) được truyền qua các tầng TCP, IP, và Network Access Protocol trước khi được đưa vào mạng vật lý và truyền đến Router J.
* **Router J** đóng vai trò định tuyến gói tin từ Network 1 sang Network 2, sử dụng các giao thức truy cập mạng để chuyển đổi giữa các mạng vật lý khác nhau.
* **Host B** nhận dữ liệu qua tầng IP, sau đó chuyển dữ liệu qua TCP để đảm bảo rằng nó được gửi đến đúng ứng dụng (App X hoặc App Y) theo đúng thứ tự.

## Một số bộ giao thức kết nối mạng & Các cổng phổ biến

A diagram of a number of different types of numbers

Description automatically generated with medium confidence

A screenshot of a table

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Service** | **Port** | **Function** |
| FTPS | 989990 | File transfer (secure) |

## Cổng (Port)

* Trong lập trình mạng, khái niệm về port đóng vai trò quan trọng như là các "cổng kết nối" cho các ứng dụng và dịch vụ trên internet; *Để tạo* ***điểm truyền/ nhận dữ liệu***
* Port (cổng) là một điểm truy cập dịch vụ (service access point) trên một thiết bị, cho phép giao tiếp với các ứng dụng khác hoặc với các thiết bị mạng khác.
* *Mỗi thiết bị kết nối với internet thông qua một địa chỉ IP*, và trong mỗi thiết bị này, có hàng ngàn cổng (ports) sẵn sàng để tiếp nhận và gửi dữ liệu. Cụ thể, mỗi dịch vụ hoặc ứng dụng trên một thiết bị có thể lắng nghe tại một cổng cụ thể.
* Mỗi cổng được định danh bằng một số nguyên dương (0 -65535)
* Ví dụ, các trang web thường lắng nghe tại cổng 80 (cho HTTP) hoặc cổng 443 (cho HTTPS). Đồng thời, các dịch vụ email thường sử dụng cổng 25 (SMTP) để gửi email và cổng 110 (POP3) hoặc cổng 143 (IMAP) để nhận email.
* Chức năng chính của port là cho phép **OS *phân biệt các dịch vụ hoặc ứng dụng/ tiến trình khác nhau*** *trên cùng một thiết bị.*
* *Nói đơn giản, port giúp xác định tiến trình (process) nào sẽ được nhận dữ liệu tương ứng.*
* Khái niệm về port cho phép các ứng dụng và dịch vụ trên internet tương tác với nhau một cách cấu trúc và có tổ chức. Có thể nhiều ứng dụng chạy trên cùng một máy tính và sử dụng các cổng khác nhau, **giúp chúng hoạt động song song mà không gây ra xung đột hoặc giao lẫn** với nhau. Port không chỉ giới hạn trong việc truy cập internet; chúng cũng được sử dụng trong các mạng nội bộ của doanh nghiệp để kiểm soát việc truy cập các dịch vụ và tài nguyên mạng.
* Lưu ý: **Mỗi process có thể chiếm dụng nhiều ports, nhưng mỗi port chỉ được đăng ký cho một process**
* Qua việc quản lý và điều hướng dữ liệu thông qua các cổng, port đóng vai trò quan trọng trong việc duy trì an toàn, hiệu quả và cấu trúc hóa của mạng một cách có tổ chức.

A diagram of a diagram

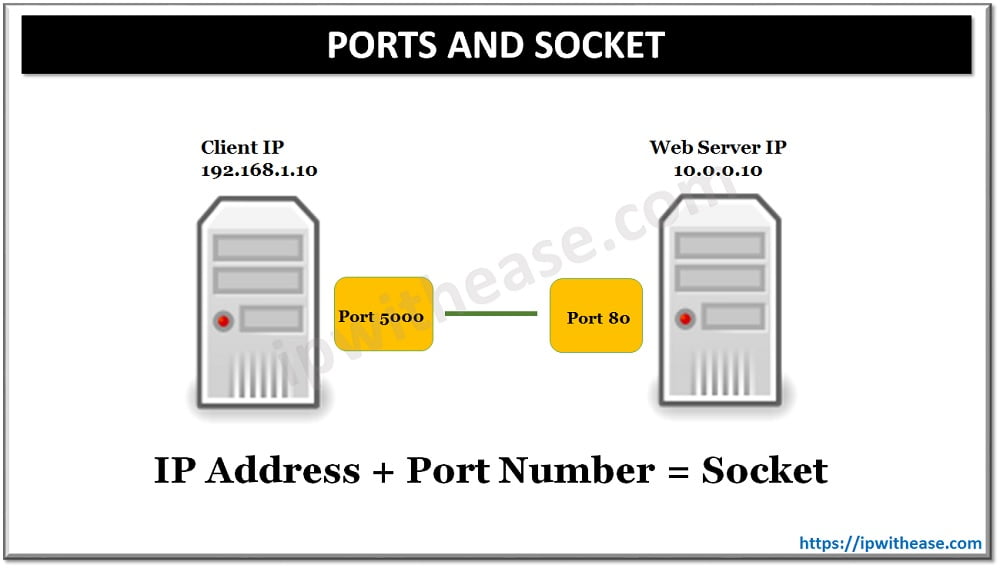
Description automatically generated

## Interface

* Interface (giao diện) là điểm kết nối giữa hai hệ thống, có thể là giữa **phần cứng với phần cứng, phần mềm với phần mềm, hoặc phần mềm với phần cứng.**
* Trong ngữ cảnh mạng, một interface thường đề cập đến một điểm truy cập vật lý hoặc logic trên một thiết bị (ví dụ: một cổng mạng, một cổng USB, một API (Application Programming Interface) giữa các phần mềm).
* Interfaces có thể bao gồm các giao diện mạng (network interfaces) như Ethernet, Wi-Fi, hoặc các giao diện phần mềm giữa các ứng dụng.
* Chức năng chính của interface là **cung cấp điểm kết nối và giao tiếp giữa các hệ thống** khác nhau (có thể là phần cứng, phần mềm, hoặc giữa phần cứng và phần mềm).
* Interface tạo điều kiện cho việc truyền dữ liệu và giao tiếp giữa các thành phần khác nhau của hệ thống.
* Interface có thể xuất hiện ở nhiều tầng khác nhau trong mô hình TCP/IP, bao gồm tầng vật lý (Physical Layer) (như các cổng mạng), tầng liên kết dữ liệu (Data Link Layer) (như giao thức Ethernet), và tầng ứng dụng (Application Layer) (như API).
* Ví dụ:
  + Ethernet Interface: **Giao diện vật lý** mà một thiết bị sử dụng để kết nối với mạng LAN qua cáp Ethernet.
  + USB Interface: Giao diện kết nối cho phép các thiết bị như bàn phím, chuột, và ổ đĩa cứng kết nối với máy tính.

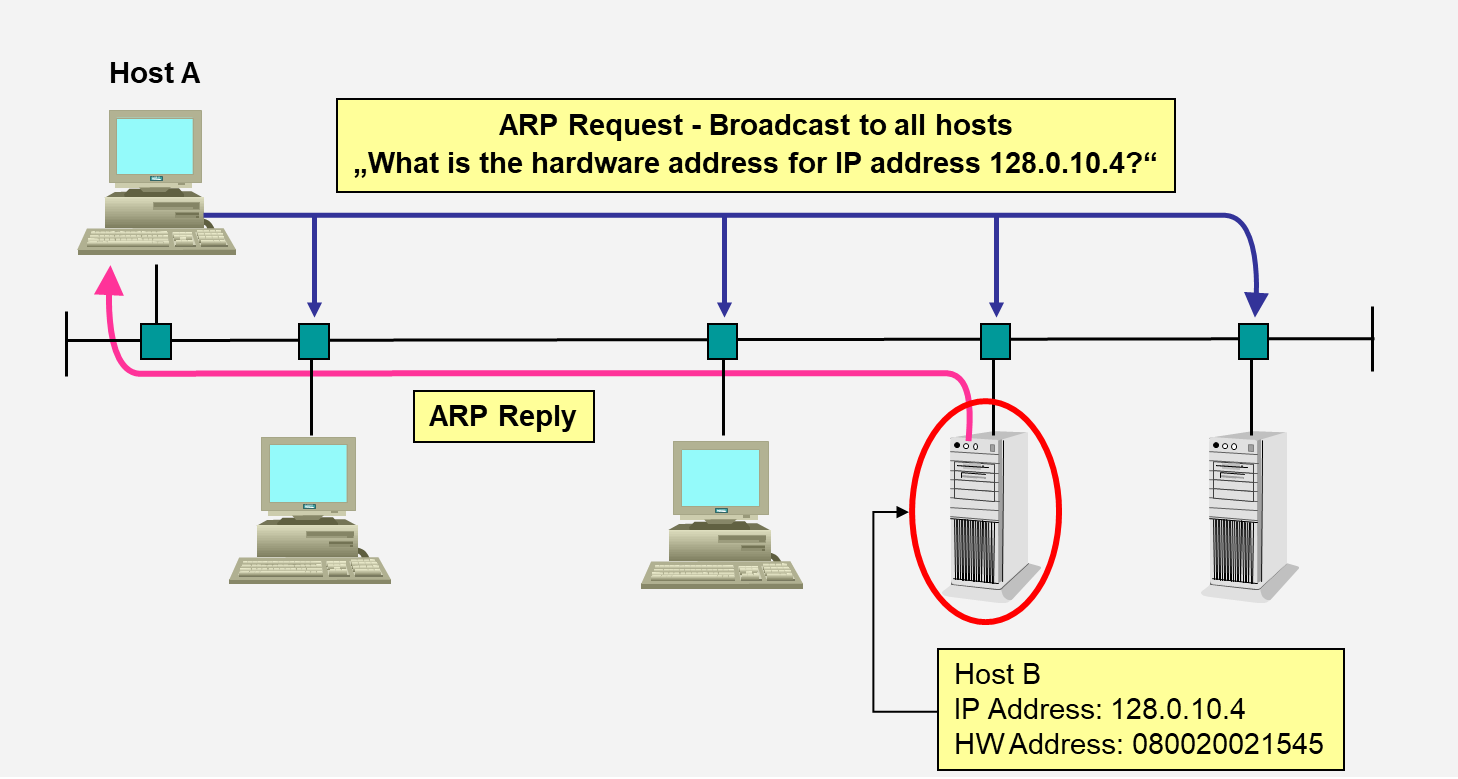
## Socket

* Socket là **điểm cuối** trong liên kết truyền thông hai chiều (two-way communication) biểu diễn kết nối giữa Client - Server.
* Socket là sự kết hợp của:
  + Địa chỉ IP (IP Address)
  + Port (Cổng)
  + Giao thức truyền thông (Protocol: TCP hoặc UDP)
* Là một trong những phương thức giao tiếp.
* **Các lớp Socket được ràng buộc với một cổng port** (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới.
  + Quá trình gửi/nhận từ Socket.
  + Socket có thể hiểu như là cái cửa/cổng.
  + Quá trình gửi đẩy dữ liệu ra khỏi cửa/cổng.
  + Quá trình gửi dựa vào tầng Transport ở phía bên kia cửa/cổng để gửi dữ liệu đến cửa/cổng tại quá trình nhận.
  + Hai cửa/cổng có liên quan với nhau: một ở mỗi bên.
* Socket là một điểm cuối trong quá trình giao tiếp giữa hai thiết bị qua mạng. **Nó giúp xác định một kết nối cụ thể bằng cách ghép:**
  + **Địa chỉ IP nguồn + Port nguồn**
  + **Địa chỉ IP đích + Port đích**



**Network Access Layer: ARP & RARP**

* ARP - Address Resolution Protocol (RFC826)
* RARP - Reverse ARP (RFC903)
* Các giao thức dùng để chuyển đổi địa chỉ luận lý (địa chỉ IP) thành địa chỉ vật lý (địa chỉ MAC) hoặc ngược lại.
* Ví dụ: 192.168.0.3 ⇄ 00:00:1A:3E:02:56



A diagram of a network

Description automatically generated

## Ứng dụng mạng

* Những chương trình phần mềm **sử dụng môi trường mạng để trao đổi dữ liệu**
* Các ứng dụng phổ biến như trình duyệt web (Chrome, Firefox), ứng dụng email (Gmail, Outlook), hoặc ứng dụng trò chuyện (Skype, Zalo) đều là các ứng dụng mạng.
* Thường có 2 đối tượng tham gia tương tác:
  + **Client (Khách hàng)**: Thiết bị hoặc phần mềm yêu cầu dịch vụ từ một máy chủ.
  + **Server (Máy chủ)**: Thiết bị hoặc phần mềm cung cấp dịch vụ theo yêu cầu từ phía client.
  + Ví dụ: Trong mô hình web, trình duyệt web là **client**, yêu cầu trang web từ máy chủ web (**server**), và máy chủ sẽ trả về trang web đã yêu cầu.

## Lập trình mạng

* Là lập trình có sử dụng trực tiếp các protocols ở các layers của kiến trúc TCP/IP
* Dùng ngôn ngữ lập trình để tạo ra ứng dụng mạng
  + Xây dựng cấu trúc các đối tượng: Thiết kế các thành phần như *client, server, hoặc các module* khác, mỗi đối tượng sẽ có trách nhiệm và vai trò cụ thể.
  + Xây dựng giao thức liên kết giữa các đối tượng
  + Quản lý- đồng bộ hoạt động của hệ thống
    - Nhằm đảm bảo tất cả các đối tượng (client, server) hoạt động đúng thứ tự và nhất quán.
    - Điều này có thể bao gồm việc xử lý các lỗi kết nối, đồng bộ hóa dữ liệu giữa các client và server, và duy trì trạng thái hệ thống.
  + **Chủ yếu theo mô hình Client/ Server** (Ngoài ta còn có *Peer-to-Peer (P2P), Mô hình lai (Hybrid Model), Mô hình Publish/Subscribe (Pub/Sub)…)*

### Mô hình Peer-to-Peer và mô hình Client/Server

|  |  |
| --- | --- |
| **Peer-to-peer**   * + mọi trạm làm việc ngang hàng nhau   + Mỗi thiết bị (Peer) có thể vừa là Client vừa là Server, tự chia sẻ tài nguyên trực tiếp với các thiết bị khác trong mạng.   + Uncontrolled   + Khó quản lý: Do không có máy chủ trung tâm, việc kiểm soát truy cập và bảo mật khó khăn hơn.   + Không tối ưu trong việc tìm kiếm và truyền tải dữ liệu lớn. |  |
| **Client - server**   * + là mô hình mà một thiết bị (Client) gửi yêu cầu đến một thiết bị khác (Server), và Server sẽ xử lý yêu cầu đó rồi trả lại kết quả cho Client.   + server kiểm soát quá trình truy cập   + web, email, ftp,...   + Điểm hỏng duy nhất: Nếu Server gặp sự cố, dịch vụ có thể bị gián đoạn. |  |

### Mô hình Client/Server

Mô hình **Client/Server** được sử dụng chủ yếu trong lập trình mạng vì nó có một số ưu điểm nổi bật so với các mô hình khác:

**1. Dễ dàng quản lý và bảo trì:**

* Trong mô hình **Client/Server**, toàn bộ dữ liệu và logic xử lý chính thường được tập trung tại **máy chủ (Server)**. Điều này giúp quản trị viên mạng dễ dàng quản lý, cập nhật và bảo trì hệ thống.
* Ví dụ, nếu cần cập nhật phần mềm hoặc thay đổi cấu hình, chỉ cần thực hiện trên Server thay vì phải cập nhật trên tất cả các máy **Client**.

**2. Bảo mật cao hơn:**

* **Server** đóng vai trò trung tâm và quản lý tất cả các kết nối và truy cập dữ liệu, giúp hệ thống dễ dàng kiểm soát quyền truy cập, bảo mật thông tin, và đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể truy cập dữ liệu.
* Với các chính sách bảo mật được triển khai trên Server, việc ngăn chặn và xử lý các truy cập trái phép, tấn công mạng trở nên đơn giản hơn.

**3. Tài nguyên được quản lý tập trung:**

* Trong mô hình Client/Server, các tài nguyên như cơ sở dữ liệu, ứng dụng, và file đều được lưu trữ trên **Server**. Điều này giúp giảm tải việc lưu trữ và quản lý tài nguyên trên các **Client**, đồng thời dễ dàng kiểm soát phiên bản, duy trì tính toàn vẹn của dữ liệu.
* Tài nguyên được phân bổ và cung cấp từ Server một cách có hệ thống, giúp tối ưu hiệu suất sử dụng tài nguyên.

**4. Hiệu quả khi xử lý nhiều người dùng:**

* Server có thể phục vụ nhiều **Client** cùng một lúc. Nhờ vào các máy chủ mạnh mẽ và có khả năng mở rộng, hệ thống có thể đáp ứng đồng thời một lượng lớn người dùng mà không bị gián đoạn.
* Mô hình này phù hợp với các dịch vụ có hàng nghìn, hàng triệu người dùng truy cập, chẳng hạn như các trang web, hệ thống email, hay dịch vụ truyền phát video (streaming).

**5. Khả năng mở rộng tốt:**

* **Client/Server** có khả năng mở rộng hệ thống một cách dễ dàng. Nếu có nhu cầu mở rộng hệ thống (ví dụ như thêm nhiều **Client** hoặc xử lý nhiều yêu cầu hơn), chỉ cần nâng cấp phần cứng hoặc mở rộng số lượng **Server**.
* Điều này giúp mô hình Client/Server phù hợp với các hệ thống lớn và phức tạp như hệ thống ngân hàng, mạng xã hội, và các dịch vụ doanh nghiệp.

**6. Tính đồng nhất và nhất quán cao:**

* Do tất cả các **Client** đều truy cập vào **Server** để lấy thông tin và xử lý dữ liệu, hệ thống đảm bảo tính nhất quán và đồng nhất của dữ liệu. Điều này rất quan trọng đối với các hệ thống yêu cầu tính chính xác cao như ngân hàng, quản lý dữ liệu khách hàng, hay các hệ thống giao dịch trực tuyến.

**7. Hỗ trợ dễ dàng triển khai các mô hình dịch vụ:**

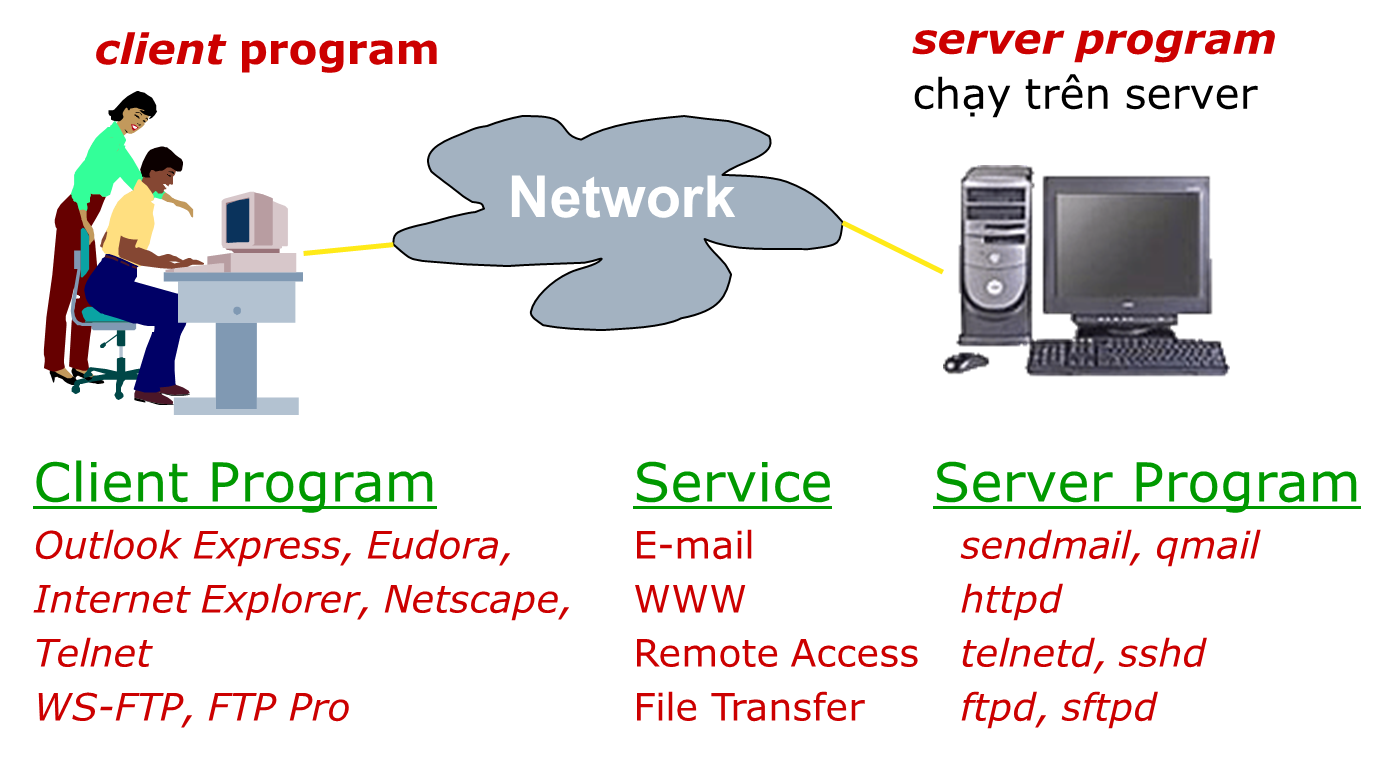
* Mô hình Client/Server là nền tảng của các mô hình dịch vụ hiện đại như **Software as a Service (SaaS)**, **Platform as a Service (PaaS)**, hay **Infrastructure as a Service (IaaS)**.
* Các công ty có thể cung cấp dịch vụ thông qua Server cho nhiều Client khác nhau, mà không cần người dùng phải cài đặt bất kỳ phần mềm phức tạp nào trên máy tính cá nhân của họ.

**8. Khả năng kiểm soát và theo dõi hoạt động:**

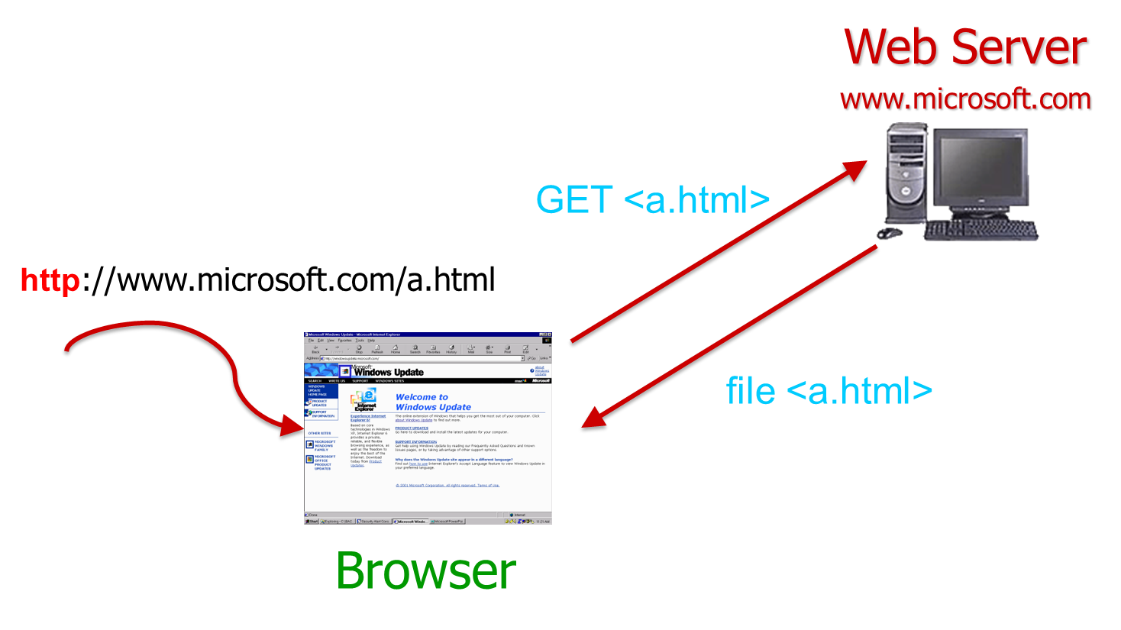
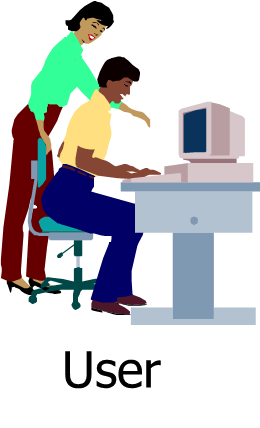
* Vì *tất cả các yêu cầu đều phải thông qua* ***Server***, hệ thống có khả năng giám sát và theo dõi hoạt động của từng **Client** một cách dễ dàng. Điều này giúp phát hiện sớm các vấn đề như lỗi kết nối, tấn công mạng, hoặc việc sử dụng tài nguyên không đúng cách.
* Ví dụ: Quản trị viên có thể theo dõi ai đang sử dụng dịch vụ, những dữ liệu nào đang được yêu cầu, và đảm bảo rằng hệ thống hoạt động ổn định.

**Nhược điểm của mô hình Client/Server:**

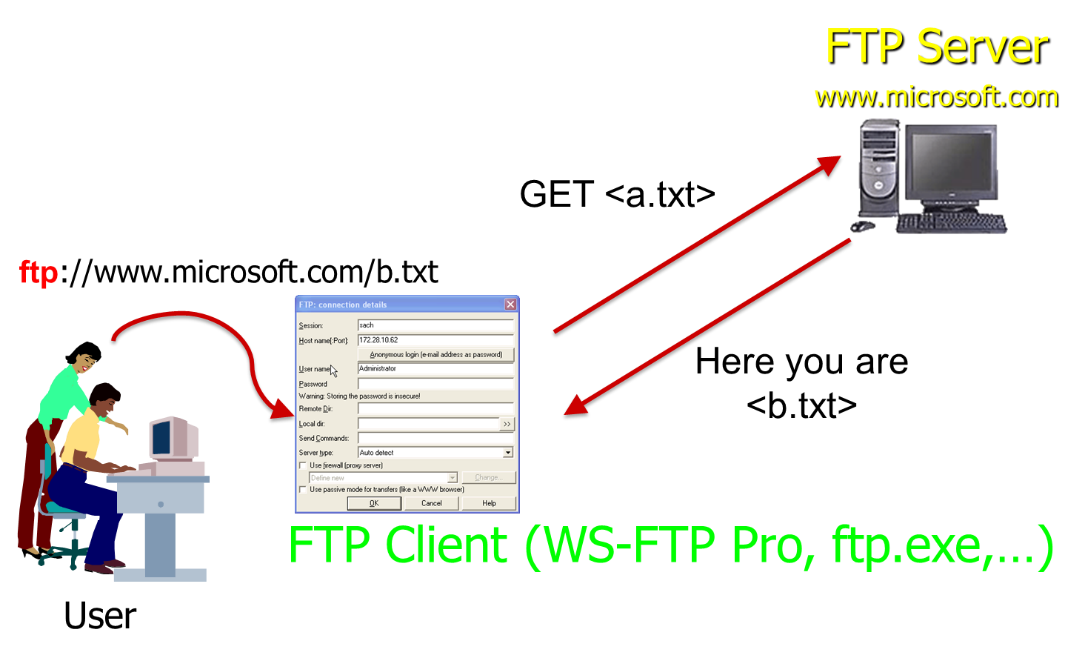
* **Điểm hỏng duy nhất**: Nếu Server gặp sự cố hoặc bị tấn công, toàn bộ hệ thống có thể ngừng hoạt động.
* **Chi phí cao**: Việc duy trì và nâng cấp Server có thể đòi hỏi chi phí lớn, đặc biệt đối với các doanh nghiệp vừa và nhỏ.
* **Hiệu suất bị giới hạn**: Khi có quá nhiều yêu cầu từ Client, Server có thể bị quá tải, ảnh hưởng đến hiệu suất của hệ thống.



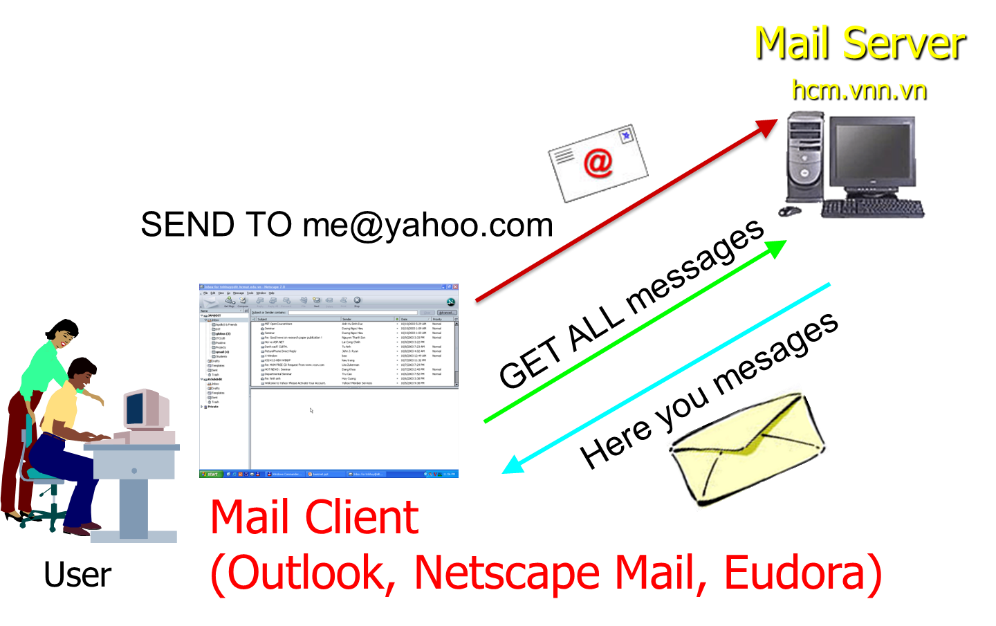
### WWW - Kết nối đến Web Server



### File Transfer Protocol (FTP) - FTP client và FTP Server



### Email



### Webmail

* cách thức hoạt động tương tự Email.
* WebMail Client = Browser (Yahoo, Hotmail, FPTNET, VASC,…)

### Chat

