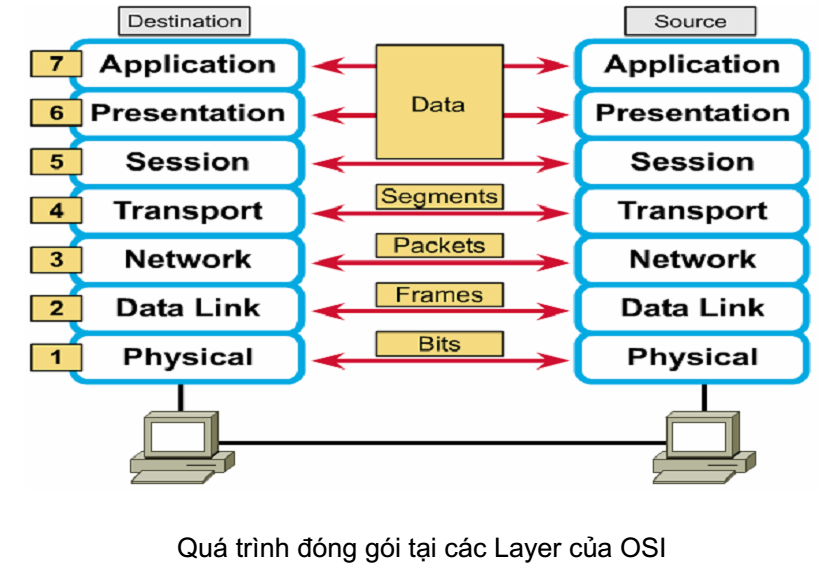
1. Mô hình OSI
   1. Mục đích của mô hình OSI

Mô hình OSI đưa ra nhằm:

* Cách thức cho các thiết bị mạng có thể truyền dữ liệu được với nhau.
* Cách thức khi nào thiết bị khi nào thiết bị được truyền dữ liệu khi nào không được truyền dữ liệu.
* Phương pháp đảm bảo mức độ tin cậy, tốc độ truyền dữ liệu.
* Cách thức đảm bảo các thiết bị mạng duy trùy tốc độ truyền dữ liệu thích hợp.
* Cách thức thiết lập kết nối, truyền và sắp xếp dữ liệu.
  1. Cấu trúc của mô hinh OSI
* Tầng 1(tâng vật lý-Physical): cung cấp các phương tiện truyền tin, thủ tục khởi động, duy trì hủy bỏ các liên kết vật lý cho phép truyền các dòng dữ liệu ở dạng bit.
* Tầng 2(tâng liên kết dữ liệu-Data link): thiết lập, duy trì, hủy bỏ các liên kết dữ liệu, kiểm soát luồng dữ liệu, khắc phục sai sót khi truyền tin.
* Tầng 3(tầng mạng-Network): chọn đường truyền tin trong mạng, thực hiện kiểm soát luồng dữ liệu, khắc phục sai sót, cắt hợp dữ liệu.
* Tầng 4(tâng giao vận-Transport): kiểm soát giữa các nút của luồng dữ liệu, khắc phục sai sót, có thể thực hiện ghép kênh và cắt hợp dữ liệu.
* Tầng 5(tầng phiên-Session): thiết lập, duy tri đồng bộ hóa và hủy bỏ các phiên truyền thông. Liên kết phải được thiết lập thông qua đối thoại và các tham số điều khiển.
* Tầng 6(tầng trình dữ liệu-Presentation): biểu thông tin theo cú phát dữ liệu của người sử dụng. Loại mã sử dụng và vấn đề nén dữ liệu.
* Tầng 7(tầng ứng dụng-Application): là giao diện giữa người dùng và môi trường trong hệ thống mở. Xử lý ngữ nghĩa thông tin, tầng này có chức năng cho phép truy cập và quản chuyển giao tiếp, thư tín điện tử.



1. Giao thức TCP/IP

Bộ giao thức TCP/IP - Internet protocol suite hoặc IP suite hoặc TCP/IP protocol suite/ bộ giao thức liên mạng, là một bộ các giao thức truyền thông cài đặt chồng giao thức mà Internet và hầu hết các mạng máy tính thương mại đang chạy trên đó. Bộ giao thức này được đặt tên theo hai giao thức chính của nó là TCP (Giao thức Điều khiển Giao vận) và IP (Giao thức Liên mạng). Chúng cũng là hai giao thức đầu tiên được định nghĩa.

Như nhiều bộ giao thức khác, bộ giao thức TCP/IP có thể được coi là một tập hợp các tầng, mỗi tầng giải quyết một tập các vấn đề có liên quan đến việc truyền dữ liệu, và cung cấp cho các giao thức tầng cấp trên một dịch vụ được định nghĩa rõ ràng dựa trên việc sử dụng các dịch vụ của các tầng thấp hơn. Về mặt lôgic, các tầng trên gần với người dùng hơn và làm việc với dữ liệu trừu tượng hơn, chúng dựa vào các giao thức tầng cấp dưới để biến đổi dữ liệu thành các dạng mà cuối cùng có thể được truyền đi một cách vật lý.

1. Mô hình Client-Server

Theo mô hình Client-Server, một chương trình ứng dụng được chia thành hai phần:

* Quá trình chuyên cung cấp một số phục vụ nào đó ví dụ phục vụ web, phục vụ tập tin, phục vụ thư điện tử… gọi là server.
* Quá trình có yêu cầu sử dụng các dịch vụ do server cung cấp được gọi là client.

Giao tiếp giửa server và client được thực hiện dưới hình thức trao đổi thông điệp(message). Request message từ client và Reply message từ phía server.

1. Socket
   1. Giới thiệu socket

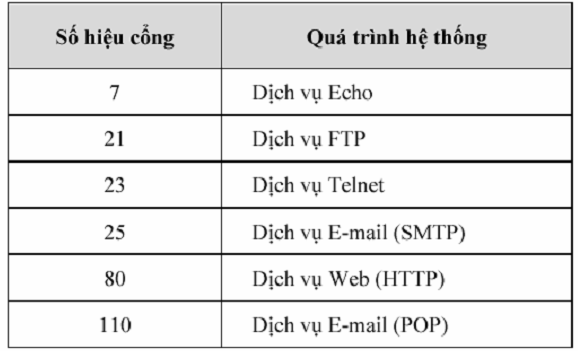
Có hai kiểm socket:

* AF\_UNIX: chỉ cho phép giao tiếp giửa các quá trình trong cùng một máy tính.
* AF\_INET: cho phép giao tiếp giữa các quá trình trên các máy tính khác nhau
  1. Khái niệm cổng(port)

Để thực hiện các cuộc giao tiếp, quá trình cần có cổng của socket mà mình sử dụng. Mỗi cổng giao tiếp thể hiện một địa chỉ xác định trong hệ thống.

Số hiệu cổng gán cho socket phải duy nhất trên máy tính có giá trị trong khoảng 0 – 65535(16 bit), các cổng từ 1 tới 1023 được dành riêng cho các quá trình hệ thống.

Một số cổng thông dụng:



* 1. Giao thức giao tiếp TCP và UDP

Giao tiếp có kết nối sử dụng giao thức TCP

Giao tiếp không có kết nối sử dụng giao thức TCP

|  |  |
| --- | --- |
| Giao tiếp có kết nối TCP | Giao tiếp không kết nối UDP |
| Tồn tại kênh giao tiếp giữa hai bên giao tiếp.  Dữ liệu được gửi đi theo chế độ đảm bảo: có kiểm tra lỗi, truyền lại gói tin lỗi hay mất, đảm bảo thứ tự đến các gói tin…  Dữ liệu chính xác, tốc độ truyền chậm. | Không tồn tại kênh giao tiếp ảo giữa hai bên giao tiếp.  Dữ liệu được gửi theo chế độ không đảm bảo.  Dữ liệu không chính xác, tốc độ truyền nhanh.  Thích hợp cho các ứng dụng cần tốc độ không cần chính xác cao… |

* 1. Mô hình Client – Server sử dụng TCP

Trước tiến ta sẽ tìm hiểu chương trình Server ở chế độ lặp

Giai đoạn một: Server tạo socket, gán số hiệu cổng và lắng nghe yêu cầu kết nối.

Socket()

Bind()

Listen()

Server

* Quá trình Socket(): server yêu cầu một socket để có thể sử dụng các dịch vụ của tầng vận chuyển.
* Quá trình Bind(): server yêu cầu gán số hiệu cổng cho socket.
* Quá trình Listen(): server lắng nghe các yêu cầu kết nối từ các client trên cổng đã được gán

Giai đoạn 2: client tạo socket, yêu cầu thiết lập một kết nối tới server

Socket()

Connect()

Socket()

Bind()

Listen()

Client

* Quá trình Socket(): client yêu cầu khởi tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển, thông thường hệ thống tự động gán một số hiệu cổng còn rãnh cho socket của client.
* Quá trình Connect(): client gửi yêu cầu connect tới server có địa chỉ IP và port xác định.
* Quá trình Accept(): server chấp nhận kết nối của client khi đó một kênh giao tiếp ảo được hình thành, client và server có thể trao đổi thông tin qua kênh giao tiếp ảo này.

Giai đoạn 3: trao đổi thông tin giửa client và server

Accept()

Client

Server

Request Message

Reply Message

* Sau khi tiếp nhận yêu cầu kết nối, thông thường server thực hiện lệnh read và nghẽ(blocked) cho tới khi có thông điệp từ client gửi tới.
* Server phân tích và thực thi yêu cầu, kết quả sẽ được gửi về cho client bằng lệnh write().
* Sau khi gửi yêu cầu bằng lệnh write(), client chờ nhận thông điệp từ server bằng lệnh read().
* Việc trao đổi giữa client và server phải tuân thủ theo giao thức của ứng dụng.

Giao đoạn 4: kết thúc phiên làm việc

Accept()

Client

Server

Request Message

Reply Message

Close()

Close()

* Các câu lệnh read(), write() có thể thực hiện được nhiều lần.
* Kênh ảo sẽ mất đi khi server hoặc client thực hiện socket bằng lệnh close().
  1. Mô hình Client – Server sử dụng UDP

Giai đoạn 1: server tạo socket, gán số hiệu cổng

Socket()

Bind()

Server

* Quá trình socket(): server gửi một yêu cầu socket để có thể sử dụng các dịch vụ của tầng vận chuyển.
* Quá trình bind(): server yêu cầu gán số hiệu cổng cho socket.

Giai đoạn 2: client tạo socket

Socket()

Client

Server

Socket()

Bind()

* Quá trình socket(): client yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển, thông thường hệ thống tự động gán một số hiệu cổng còn rảnh cho socket của client.

Giai đoạn 3: trao đổi thông tin giữa client và server

Socket()

Client

Server

Socket()

Bind()

Sau khi tạo socket(), client và server có thể trao đổi thông tin với nhau thông qua hàm sendto() và receivefrom().

* 1. Mô hình Client – Server sử dụng TCP/UDP

Chương trình ở server ở chế độ đồng thời sử dụng TCP gồm các bước như sau:

* Tạo socket(), đăng ký địa chỉ socket với hệ thống.
* Đặt socket ở trạng thái chờ, lắng nghe kết nối.
* Khi có request từ client, chấp nhận kết nối, tạo ra một process con để xử lý. Quay lại trạng thái chờ lắng nghe kết nối mới.
* Công việc của process mới gồm:
  + Nhập thông tin kết nối của client.
  + Giao tiếp với client theo giao thức đã thiết kế.
  + Đóng kết nối và kết thúc process con.

Chương trình server ở chế độ đồng thời sử dụng UDP gồm các bước sau:

* Tạo socket.
* Lặp việc nhận dữ liệu từ client đối với một dữ liệu nhận tạo process(quá trình) để xử lý, tiếp tục nhận dữ liệu từ client.
* Công việc của process mới:
  + Nhận thông tin của process cha truyền đến, lấy thông tin socket.
  + Xử lý và gửi thông tin cho client theo giao thức ứng dụng đã thiết kế.
  + Kết thúc.