

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



Phạm Tùng Lâm

**XÂY DỰNG ỨNG DỤNG GAME
ĐA NGƯỜI DÙNG SỬ DỤNG SOCKET
BẰNG NGÔN NGỮ PYTHON**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ kỹ thuật điện tử viễn thông – Chất lượng cao

HÀ NỘI - 2022

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

Phạm Tùng Lâm

**XÂY DỰNG ỨNG DỤNG GAME
ĐA NGƯỜI DÙNG SỬ DỤNG SOCKET
BẰNG NGÔN NGỮ PYTHON**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ kỹ thuật điện tử viễn thông - Chất lượng cao

Cán bộ hướng dẫn: TS. Bùi Trung Ninh

HÀ NỘI - 2022

LỜI CẢM ƠN

Quá trình thực hiện khóa luận tốt nghiệp là giai đoạn quan trọng nhất trong quãng đời mỗi sinh viên. Khóa luận tốt nghiệp là tiền đề nhằm trang bị cho chúng em những kỹ năng nghiên cứu, những kiến thức quý báu trước khi lập nghiệp.

Trước hết, chúng em xin chân thành cảm ơn quý thầy, cô Khoa Điện tử - Viễn thông. Đặc biệt là các thầy, cô trong bộ môn Hệ thống viễn thông đã tận tình chỉ dạy và trang bị cho em những kiến thức cần thiết trong suốt thời gian ngồi trên ghế giảng đường, làm nền tảng cho em có thể hoàn thành được bài khóa luận này.

Em xin trân trọng cảm ơn thầy Bùi Trung Ninh đã tận tình giúp đỡ, định hướng cách tư duy và cách làm việc khoa học. Đó là những góp ý hết sức quý báu không chỉ trong quá trình thực hiện khóa luận này mà còn là hành trang tiếp bước cho em trong quá trình học tập và lập nghiệp sau này.

Khóa luận của em còn những hạn chế về năng lực và những thiếu sót trong quá trình nghiên cứu. Em xin lắng nghe và tiếp thu những ý kiến của giáo viên phản biện để hoàn thiện, bổ sung kiến thức.

Em xin chân thành cảm ơn!

TÓM TẮT

Tóm tắt: Cùng với sự phát triển nhanh chóng của Internet, Game Online hiện nay cũng đang thịnh hành và trở thành một phần không thể thiếu của nhiều tầng lớp trong xã hội. Có thể nhận thấy rõ ràng rằng để lập trình được một Game Online, ngoài việc lập trình viên phải có những kiến thức về đồ họa, về thuật toán... thì một phần không thể thiếu đó là kiến thức về mạng và việc truyền thông tin trên mạng. Vì vậy, nếu coi Game Online là công cụ để lập trình viên tìm hiểu và thực hành những kiến thức về mạng thì đây thực sự là một công cụ rất hữu hiệu. Cũng vì lí do này nên em đã quyết định chọn đề tài “**Xây dựng ứng dụng game đa người dùng sử dụng socket bằng ngôn ngữ python**”. Nội dung của khóa luận sẽ tập trung vào lập trình socket, sự giao tiếp client – server cũng như thuật toán của game mà sẽ không chú trọng vào phần đồ họa, hình ảnh của game.

Từ Khóa: *Lập trình mạng socket, ngôn ngữ Python, game đa người dùng.*

LỜI CAM ĐOAN

Em xin cam đoan kết quả đạt được trong khóa luận là sản phẩm của riêng cá nhân, không sao chép lại của người khác. Trong toàn bộ nội dung của khóa luận, những điều được trình bày hoặc là của cá nhân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp. Em xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Hà Nội, ngày tháng năm 2022

Sinh viên

MỤC LỤC

LỜI NÓI ĐẦU

DANH MỤC CÁC TỪ VIẾT TẮT

DANH MỤC CÁC HÌNH ẢNH

DANH MỤC CÁC BẢNG BIỂU

Chương 1: TỔNG QUAN VỀ MẠNG MÁY TÍNH VÀ NGÔN NGỮ PYTHON

- 1.1. Định nghĩa mạng máy tính
- 1.2. Phân loại mạng máy tính
- 1.3. Mô hình phân tầng
 - 1.3.1. Mô hình OSI
 - 1.3.2. Mô hình TCP/IP
- 1.4. Các Giao thức mạng
 - 1.4.1. Giao thức TCP
 - 1.4.2. Giao thức UDP
- 1.5. Các mô hình ứng dụng mạng
 - 1.5.1. Mô hình khách – máy chủ (Client – Server)
 - 1.5.2. Mô hình mạng ngang hàng (Peer to Peer)
 - 1.5.3. Mô hình mạng lai (Hybrid)
- 1.6. Tổng quan về ngôn ngữ python

Chương 2: LẬP TRÌNH SOCKET

- 2.1. Khái niệm lập trình socket
- 2.2. Một số thuộc tính của socket
- 2.3. Socket trong python
- 2.4. Demo chương trình lập trình socket bằng python

Chương 3: Xây dựng chương trình ứng dụng

- 3.1. Giới thiệu về game
- 3.2. Phân tích và thiết kế
- 3.3. Cài đặt và kết quả đạt được
- 3.4. Nhận Xét

KẾT LUẬN

PHỤ LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

DANH MỤC CÁC HÌNH ẢNH

DANH MỤC CÁC BẢNG BIỂU

CHƯƠNG 1: TỔNG QUAN VỀ MẠNG MÁY TÍNH VÀ NGÔN NGỮ PYTHON

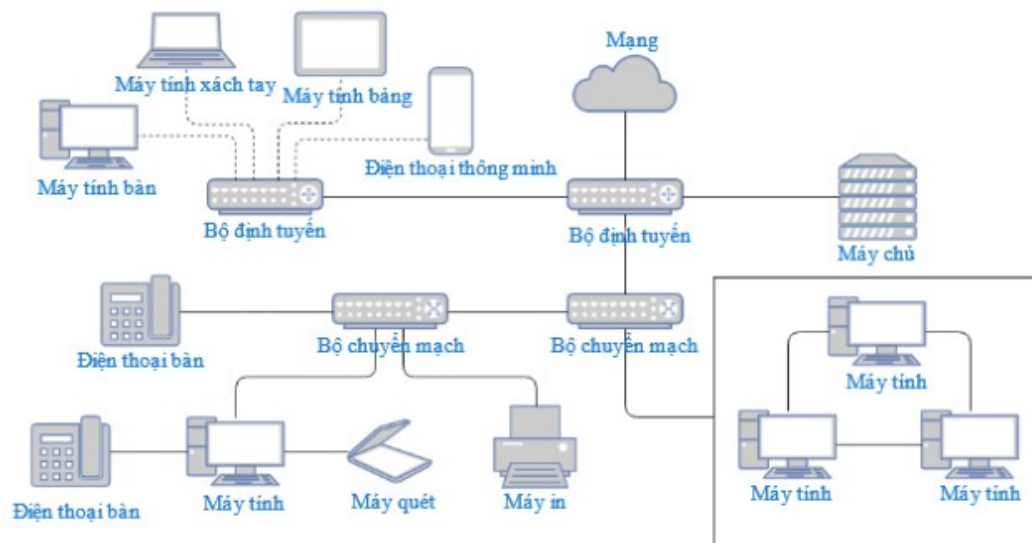
Trước những năm 1970 đã bắt đầu hình thành các máy tính nối với nhau thành mạng và các thiết bị đầu cuối dữ liệu đã kết nối trực tiếp vào máy tính trung tâm để tận dụng tài nguyên chung, khai thác dữ liệu, giảm giá thành truyền số liệu, sử dụng tiện lợi và nhanh chóng hơn. Cùng với thời gian xuất hiện các máy tính Mini Computer và máy tính cá nhân (Personal Computer) đã tăng yêu cầu truyền số liệu giữa máy tính - trạm đầu cuối (Terminal) và ngược lại hình thành nhiều mạng cục bộ, mạng diện rộng trong phạm vi lớn. Do đó mạng máy tính ngày càng được phát triển để đáp ứng với nhu cầu của người sử dụng.

1.1. Định nghĩa mạng máy tính

Mạng máy tính là một nhóm các máy tính, thiết bị ngoại vi được nối kết với nhau thông qua các phương tiện truyền dẫn như cáp, sóng điện từ, tia hồng ngoại... giúp cho các thiết bị này có thể trao đổi dữ liệu với nhau một cách dễ dàng. Khái niệm mạng liên quan đến nhiều vấn đề, bao gồm:

- ☐ Giao thức truyền thông (protocol): Mô tả những nguyên tắc mà tất cả các thành phần mạng cần tuân thủ để có thể trao đổi với nhau
- ☐ Topo (mô hình ghép nối mạng/ hình trạng mạng): Mô tả cách thức nối các thiết bị với nhau.
- ☐ Địa chỉ: Mô tả cách thức định vị một đối tượng trên mạng.
- ☐ Định tuyến (routing): Mô tả cách thức dữ liệu truyền từ thiết bị này sang thiết bị khác trên mạng.
- ☐ Tính tin cậy (reliability): Giải quyết tính toàn vẹn của dữ liệu. đảm bảo dữ liệu nhận được chính xác như dữ liệu gửi đi.
- ☐ Khả năng liên tác (interoperability): Chỉ mức độ các sản phẩm phần mềm và phần cứng của các hãng sản xuất khác nhau có thể làm việc cùng nhau.
- ☐ An ninh (security): Đảm bảo an toàn, hoặc bảo vệ tất cả các thành phần của mạng
- ☐ Chuẩn (standard): Thiết lập các quy tắc và luật lệ cụ thể cần phải tuân theo.

Mạng viễn thông cũng là mạng máy tính. Các node chuyển mạch là hệ thống máy tính được kết nối với nhau bằng các đường truyền dẫn và hoạt động truyền dẫn tuân theo các chuẩn mô hình tham chiếu OSI.



Hình 1.1: Mô hình thể hiện một số thành phần của mạng máy tính

Mạng máy tính bao gồm nhiều thành phần, các thành phần được nối với nhau theo một cách thức nào đó và cùng sử dụng chung một ngôn ngữ;

- **Các thiết bị đầu/cuối** (end system) kết nối với nhau tạo thành mạng có thể là các máy tính (Computer), hoặc các thiết bị khác. Ngày nay có nhiều loại thiết bị có khả năng kết nối vào mạng máy tính như điện thoại di động, máy tính bảng, PDA, tivi,...
- **Môi trường truyền** (media) thực hiện việc truyền dẫn các tín hiệu vật lý. Môi trường truyền có thể là các loại dây dẫn (cáp), song (đối với các mạng không dây). Môi trường truyền là hệ thống các thiết bị truyền dẫn có dây hay không dây dùng để chuyển các tín hiệu điện tử từ máy tính này đến máy tính khác. Các tín hiệu điện tử đó biểu thị các giá trị dữ liệu dưới dạng các xung nhị phân (on – off). Tất cả các tín hiệu được truyền giữa các máy tính đều thuộc một dạng sóng điện từ. Tùy theo tần số của sóng điện từ có thể dùng các môi trường truyền vật lý khác nhau để truyền các tín hiệu. Ở đây môi trường truyền được kết nối có thể là dây cáp đồng trục, cáp xoắn, cáp quang, dây điện thoại, sóng vô tuyến ... Các môi trường truyền dữ liệu tạo nên cấu trúc của mạng. Hai khái niệm môi trường truyền và cấu trúc là những đặc trưng cơ bản của mạng máy tính.
- **Giao thức** (protocol) là quy tắc quy định cách thức trao đổi dữ liệu giữa các thực thể. Việc trao đổi thông tin, cho dù là đơn giản nhất, cũng đều phải tuân

theo những quy tắc nhất định. Việc truyền tín hiệu trên mạng cần phải có những quy tắc, quy ước về nhiều mặt, từ khuôn dạng (cú pháp, ngữ nghĩa) của dữ liệu cho tới các thủ tục gửi, nhận dữ liệu, kiểm soát hiệu quả, chất lượng truyền tin và xử lý các lỗi. Yêu cầu về xử lý và trao đổi thông tin của người sử dụng càng cao thì các quy tắc càng nhiều và phức tạp hơn. Tập hợp tất cả những quy tắc, quy ước đó được gọi là giao thức (Protocol) của mạng. Rõ ràng là các mạng có thể sử dụng các giao thức khác nhau tùy sự lựa chọn của người thiết kế, tuy nhiên các tổ chức chuẩn quốc tế đã đưa ra một số giao thức chuẩn được sử dụng trong nhiều mạng khác nhau để thuận lợi cho việc kết nối chung.

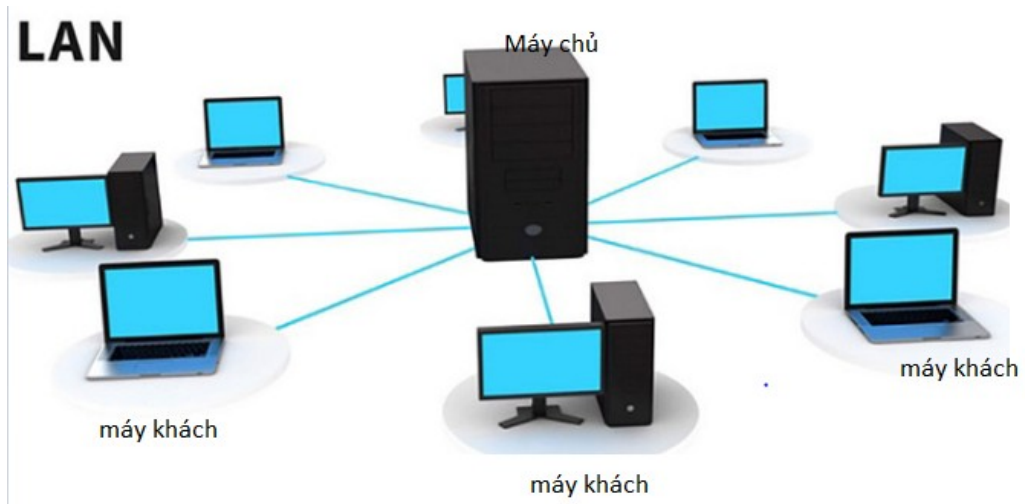
Tốc độ truyền dữ liệu trên đường truyền còn được gọi là **thông** lượng của đường truyền – thường được tính bằng số lượng **bit** được truyền đi trong một giây (bps).

1.1. Phân loại mạng máy tính

Có rất nhiều kiểu mạng máy tính khác nhau. Việc phân loại chúng dựa trên các đặc điểm chung. Ví dụ. mạng máy tính thường được phân loại vùng địa lý (diện tích hoạt động) (ví dụ: mạng cục bộ, mạng diện rộng ...); theo topo (mô hình ghép nối mạng) (ví dụ: point to point hay broadcast), hoặc **hteo** kiểu đường truyền thông **mà** mạng sử dụng.

a. Phân loại mạng theo khoảng cách địa lý

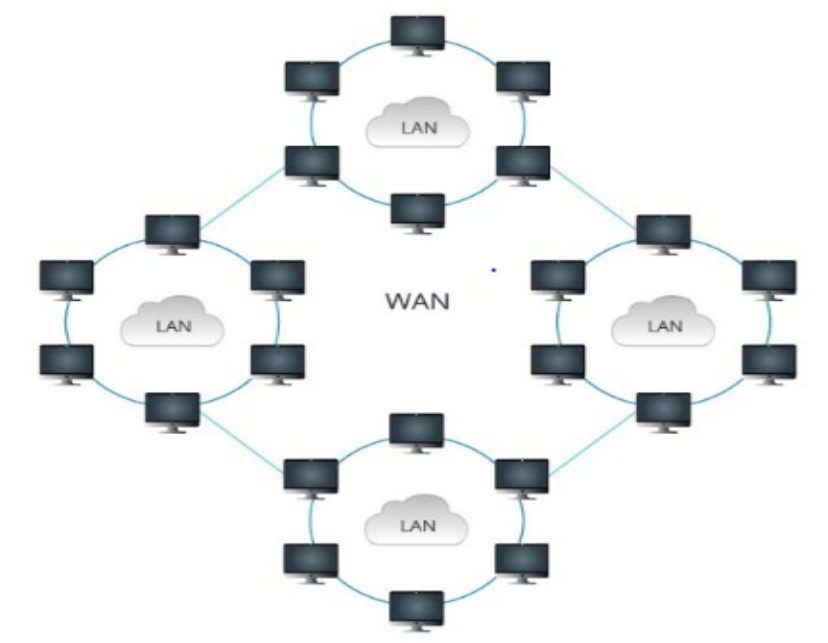
Mạng cục bộ LAN (Local Area Networks): liên kết các tài nguyên máy tính trong một vùng địa lý có kích thước hạn chế. Đó có thể là một phòng, vài phòng trong một tòa nhà, hoặc vài tòa nhà trong một khu nhà. Cụm từ “kích thước hạn chế: không được xác định cụ thể nên người ta xác định phạm vi của mạng LAN bằng cách xác định bán kính nằm trong khoảng vài chục mét đến vài kilômét. Công nghệ truyền dẫn sử dụng trong mạng LAN **thường** là quảng bá (Broadcast), bao gồm một **cáp** đơn nối tất cả các máy. Tốc độ truyền dữ liệu cao, từ 10÷100 Mbps đến hàng trăm Gbps, thời gian trễ nhỏ (cỡ 10μs), độ tin cậy cao, tỷ số lỗi bit từ 10^{-8} đến 10^{-11} .



Hình 1.2: Mô hình mạng LAN

Mạng diện rộng WAN (Wide Area Network) liên kết các tài nguyên máy tính trong một vùng địa **l** rộng (có bán kính trên 100km) như thị xã, thành phố, quốc gia. Có thể coi mạng WAN gồm nhiều mạng LAN khác nhau. Đặc trưng cơ bản của một mạng WAN:

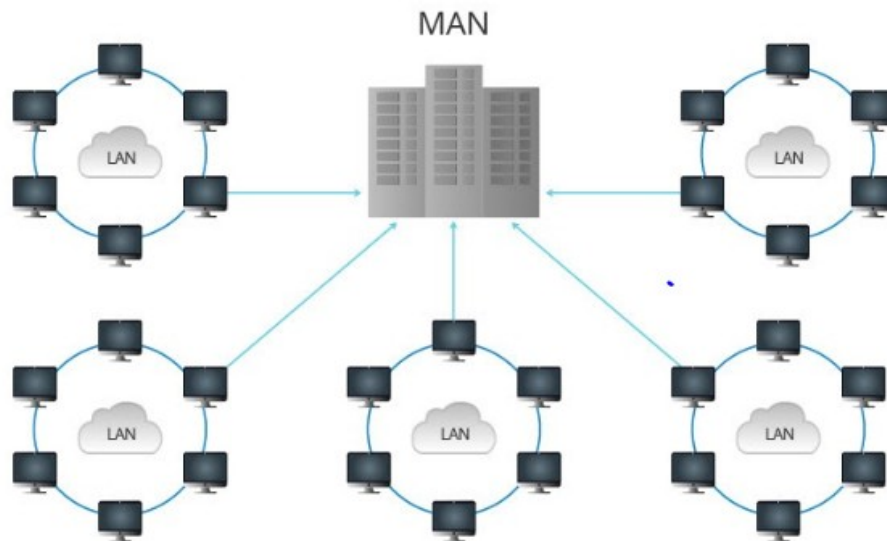
- ☐ Hoạt động trên phạm vi một quốc gia hoặc toàn cầu
- ☐ Tốc độ truyền dữ liệu thấp so với mạng cục bộ
- ☐ Lỗi truyền cao



Hình 1.3: Mô hình mạng WAN

Mạng đô thị MAN (Metropolitan Area Networks): là mạng được cài đặt trong phạm vi một đô thị hoặc một trung tâm kinh tế - xã hội có bán kính tổng

khoảng 100km trở lại. Giả sử có một công ty kinh doanh có nhiều tòa nhà trong thành phố. Mỗi tòa nhà có một mạng LAN riêng, những mạng LAN này được kết nối với nhau, kết quả ta có một mạng MAN. Nhìn chung, mạng MAN được dùng để chỉ các mạng điện thoại có diện tích hoạt động lớn hơn LAN nhưng nhỏ hơn WAN



Hình 1.4: Mô hình mạng MAN

Mạng cá nhân PAN(Person Area Network): là một mạng kết nối các máy tính/thiết bị trong phạm vi của một cá nhân. Vì PAN cung cấp phạm vi mạng trong bán kính quanh một người, thường là khoảng 10 mét (33 feet), nó được gọi là Personal Area Network. Mạng Personal Area Network thường bao gồm máy tính, điện thoại, máy tính bảng, máy in, PDA (Personal Digital Assistant) và các thiết bị giải trí khác như loa, game console, v.v...

Mạng toàn cầu GAN (Global Area Network): phạm vi của mạng trải rộng toàn Trái đất. Việc kết nối các máy tính được thực hiện thông qua mạng viễn thông và vệ tinh. Khoảng cách địa lý có tính chất tương đối đặc biệt trong thời đại ngày nay những tiến bộ và phát triển của công nghệ **truyền** dẫn và quản lý mạng nên ranh giới khoảng cách địa lý giữa các mạng là mờ nhạt.

Tuy nhiên về sau người ta thường quan niệm chung bằng cách đồng nhất 4 loại thành 2 loại sau:

- **WAN** là mạng lớn trên diện rộng, hệ mạng này có thể truyền thông và trao đổi dữ liệu với một phạm vi lớn có **khoảng** cách xa như trong một quốc gia hay quốc tế.

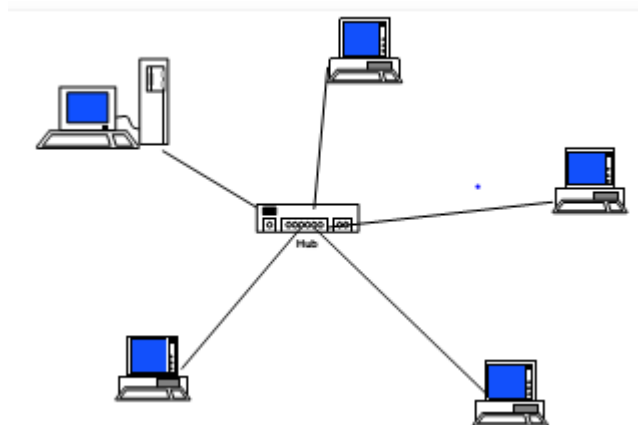
- **LAN** là mạng cục bộ được bố trí trong phạm vi hẹp như một cơ quan, một Bộ, Ngành ... một số mạng LAN có thể nối lại với nhau để tạo thành một mạng LAN lớn hơn

b. Phân loại mạng theo topology

Theo topology, mạng được chia làm các loại như mạng hình sao (Star topology), mạng tuyến tính (Bus topology), mạng vòng (Ring topology) và mạng kết hợp.

- **Mạng hình sao** (Star topology)

Mạng hình sao có tất cả các trạm được kết nối với một thiết bị trung tâm có nhiệm vụ nhận tín hiệu từ các trạm và chuyển đến trạm đích. Tùy theo yêu cầu truyền thông trên mạng mà thiết bị trung tâm có thể là hub, switch, router hay máy chủ trung tâm. Vai trò của thiết bị trung tâm là thiết lập các liên kết Point –to – Point.

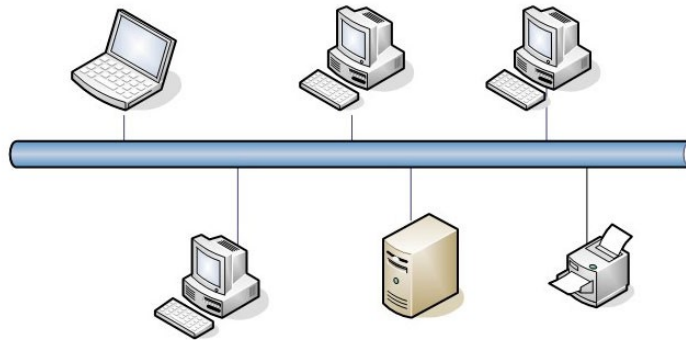


Hình 1.5: Mô hình **mạng** hình sao

- Ưu điểm: Thiết lập mạng đơn giản, dễ dàng cấu hình lại mạng (thêm, bớt các trạm), dễ dàng kiểm soát và khắc phục sự cố, tận dụng được tối đa tốc độ truyền của đường truyền vật lý.
- Khuyết điểm: Độ dài đường truyền nối một trạm với thiết bị trung tâm bị hạn chế (bán kính khoảng 100m với công nghệ hiện nay).
- **Mạng tuyến tính** (Bus topology)

Tất cả các node truy nhập chung trên một đường truyền vật lý được giới hạn hai đầu bằng hai đầu nối đặc biệt gọi là terminator. Mỗi trạm được nối với trục chính (BUS) qua một đầu nối chữ T (T-connector) hoặc một thiết

bị thu phát (transceiver). Chuẩn IEEE 802.3 được gọi là Ethernet, là một **mạng** hình BUS quảng bá với cơ chế điều **khiển** quảng bá động phân tán, trao đổi thông tin với tốc độ 10 Mbps hoặc 100 Mbps

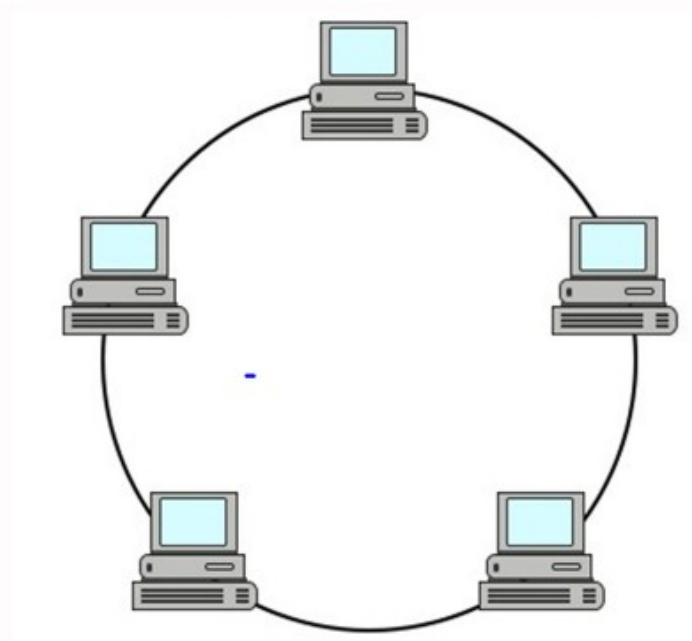


Hình 1.6: Mô hình mạng tuyến tính

- ☐ Ưu điểm: Dễ thiết kế và chi phí thấp, nếu một nút mạng hỏng thì không ảnh hưởng đến hoạt động của toàn mạng.
- ☐ Nhược điểm: Tính ổn định kém, dễ xảy ra xung đột thông tin trên đường truyền.
- ☐ Mạng vòng (Ring topology)

Với mạng vòng (Ring topology) tất cả các node cùng truy nhập chung trên một đường truyền vật lý. Tín hiệu được lưu chuyển trên vòng theo một chiều duy nhất, theo liên kết điểm - điểm. Dữ liệu được chuyển một cách tuần tự từng bit quanh vòng, qua các bộ chuyển tiếp. Bộ chuyển tiếp có ba chức năng: chèn, nhận và hủy bỏ thông tin. Các bộ chuyển tiếp sẽ kiểm tra địa chỉ đích trong các gói dữ liệu khi đi qua nó.

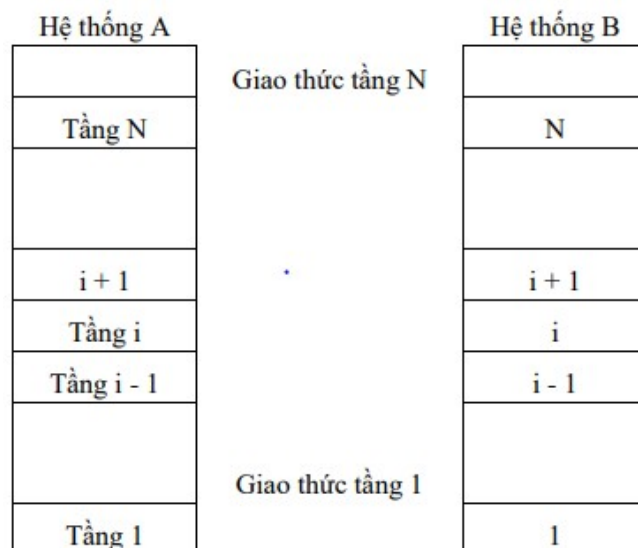
- ☐ Ưu điểm: Với dạng kết nối này có ưu điểm là không tốn nhiều dây cáp, tốc độ truyền dữ liệu cao, không gây ách tắc.
- ☐ Nhược điểm; Các giao thức để truyền dữ liệu phức tạp và nếu có trục trặc trên một trạm thì cũng ảnh hưởng đến toàn mạng



Hình 1.7: Mô hình **mạng** vòng

1.3. Mô hình phân tầng

Để giảm phức tạp của việc thiết kế và cài đặt mạng, hầu hết các mạng máy tính đều có phân tích, thiết kế theo quan điểm phân tầng (layering). Sự phân tầng giao thức rất quan trọng vì nó cung cấp sự hiểu biết sâu sắc về các thành phần giao thức khác nhau cần thiết cho mạng và thuận tiện cho **vệc** thiết kế và cài đặt các phần mềm truyền thống. Mỗi tầng thực hiện một số chức năng xác định và cung cấp một số dịch vụ nhất định cho tầng cao hơn.



Hình 1.8: Kiến trúc phân tầng tổng quát

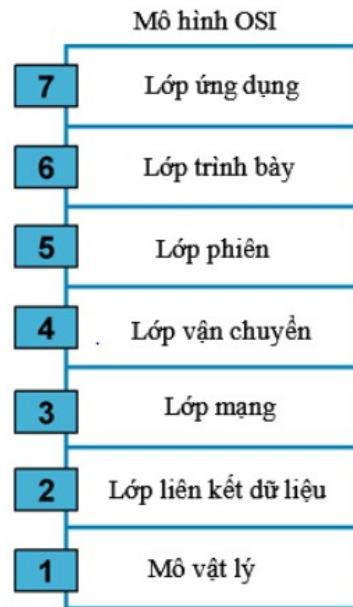
Nguyên tắc phân tầng:

- Giữa 2 tầng liền kề trong một hệ thống giao tiếp với nhau qua 1 giao diện qua đó xác định các hàm nguyên thủy và các dịch vụ tầng dưới cung cấp.
- Giữa hai tầng đồng mức ở hai hệ thống giao tiếp với nhau thông qua các luật lệ, qui tắc được gọi là giao thức.
- Trong thực tế, dữ liệu không được truyền trực tiếp từ tầng thứ i của hệ thống này sang tầng thứ i của hệ thống khác (trừ tầng thấp nhất). Mà việc kết nối giữa hai hệ thống được thực hiện thông qua hai loại liên kết: liên kết vật lý ở tầng thấp nhất và liên kết logic (ảo) ở các tầng cao hơn.

Khi ta nghiên cứu **hoạt** động mạng gồm kết nối Vật lý, giao thức và ứng dụng ta có thể thấy những yếu tố mạng này từ một hệ thống phân cấp các ứng dụng ở trên đỉnh và kết nối ở dưới đáy. Những giao thức cung cấp một cầu nối giữa các ứng dụng và kết nối vật lý. Để hiểu hệ thống phân cấp giữa các yếu tố mạng ta cần một “tiêu chuẩn so sánh” hoặc mô hình xác định những chức năng này. Một mô hình phổ biến nhất là mô hình OSI. Một mô hình khác, mô hình **DoD** (Department of Defense), được thiết kế đặc biệt cho việc mô tả các giao thức TCP/IP.

1.3.1. Mô hình OSI

Mô hình kết nối các hệ thống mở OSI là mô hình căn bản về các tiến trình truyền thông, thiết lập các tiêu chuẩn kiến trúc mạng ở mức Quốc tế, là cơ sở chung để các hệ thống khác nhau có thể liên kết và truyền thông được với nhau. Mô hình OSI tổ chức các giao thức truyền thông thành 7 lớp, mỗi một lớp giải quyết một phần hẹp của tiến trình truyền thông, chia tiến trình truyền thông thành nhiều lớp và trong mỗi lớp có thể có nhiều giao thức khác nhau thực hiện các nhu cầu truyền thông cụ thể.



Hình 1.9: Mô hình OSI

- **Lớp vật lý:** Là tầng thứ nhất trong bảy tầng Mô hình OSI, tầng này chịu trách nhiệm ứng đối với các đòi hỏi về dịch vụ từ Datalink Layer. Chức năng và dịch vụ chính mà tầng vật lý giải quyết là:
 - Thiết lập hoặc ngắt mạch kết nối điện (electrical connection) với một phương tiện truyền thông (transmission medium)
 - Tham gia vào quy trình mà trong đó các tài nguyên truyền thông được chia sẻ hiệu quả giữa nhiều người dùng. Chẳng hạn giải quyết tranh chấp tài nguyên (contention) và điều khiển lưu lượng
 - Điều biến (modulation), hoặc biến đổi giữa biểu diễn dữ liệu số (digital data) của các thiết bị người dùng và các tín hiệu tương ứng được truyền qua kênh truyền thông (communication channel)
- **Lớp liên kết dữ liệu:** Lớp này đảm bảo việc biến đổi các tín dạng bit nhận được từ lớp dưới (vật lý) sang khung số liệu, thông báo cho hệ phát kết quả thu được sao cho các thông tin truyền lên cho tầng Network không có lỗi.

Ví dụ: Giao thức BSC, SDLC, HDLC, LAPB, LAPD.

- **Lớp mạng:** Định rõ các thủ tục cho các chức năng như định tuyến, điều khiển độ lưu lượng, thiết lập cuộc gọi và kết thúc các thông tin người sử dụng mạng lưới, xây dựng dựa trên kiểu kết nối từ nút đến nút do lớp liên kết thông tin cung cấp

Ví dụ: Giao thức IPX, X.25PLP, IP

- **Lớp vận chuyển:** Lớp này duy trì kiểm soát dòng chảy của dữ liệu và thực hiện kiểm tra lỗi và khôi phục dữ liệu giữa các thiết bị. Ví dụ phổ biến nhất của tầng giao vận là Transmission Control Protocol (TCP) và User Datagram Protocol (UDP).

Ví dụ: Giao thức SPX, TCP, UDP.

- **Lớp phiên:** : Định rõ thông tin từ quá trình này đến quá trình kia, khôi phục lỗi, đồng bộ phiên. Lớp phiên có nhiệm vụ thiết lập (và hủy bỏ) một kênh thông tin (đối thoại) giữa hai thực thể giao thức lớp ứng dụng đang thông tin trong một giao dịch mạng đầy đủ.

- **Lớp trình bày:** : liên quan đến việc biểu diễn (cú pháp) của số liệu khi chuyển đi giữa hai tiến trình ứng dụng đang thông tin. Để có được một kết nối các hệ thống mở đúng nghĩa, một số dạng cú pháp số liệu trừu tượng phổ biến được định nghĩa để các tiến trình ứng dụng sử dụng cùng với những cú pháp chuyển số liệu có liên quan. Một chức năng khác của lớp trình bày liên quan đến vấn đề an toàn số liệu..
- **Lớp ứng dụng:** Là mức cao nhất của mô hình OSI, cung cấp phương tiện để người sử dụng có thể truy cập được vào môi trường OSI đồng thời cung cấp dịch vụ thông tin phân tán, thông thường là một chương trình/tiến trình ứng dụng - một loạt các dịch vụ thông tin phân tán trên khắp mạng. Các dịch vụ này bao gồm quản lý và truy cập việc chuyển file, các dịch vụ trao đổi thông báo và tài liệu **chung** như thư tín điện tử.

1.3.2. Mô hình TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) là **chồng** giao thức cùng hoạt động nhằm cung cấp các phương tiện truyền thông liên mạng. Năm 1981, TCP/IP phiên bản 4 (IPv4) được hoàn thành và sử dụng phổ biến trên máy tính sử dụng hệ điều hành UNIX, trở thành một trong những giao thức cơ bản của hệ điều hành Windows 9x. Năm 1994, một phiên bản mới IPv6 được hình thành trên cơ sở cải tiến những hạn chế của IPv4. TCP/IP bao gồm cả các giao thức định hướng mạng và các giao thức **hỗ** trợ ứng dụng. Bởi vì TCP/IP đang được sử dụng rộng rãi với một liên mạng đang tồn tại cho nên rất nhiều giao thức của TCP/IP đã được sử dụng rộng rãi bởi các tổ chức thương mại và các cơ quan Nhà nước để tạo ra các môi trường kết nối hệ thống mở.

Mặc dù có nhiều giao thức trong bộ giao thức truyền thông TCP/IP, hai giao thức quan trọng nhất được lấy tên đặt cho bộ giao thức này là TCP (Transmission Control Protocol) và IP (Internet Protocol). Mô hình TCP/IP được chia thành 4 tầng: tầng truy cập mạng (network access), tầng liên mạng (internet), tầng vận chuyển (transport) và tầng ứng dụng (application),

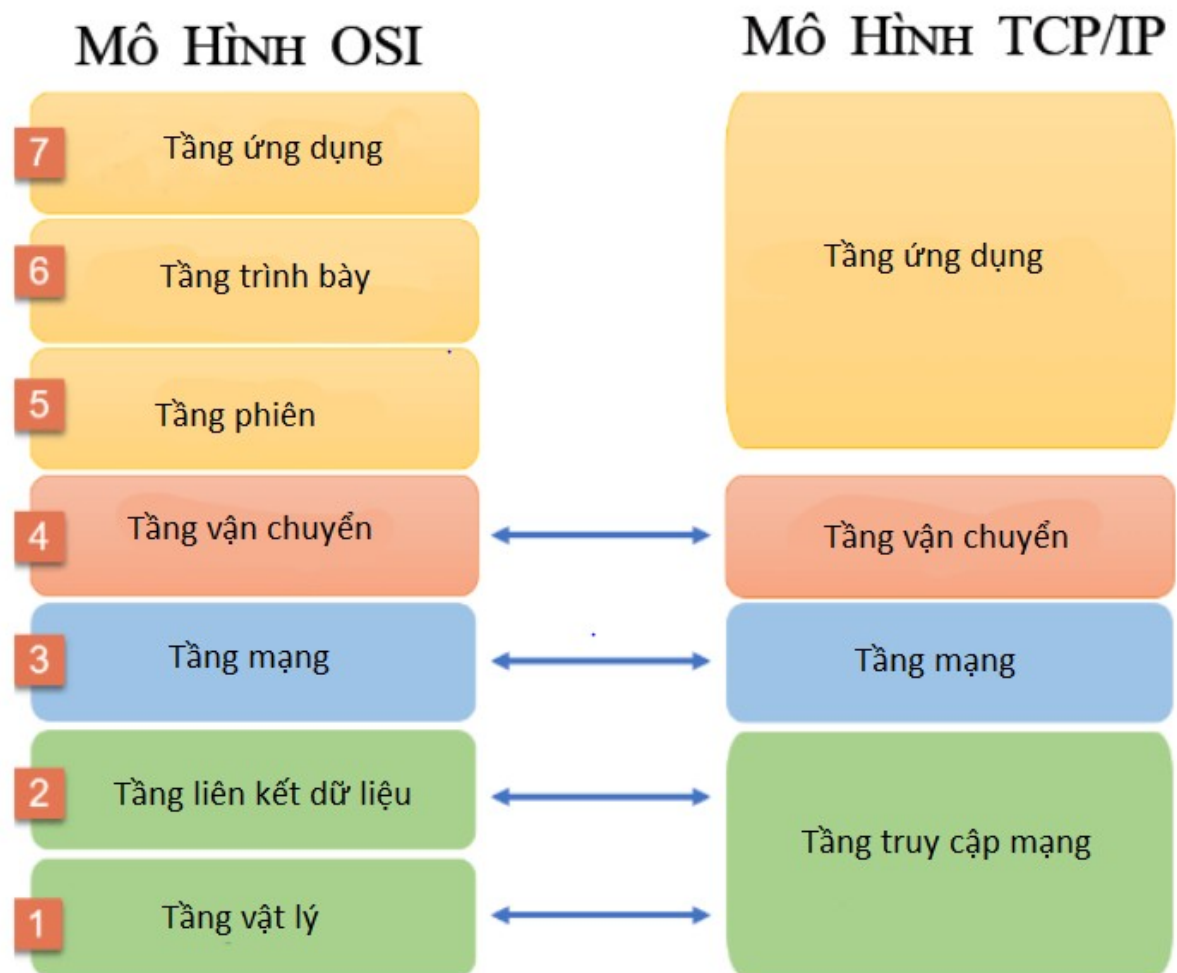
Mô hình TCP/IP



Hình 1.10: Mô hình TCP/IP

- **Tầng ứng dụng** (Application Layer): Ứng với các lớp Session, Presentation và Application trong mô hình OSI. Tầng ứng dụng hỗ trợ các ứng dụng cho các giao thức lớp Host to Host. Cung cấp giao diện cho người sử dụng mô hình TCP/IP. Các giao thức ứng dụng gồm TELNET (truy nhập từ xa), FTP (truyền File), SMTP (thư điện tử).

- **Tầng vận chuyển** (Transport Layer): Ứng với tầng vận chuyển (Transport Layer) trong mô hình OSI, tầng vận chuyển thực hiện những kết nối giữa hai máy chủ trên mạng bằng 2 giao thức: giao thức điều khiển trao đổi dữ liệu TCP (Transmission Control Protocol) và giao thức dữ liệu người sử dụng UDP (User Datagram Protocol). Giao thức TCP là giao thức kết nối hướng liên kết (Connection - Oriented) chịu trách nhiệm đảm bảo tính chính xác và độ tin cậy cao trong việc trao đổi dữ liệu giữa các thành phần của mạng, tính đồng thời và kết nối song công (Full Duplex). Khái niệm tin độ cậy cao nghĩa là TCP kiểm soát lỗi bằng cách truyền lại các gói tin bị lỗi. Giao thức TCP cũng hỗ trợ những kết nối đồng thời. Nhiều kết nối TCP có thể được thiết lập tại một máy chủ và dữ liệu có thể được truyền đi một cách đồng thời và độc lập với nhau trên các kết nối khác nhau. TCP cung cấp kết nối song công (Full Duplex), dữ liệu có thể được trao đổi trên một kết nối đơn theo 2 chiều. Giao thức UDP được sử dụng cho những ứng dụng không đòi hỏi độ tin cậy cao.
- **Tầng mạng** (Internet Layer): Ứng với lớp mạng (Network Layer) trong mô hình OSI, tầng mạng cung cấp một địa chỉ logic cho giao diện vật lý mạng. Giao thức thực hiện của tầng mạng trong mô hình DOD là giao thức IP kết nối không liên kết (Connectionless), là hạt nhân hoạt động của Internet. Cùng với các giao thức định tuyến RIP, OSPF, BGP, lớp mạng IP cho phép kết nối một cách mềm dẻo và linh hoạt các loại mạng "vật lý" khác nhau như: Ethernet, Token Ring, X.25... Ngoài ra tầng này còn hỗ trợ các ánh xạ giữa địa chỉ vật lý (MAC) do lớp Network Access Layer cung cấp với địa chỉ logic bằng các giao thức phân giải địa chỉ ARP (Address Resolution Protocol) và phân giải địa chỉ đảo RARP (Reverse Address Resolution Protocol). Các vấn đề có liên quan đến chuẩn đoán lỗi và các tình huống bất thường liên quan đến IP được giao thức ICMP (Internet Control Message Protocol) thống kê và báo cáo. Tầng trên sử dụng các dịch vụ do tầng Liên mạng cung cấp.
- **Tầng truy nhập mạng** (Network Access Layer): Tương ứng với tầng Vật lý và Liên kết dữ liệu trong mô hình OSI, tầng truy nhập mạng cung cấp các phương tiện kết nối vật lý cáp, bộ chuyển đổi (Transceiver), Card mạng, giao thức kết nối, giao thức truy nhập đường truyền như CSMA/CD, Token Ring, Token Bus...). Cung cấp các dịch vụ cho lớp Internet phân đoạn dữ liệu thành các khung.



Hình 1.11: Mô hình OSI và TCP/IP

1.4. Các giao thức mạng

Internet cung cấp hai giao thức giao vận cho tầng ứng dụng là UDP và TCP. Khi xây dựng ứng dụng cho Internet, một trong những quyết định đầu mà nhà **thieeys** kế phải đưa ra là sử dụng UDP hay TCP. Mỗi giao thức cung cấp một kiểu phục vụ khác nhau cho ứng dụng

1.4.1. Giao thức TCP

Đặc trưng của giao thức TCP là hướng kết nối và cung cấp dịch vụ truyền dữ liệu tin cậy.

- **Hướng kết nối** (Connection oriented): TCP client và TCP server trao đổi các thông tin điều khiển với nhau trước khi truyền dữ liệu ứng dụng. Quá trình “bắt tay” giữa client và server như vậy cho phép cả hai bên sẵn sàng xử lý các gói dữ liệu. Sau quá trình này, xuất hiện một **đường** kết nối TCP (TCP connection) giữa socket của hai tiến trình. Đây là kết nối hai chiều (song công – full duplex) vì cho phép hai **tiền trình** có thể **đồng** thời gửi và nhận dữ liệu. Khi ứng dụng kết thúc việc gửi thông điệp, nó đóng kết nối lại. Dịch vụ này chỉ là hướng kết nối chứ không phải **mạch** ảo (virtual circuit), bởi vì hai tiến trình được kết nối một cách lỏng lẻo.
- Dịch vụ giao vận tin cậy: Tiến trình gửi có thể sử dụng TCP để truyền dữ liệu chính xác và đúng thứ tự. Gửi đi một luồng byte qua socket, tiến trình ứng dụng có thể tin tưởng TCP sẽ chuyển luồng byte này đến socket nhận, không bị lỗi hay trùng lặp byte.

TCP cũng có cơ chế kiểm soát tắc nghẽn, cơ chế này đáp ứng cho cả Internet chứ không phải cho hai tiến trình truyền thông với nhau. Kỹ thuật kiểm soát tắc nghẽn của TCP là giảm tốc độ gửi dữ liệu của mỗi tiến trình (client hay server) khi mạng bị tắc nghẽn.

Giới hạn tốc độ truyền có thể không **thỏa** mãn với các ứng dụng audio và video theo thời gian thực, những ứng dụng đòi hỏi phải có một băng thông tối thiểu. Hơn nữa, ứng dụng thời gian thực chấp nhận **mất** mất dữ liệu và không thực sự cần đến một dịch vụ giao văn tin cậy hoàn toàn. Vì các lý do đó, các ứng dụng thời gian thực thường chạy trên nền UDP.

Một số dịch vụ mà TCP không cung cấp. Thứ nhất, TCP không bảo đảm một tốc độ truyền tối thiểu. Tiến trình gửi không được phép truyền với bất kỳ tốc độ **nào nó** đề nghị, tốc độ này được kiểm soát bởi cơ chế kiểm soát tắc **nghiên** của TCP. Đôi khi cơ chế này khiến tiến trình **giới** phải gửi với các độ trung bình trong đôi thấp. Thứ hai, TCP không dành ra bất kỳ sự bảo đảm nào về độ trễ. Khi tiến trình gửi chuyển dữ liệu cho socket TCP, dữ liệu cuối cùng sẽ đến được socket nhận, nhưng TCP không bảo đảm dữ liệu sau bao lâu mới tới được đích. Với những quan sát trên môi trường Internet thực, có thể phải chờ vài giây, thậm chí đến vài phút để TCP gửi thành công một thông điệp (ví dụ, một trang Web HTML từ Web server đến Web client). Nói tóm lại, TCP bảo đảm việc truyền tất cả dữ liệu một cách chính xác, nhưng không bảo đảm về tốc **độ** truyền và độ trễ.

1.4.2. Giao thức UDP

UDP là giao thức giao vận khá đơn giản, với mô hình phục vụ tối thiểu. UDP không hướng kết nối, nghĩa là không có giai đoạn "bắt tay" trước khi hai tiến trình bắt đầu trao đổi dữ liệu. UDP không cung cấp dịch vụ truyền tin cậy. Khi tiến trình gửi **chuyển** thông điệp qua cổng UDP, UDP không đảm **bảo** thông điệp sẽ đến được cổng tiến trình nhận. Hơn nữa, các thông điệp đến đích có thể không đúng thứ tự.

Mặt khác, UDP không có cơ chế kiểm soát tắc nghẽn, vì vậy tiến trình gửi có thể đẩy dữ liệu ra cổng UDP với tốc độ bất kỳ. Mặc dù, không phải Tất cả dữ liệu đều **tới** được đích, nhưng phần lớn dữ liệu có thể tới được. Ứng dụng thời gian thực thường lựa chọn UDP ở tầng giao vận. Giống TCP. UDP không bảo đảm về độ trễ.

Bảng 1.1 trình bày các giao thức giao vận của các ứng dụng mạng phổ biến. Thư điện tử, truy cập từ xa. Web và truyền file sử dụng TCP do TCP cung cấp dịch vụ truyền dữ liệu tin cậy, bảo đảm rằng mọi dữ liệu sẽ tới được đích

Người ta thấy rằng, điện thoại qua Internet chạy trên nền UDP, mỗi phía của ứng dụng này cần gửi dữ liệu qua mạng với tốc độ tối thiểu nào đó (Bảng 1,1). Hơn nữa, ứng dụng điện thoại qua Internet chấp nhận mất mát dữ liệu. Vì thế chúng không cần dịch vụ truyền tin cậy của TCP.

Ứng dụng	Giao thức ứng dụng	Giao thực giao vận
Thư điện tử	SMTP [RFC 821]	TCP
Truy cập từ xa	Telnet [RFC 854]	TCP
Web	HTTP [TEC 2068]	TCP
Truyền File	FTP [RFC 959]	TCP
Remote File Server	NFS	UDP hoặc TCP
Streaming Multimedia	Giao thức riêng, không công bố (ví dụ Real NetWork)	UDP hoặc TCP
Điện thoại Internet	Giao thức riêng, không công bố (ví dụ Vocaltec)	Thường là UDP

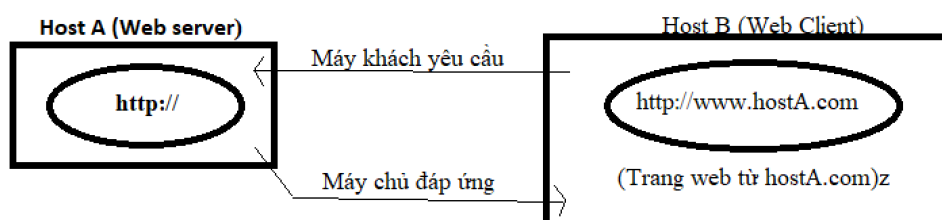
Bảng 1.1. Các ứng dụng phổ biến và giao thức giao vận tương ứng

1.5. Mô hình ứng dụng mạng

1.5.1. Mô hình Khách – Máy chủ (Client – Server)

Mô hình Client/Server mô tả các dịch vụ mạng và các ứng dụng được sử dụng để truy nhập các dịch vụ. Là mô hình phân chia các thao tác thành hai phần: phía Client cung cấp cho người sử dụng một giao diện để yêu cầu dịch vụ từ mạng và phía Server tiếp nhận các yêu cầu từ phía Client và cung cấp các dịch vụ một cách thông suốt cho người sử dụng.

Chương trình Server được khởi động trên một máy chủ và ở trạng thái sẵn sàng nhận các yêu cầu từ phía Client. Chương trình Client cũng được khởi động một cách độc lập với chương trình Server. Yêu cầu dịch vụ được chương trình Client gửi đến máy chủ cung cấp dịch vụ và chương trình Server trên máy chủ sẽ đáp ứng các yêu cầu của Client. Sau khi thực hiện các yêu cầu từ phía Client, Server sẽ trở về trạng thái chờ các yêu cầu khác.



Hình 1.12. Mô hình Khách – Máy chủ (Client – Server)

Trong mô hình Client/Server nhiều lớp, quá trình xử lý được phân tán trên 3 lớp khác nhau với các chức năng riêng biệt. Mô hình này thích hợp cho việc tổ chức hệ thống thông tin trên mạng Internet/ Intranet. Phát triển mô hình 3 lớp sẽ khắc phục được một số hạn chế của mô hình 2 lớp. Các hệ cơ sở dữ liệu được cài đặt trên các máy chủ Web Server và có thể được truy nhập không hạn chế các ứng dụng và số lượng người dùng.

Lớp khách (Clients) cung cấp dịch vụ trình bày (Presentation Services), giao tiếp người sử dụng với lớp giao dịch thông qua trình duyệt Browser hay trình ứng dụng để thao tác và xử lý dữ liệu. Giao diện người sử dụng là trình duyệt Internet Explorer hay Netscape.

Lớp giao dịch (Business) cung cấp các dịch vụ quản trị, tổ chức và khai thác cơ sở dữ liệu. Các component trước đây được cài đặt trên lớp khách, nay được cài đặt trên lớp giao dịch. Ví dụ, một người sử dụng trên máy khách đặt mua hàng, lớp giao dịch kiểm tra mã mặt hàng để quyết định tiếp tục bán hay không bán. Thành phần của lớp giao dịch trong mô hình Internet là Web Server và COM+/MTS. Công nghệ của Microsoft với Web Server là IIS (Internet Information Services) sử dụng ASP để kết nối Client với COM. Web Server giao tiếp với COM+/MTS component qua COM.

COM+/MTS component điều khiển tất cả giao tiếp với lớp dữ liệu nguồn thông qua ODBC hoặc OLE - DB.

Lớp nguồn dữ liệu (Data Source) cung cấp các dịch vụ tổ chức và lưu trữ các hệ cơ sở dữ liệu quan hệ. Sẵn sàng cung cấp dữ liệu cho lớp giao dịch. Đặc trưng của lớp này là ngôn ngữ tìm kiếm, truy vấn dữ liệu SQL.

1.5.2. Mô hình ngang hàng (Peer to peer)

Trong mô hình ngang hàng tất cả các máy đều là máy chủ đồng thời cũng là máy khách. Các máy trên mạng chia sẻ tài nguyên không phụ thuộc vào nhau. Mạng ngang hàng thường được tổ chức thành các nhóm làm việc Workgroup. Mô hình này không có quá trình đăng nhập tập trung, nếu đã đăng nhập vào mạng có thể sử dụng tất cả tài nguyên trên mạng. Truy cập vào các tài nguyên phụ thuộc vào người đã chia sẻ các tài nguyên đó, vì vậy có thể phải biết mật khẩu để có thể truy nhập được tới các tài nguyên được chia sẻ.

1.5.3. Mô hình lai (Hybrid)

Đây là mô hình kết hợp giữa Client-Server và Peer-to-Peer. Phần lớn các mạng máy tính trên thực tế thuộc mô hình này.

1.6. Tổng quan về ngôn ngữ python

1.6.1. Giới thiệu ngôn ngữ Python

Là ngôn ngữ lập trình bậc cao, phục vụ cho các mục đích lập trình đa năng. Ưu điểm nổi bật nhất đó chính là dễ đọc, dễ nhớ, dễ học. Python là ngôn ngữ có cấu trúc tương đối rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của python còn cho phép người dùng sử dụng để viết mã lệnh với số lần gõ phím tối thiểu.

Thậm chí trong năm 2021, Python được đánh giá là Top 2 ngôn ngữ lập trình nên học và top 1 ngôn ngữ lập trình được yêu thích nhất. Chính vì thế, Python trở thành lựa chọn hàng đầu cho người mới, hay thậm chí là các lập trình viên chuyên nghiệp

1.6.2. Các giai đoạn phát triển của Python

Để có được một ngôn ngữ lập trình hữu ích như hiện nay, Python đã phải trải qua nhiều giai đoạn phát triển của mình. Quá trình phát triển của Python gồm có 3 giai đoạn:

- Giai đoạn Python 1 – phiên bản 1.x:

Vào đầu năm 1991, lập trình viên vĩ đại **tại** người Ba Lan – Guido Van Rossum đã khai sinh ra ngôn ngữ lập trình Python cơ bản. Vào thời điểm đó,

Python được tạo ra dùng để chạy trên hệ điều hành Unix. Phiên bản tối ưu nhất trong giai đoạn này là 1.6.1

□ Giai đoạn 2 – Phiên bản 2.x:

Năm 2000, Rossum và đội ngũ nghiên cứu Python chuyển đi và thành lập nên BeOpen Python Labs Team và cho ra mắt phiên bản Python 2.0. Tuy nhiên, đến phiên bản 2.1 thì ngôn ngữ này thuộc về quyền sở hữu của Python Software Foundation. Đến năm 201 thì Python chính thức ngừng cập nhật tại phiên bản 2.7.

□ Giai đoạn 3 – Phiên bản 3.x:

Python 3.0 ra đời năm 2008 với sứ mệnh cải thiện và loại bỏ những điểm **trùng lặp** trong cấu trúc và câu lệnh của phiên bản 2.x. Khi sử dụng sẽ có những tiện ích giúp chuyển đổi 2.x sang 3.x một cách thuận tiện.

1.6.3. Các tính năng nổi bật của ngôn ngữ python

- **Miễn phí, mã nguồn mở:** Có thể thỏa mái sử dụng và phân phối python thậm chí là có thể sử dụng chúng để phục vụ cho mục đích thương mại. Bởi chúng là mã nguồn mở, ta không chỉ sử dụng các phần mềm, chương trình được viết trong python mà còn có thể thay đổi mã nguồn. Python có một cộng đồng lớn, thường xuyên cập nhật, không ngừng cải tiến.
- **Ngôn ngữ lập trình đơn giản, dễ đọc:** Python có cấu trúc ngữ pháp đơn giản, rõ ràng. Nó dễ đọc và viết đơn giản hơn nhiều khi so sánh với ngôn ngữ lập trình khác như C ++, Java, C#. Python làm cho việc lập trình trở nên thú vị, giúp các lập trình viên tập trung vào những giải pháp chứ không phải là cú pháp.
- **Khả năng di chuyển:** Các chương trình trên python có thể di chuyển từ nền tảng này đến nền tảng khác mà không gặp phải bất kỳ thay đổi nào khi chạy. Nó chạy liền mạch trên các nền tảng như Mac, Windows, Linux.
- **Khả năng mở rộng và có thể nhúng:** Nếu một ứng dụng đòi hỏi sự phức tạp lớn, ta có thể dễ dàng kết hợp với các phần code bằng C, C ++ và những ngôn ngữ khác vào code python. Bởi vậy, sẽ giúp ứng dụng của ta có những tính năng tốt hơn, khả năng scripting mà các ngôn ngữ lập trình khác khó có thể làm được.
- **Hướng đối tượng:** Mọi thứ của python đều hướng đối tượng. Lập trình đối tượng sẽ giúp giải quyết các vấn đề một cách trực quan nhất. Với lập trình đối tượng, ta có thể phân chia nhiều phức tạp thành các tập nhỏ hơn bằng cách tạo ra các đối tượng.
- **Thư viện tiêu chuẩn lớn:** Ngôn ngữ python có thư viện lớn giúp cho việc lập trình trở nên dễ dàng hơn vì không phải tự viết tất cả các code. Những thư viện này được kiểm tra kỹ lưỡng, nên chắc chắn nó không làm hỏng code hay ứng dụng nào của ta.

1.6.4. Các thành phần và cú pháp cơ bản trong chương trình Python

a. Từ khóa (keyword) trong python

Từ khóa (keyword) là những từ (word) được dành riêng trong Python. Ta không thể sử dụng từ khóa để đặt tên biến, tên hàm hoặc bất kỳ định danh (identifier) nào khác. Chúng được sử dụng để xác định cú pháp và cấu trúc của ngôn ngữ Python. Trong Python, các từ khóa có sự phân biệt chữ hoa và chữ thường.

Với phiên bản Python 3.10.3 hiện tại em đang sử dụng có tất cả 35 từ khóa. Tất cả các từ khóa ngoại trừ True, False và None đều ở dạng chữ thường. Chúng ta có thể sử dụng lệnh help("keywords") trong **trình** thông dịch Python để xem danh sách tất cả các từ khóa trong Python.

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
And	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Bảng 1.2. Bảng các keyword trong Python 3.10.3

b. Định danh (identifier) trong Python

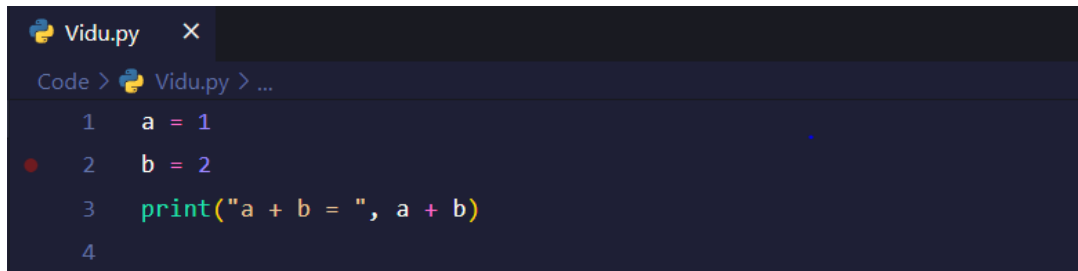
Định danh (identifier) là tên được đặt cho các thực thể như lớp, hàm, biến,... Định danh giúp phân biệt thực thể này với thực thể khác.

Những quy tắc khi đặt tên định danh:

- ☐ Tên định danh có thể bao gồm các chữ thường (a đến z), chữ hoa (A đến Z), chữ số (0 đến 9), dấu gạch dưới `_`. Ví dụ, `myClass`, `var_1` và `print_this_to_screen` là các tên định danh hợp lệ.
- ☐ Tên định danh không được bắt đầu bằng một chữ số. Ví dụ, tên định danh `1variable` không hợp lệ nhưng `variable1` thì hợp lệ.
- ☐ Không được đặt tên định danh giống với từ khóa (keyword).
- ☐ Không được sử dụng các ký hiệu đặc biệt như `!`, `@`, `#`, `$`, `%`,... trong tên định danh.
- ☐ Tên định danh có thể có độ dài bất kỳ.

c. Câu lệnh (statement) trong Python

Python sẽ thông dịch từng câu lệnh (statement) để thực thi. Một statement trong Python thường được viết trong 1 dòng. Ta không cần thiết phải thêm dấu chấm phẩy `;` vào cuối mỗi câu lệnh. Ví dụ:



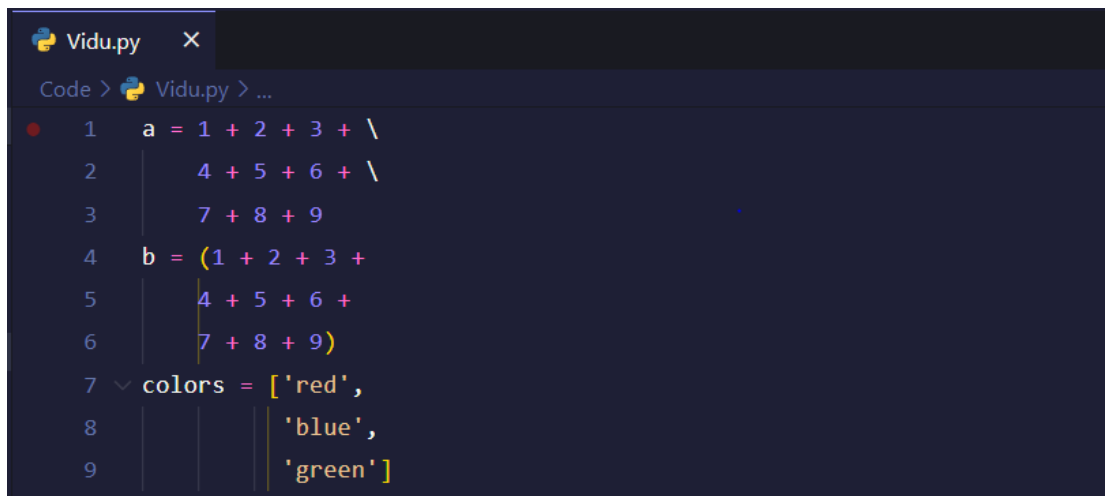
```

Vidu.py
Code > Vidu.py > ...
1 a = 1
2 b = 2
3 print("a + b = ", a + b)
4

```

Hình 1.13. Ví dụ về câu lệnh trong Python

Ta có thể viết một câu lệnh trên nhiều dòng bằng cách sử dụng thích hợp các ký tự như ký tự tiếp tục (\), dấu ngoặc đơn (), ngoặc vuông [], ngoặc nhọn {}.



```

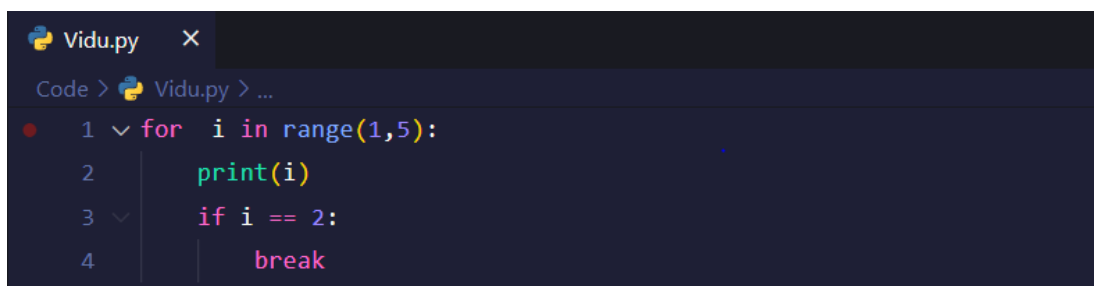
Vidu.py
Code > Vidu.py > ...
1 a = 1 + 2 + 3 + \
2     4 + 5 + 6 + \
3     7 + 8 + 9
4 b = (1 + 2 + 3 +
5     4 + 5 + 6 +
6     7 + 8 + 9)
7 colors = ['red',
8           'blue',
9           'green']

```

Hình 1.14. Ví dụ về câu lệnh nhiều dòng trong Python

d. Thụt đầu dòng (indentation) trong python

Python sử dụng thụt đầu dòng (indentation) để định nghĩa một khối lệnh (code block) như thân hàm, thân vòng lặp,... Lưu ý: Python không sử dụng dấu ngoặc nhọn {} cho code block như các ngôn ngữ C/C++, Java,... Ví dụ:



```

Vidu.py
Code > Vidu.py > ...
1 for i in range(1,5):
2     print(i)
3     if i == 2:
4         break

```

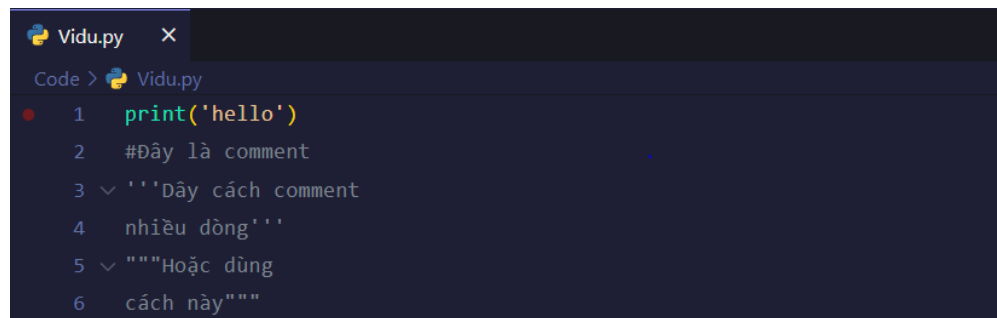
Hình 1.15. Ví dụ về thụt đầu dòng trong python

Trong python, 4 khoảng trắng được sử dụng cho thụt đầu dòng và được ưu tiên hơn các tab. Nếu sử dụng indentation không chính xác thì chương trình sẽ báo lỗi *IndentationError*.

e. Ghi chú (comment) trong python

Ghi chú (comment) được sử dụng để giải thích code đang thực hiện những gì. Việc này rất quan trọng khi đọc lại source code, bảo trì chương trình sau này. Sử dụng ký hiệu hash (#) để bắt đầu viết comment trong Python. Trình thông dịch Python sẽ bỏ qua comment bắt đầu từ ký hiệu hash (#) cho đến khi gặp ký tự bắt đầu dòng mới.

Ta có thể viết comment trên nhiều dòng, mỗi dòng bắt đầu bằng ký tự hash(#), Hoặc một cách khác để comment trên nhiều dòng bằng cách sử dụng dấu ''' hoặc """. Ví dụ:

A screenshot of a code editor window titled 'Vidu.py'. The editor shows six lines of Python code. Line 1: `print('hello')`. Line 2: `#Đây là comment`. Line 3: `'''Đây cách comment`. Line 4: `nhiều dòng'''`. Line 5: `"""Hoặc dùng`. Line 6: `cách này"""`. The code is displayed in a dark-themed editor with syntax highlighting. The first line is a standard Python print statement. The second line is a single-line comment starting with a hash symbol. The third and fourth lines show a multi-line comment using single quotes. The fifth and sixth lines show a multi-line comment using double quotes.

Hình 1.16. Ví dụ về comment trong Python

f. **Khối lệnh** (code block) trong Python

Một hoặc nhiều câu lệnh (statement) có thể tạo thành một khối lệnh (code block). Các khối lệnh thường gặp trong Python như các lệnh trong lớp (class), hàm (function), vòng lặp (loop),... Python sử dụng thụt đầu dòng (indentation) để bắt đầu định nghĩa, phân tách một code block với các code block khác. Ví dụ:



```
1  import sys
2
3  file = "text.txt"
4  try:
5      myFile = open(file, "r")
6      myLine = myFile.readline()
7      while myline:
8          print(myline)
9          myline = myFile.readline()
10     myFile.close()
11 except IOError as e:
12     print("I/O error({0}): {1}".format(e.errno, e.strerror))
13 except:
14     print("Unexoected error: ", sys.exc_info()[0])
15 print("done")
```

The image shows a Python code editor window titled 'Vidu.py'. The code is a script to read a file 'text.txt'. It uses a try-except block to handle potential I/O errors. Inside the try block, it opens the file, reads a line, and enters a while loop to print each line until the end of the file is reached. The code is annotated with Vietnamese text: 'khởi lệnh chính' (main command block) for the try block, 'khởi lệnh thân try' (try block body) for the file opening and reading, and 'khởi lệnh thân while' (while block body) for the line printing and reading loop. The code ends with a print statement and an exception handling section for IOError and other exceptions.

Hình 1.17. Ví dụ về khối lệnh trong Python

CHƯƠNG 2: LẬP TRÌNH SOCKET

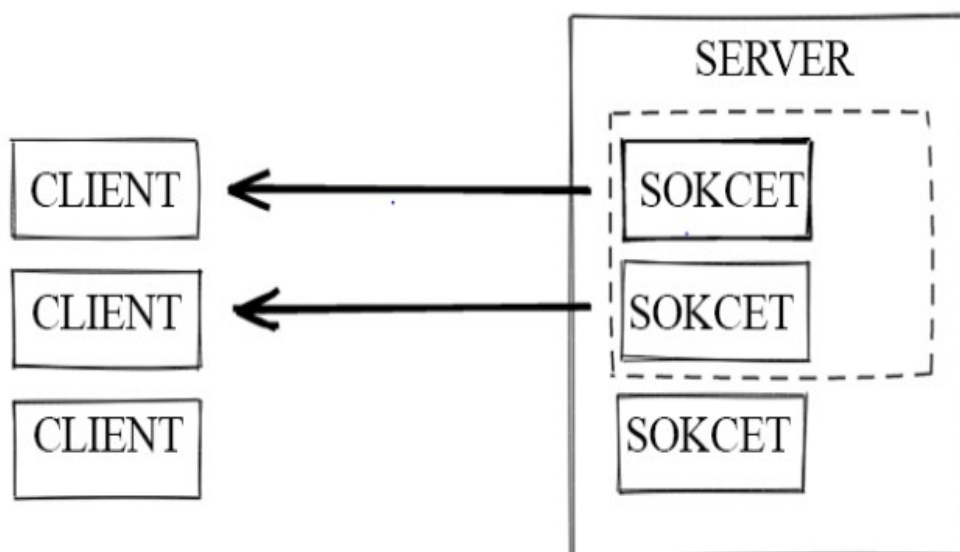
2.1. Tìm hiểu về Socket

Socket là một giao diện lập trình (API – Application Program Interface) ứng dụng mạng thông qua giao diện này có thể lập trình điều khiển việc truyền thông giữa 2 máy sử dụng các giao thức mức thấp như TCP,UDP... Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều, hay còn gọi là two-way communication để kết nối 2 process trò chuyện với nhau. Điểm cuối (endpoint) của liên kết này được gọi là Socket.

Ưu điểm lớn nhất khiến socket trở nên cần thiết là nó có thể tương thích với hầu hết các hệ điều hành, từ Windows, Linux cho đến Mac OS X... Ngoài ra, socket còn có thể kết hợp được với rất nhiều ngôn ngữ lập trình như: C, C++, Java, Visual Basic, Visual C++... Có thể thấy rằng socket phù hợp để sử dụng ở mọi điều kiện cũng như hoàn cảnh khác nhau.

Đặc biệt là lập trình viên có thể chạy song song nhiều socket trong cùng một lúc. Điều này giúp rút ngắn thời gian và tăng hiệu suất hoạt động.

Cách thức hoạt động của Socket: Thông qua TCP/IP và UDP, socket sẽ tiến hành truyền và nhận dữ liệu Internet. Từ đó tạo nên một cầu nối giữa client và server. Điều kiện để hoạt động này diễn ra là có đủ thông tin về thông số IP và **giữ** liệu cổng của 2 process muốn kết nối với nhau. Hai ứng dụng muốn liên kết có thể nằm cùng trên 1 máy hoặc 2 máy khác nhau đều được. Tuy nhiên, nếu 2 process cùng nằm trên một máy, các số liệu bắt buộc phải khác nhau hoàn toàn. Đây là điều kiện cần thiết để socket io có thể tiến hành hoạt động.



Hình 2.1. Cách hoạt động của Socket

2.2. Phân loại Socket

2.2.1. Stream Socket

Stream Socket còn được gọi là socket TCP. Nó hoạt động dựa trên giao thức hướng kết nối. Tức là chỉ sử dụng được khi máy chủ và máy khách có sự liên kết.

Stream Socket giúp quá trình truyền gửi diễn ra nhanh chóng và đúng hạn. Ngoài ra, với yếu tố đảm bảo, người dùng cũng có thể yên tâm rằng dữ liệu sẽ được chuyển đến đúng người nhận với độ tin cậy tuyệt đối. Mỗi hành động diễn ra trong quá trình dịch chuyển thông tin đều sẽ được ghi lại kết quả và truyền về cho người dùng dù có thành công hay không. Các bản ghi dữ liệu cũng không hề có giới hạn nào, bạn có thể thoải mái truyền bao nhiêu thông tin tùy thích. Song song với đó, Stream Socket còn sở hữu 2 cơ chế bao gồm quản lý luồng lưu thông trên mạng và chống tắc nghẽn nhằm tối ưu hóa thời gian truyền dữ liệu.

Điều kiện để sử dụng Stream Socket là phải có địa chỉ IP rõ ràng giữa 2 đầu kết nối. Các thông tin được gửi đi tuân tự theo kế hoạch lên trước. Mỗi thông điệp được thực hiện phải có thông báo trả về mới tính là hoàn thành. Ngoài ra, Stream Socket hoạt động dựa trên mô hình lắng nghe và chấp nhận. Có nghĩa rằng giữa 2 process phải có 1 bên yêu cầu kết nối trước.

2.2.2. Datagram socket

Datagram Socket hoạt động dựa trên giao thức UDP về việc truyền thông tin không yêu cầu sự kết nối. Để hoạt động này diễn ra, nó cung cấp connection-less point cho việc gửi và nhận thông tin. Chính vì thế mà Datagram Socket còn được gọi là socket không hướng kết nối.

Hai tiến trình có thể liên lạc với nhau thông quan Datagram Socket mà không cần IP chung. Thông điệp muốn gửi đi phải kèm theo thông điệp người nhận. Có thể gửi một thông điệp nhiều lần, tuy nhiên không thể gửi cùng một lúc. Ngoài ra, thứ tự hoàn thành dịch chuyển cũng không cố định, thông điệp gửi sau có thể đến trước và ngược lại.

Datagram Socket không đảm bảo tuyệt đối kết quả của tiến trình. Một số trường hợp ghi nhận thông điệp không thể đến tay của bên nhận. Cùng với đó, điều kiện để thực hiện các cuộc trao đổi 2 đầu là 1 trong 2 tiến trình phải công bố port của socket mà mình đang sử dụng.

Tuy nhiên vì không yêu cầu kết nối của 2 tiến trình nên quá trình truyền dữ liệu diễn ra vô cùng nhanh chóng, phù hợp để ứng dụng trong cách hoạt động như nhắn tin, chat game online...

2.2.3. Unix socket

Unix socket được biết đến như một điểm chuyển giao giữa các ứng dụng ở trong một máy tính. Vì không phải qua bước kiểm tra và routing nên quá trình truyền tin diễn ra vô cùng nhẹ nhàng và nhanh chóng. Đường chuyển khép kín đảm bảo không bị rò rỉ thông tin khi thực hiện.

Unix socket mang đến những ưu điểm tuyệt vời như: tăng tốc độ truy cập MySQL lên đến 30-50%, tăng PostgreSQL lên hơn 30%, tăng Redis lên 50%. Cùng với đó còn giảm thời gian latency xuống từ 60ms còn 5ms.

Bên cạnh đó, Unix socket vẫn còn một số nhược điểm tồn đọng như: không thể dịch chuyển giữa 2 máy khác nhau, đôi khi xảy ra delay do vấn đề phân quyền giữa các tệp tin.

2.2.4. Web socket

Không giống như 3 loại socket trên, Websockets được sử dụng nhiều nhất nhờ những ứng dụng to lớn mà nó mang lại.

Vậy websocket là gì? Websocket là một module hỗ trợ kết nối giữa hai đầu máy nhờ giao thức TCP mà không cần quan HTTP. Websocket được thiết kế chuyên dụng dành cho web nhưng vẫn có thể được dùng để ứng dụng cho các phần mềm.

Websocket sở hữu gần như hầu hết những ưu điểm của các loại socket khác như: tỷ lệ xảy ra delay thấp, dễ xử lý lỗi, khả năng dịch chuyển thông tin nhanh chóng và mạnh mẽ, phù hợp cho những hoạt động cần đến tính tức thời như chat realtime, chat online, biểu đồ chứng khoán...