LẬP TRÌNH MẠNG

- Tầng 2 data link -> kiểm tra độ chính xác thông tin

- Pm là tập hợp nhiều chương trình để xử lý nhiều yêu cầu

- Lập trình viên bắt đầu lập trình ở tầng 4

- lập trình mạng là lập trình ra các thực thể pm hoạt động ở tầng ứng dụng nhầm cung cấp d.vụ cho người dùng dựa trên các giao thức.

- Giao thức TCP/IP

- Tầng 1 : dữ liệu dạng bit. Thiết bị: repeater, hub

- Tầng 2 (Vật lý): dữ liệu là frame được truyền đi, vì nó có chức năng framing (đóng gói dữ liệu lại). Tbi: switch, bridge

- Tầng 3 : Giao thức IP (là giao thức ko liên kết -> ko có cơ chế tìm lỗi, ko hồi đáp -> ko t.báo lỗi), ICMP là giao thức được bọc chung với IP để kiểm soát lỗi. Tbi: router. Dữ liệu là package

- Tầng 4 (transport): dữ liệu là segmant

- Tầng ứng dụng: DHCP: dvu cấp phát động, BOOTP,... dữ liệu là data

DNS: dvu phân giải tên miền ( đổi từ địa chỉ IP thành tên: 8.8.8.8 -> google)

* Tầng mạng (Network Layer)
  + Thiết bị định tuyến (router)
  + xác định địa chỉ logic – giao thức IP
  + Giao thức IP là giao thức không có kết nối -> Ko nhận phản hồi -> dùng ICMP
  + 2 chức năng quan trọng của IP: Chọn đường và chuyển tiếp
  + 13 bước hoạt động của giao thức IP
    - Gói tin gồm 2 phần: header và data
    - Truyền gói tin đi:
      * Thực thể IP trạm nguồn có 4 bước
      * Thực thể IP trạm đích có 3 bước
      * Router chặn lỗi và xử lý 6 bước
* Tầng 4 (Transport Layer)
  + Dữ liệu ko qtr, cần tgian kết nối nhanh.
  + Ko ảnh hưởng đến băng thông đường truyền
  + DNS
  + Khi quét tệp tin -> ra 6 là dùng TCP, 17 là UDP
* Tầng phiên (Session layer)
  + Chỉ trong ltrinh theo cơ chế có kết nối thì có session.
  + Session cho phép lưu trữ các phiên làm việc giữa khách-chủ
* Tầng trình diễn (Presentation layer)

Truyền theo cơ chế dừng và chờ (bất đồng bộ) ==(lỗi xảy ra)🡺 IDLE\_RQ

* IDLE\_\_RQ (kiểm tra lỗi)
  + Kiểu ẩn
    - Nhận thành công (1)
    - Mất gói tin (2)
    - Mất báo nhận ACK (3)
  + Kiểu hiện
    - Nhận thành công (1)
    - Mất gói tin (2)
    - Mất báo nhận ACK (3)

Truyền liên tục, hang loạt (đồng bộ) ==(lỗi xảy ra)🡺 Contimous RQ

* Continuous RQ
  + Selective Repeat (mất 1 gói – truyền 1)
    - Kiểu ẩn
    - Kiểu hiện
  + Goback – N (mất 1 gói – truyền N)
    - Kiểu ẩn
    - Kiểu hiện

IDLE\_RQ & Continuous RQ : ARQ (Automatic Repeat request) (ở tầng 2)

* Kiểu ẩn: (mỗi lần lấy vùng RAM tin chứa 1 gói tin)
  + Chờ tgian time out, nếu chưa nhận được thông báo nhận thành công -> mất gói tin
    - Gói cũ bị mất ACK -> qa tgian time-out -> truyền gói mới -> Bắt buộc phải nhận gói mới -> có ACK để truyền đi
  + Không phân biệt được mất gói tin hay mất ACK : Chỉ dựa vào việc nhận ACK là thành công.
* Kiểu ẩn là chỉ quan tâm đi gói tin nhận được
* Kiểu thiện: Có quan tâm gói tin bị mất

Continuous RQ ( theo FIFO)

* Selective Repeat: Bên truyền gói tin sẽ tự phát hiện gói mất/ gói ko gửi thành công dựa vào gói nhận được tìm ra gói mất.

Lớp A: 0…127

Lớp B: 128…191

Lớp C: 192…223

Lớp D: 224…239

Lớp E: 240…255

* Luồng hướng Byte (8bit) gồm có:
  + Input Stream / Output Stream
* Luồng hướng Char (16 bit)
  + Reader / Writer

Thư viện: DataInput và DataOutput (luồng nhị phân)

\*\* Client ko gửi trực tiếp dữ liệu tới Server mà Trong lập trình Socket, Client gửi data lên Socket và Socket giống như cầu nối (nhận và đẩy dữ liệu) và Server nhận dữ liệu từ Socket; và tương tự.

Mô hình truyền tin Socket:

|  |  |
| --- | --- |
| Server | Client |
| Server Socket (mở ra các port) | Socket (kết nối tới IP, port) |
| Khi Client có như cầu kết nối | |
| Server chấp nhận connect   * Sau khi accept() tạo 1 Socket mới | Client tạo luồng out dữ liệu vào Socket |
| Server nhận in dữ liệu từ Socket |  |
| Sau khi xử lý xong   * Out dữ liệu | In dữ liệu   * View |
| Close() | Close() |

UDP: khi dữ liệu ko qtr, yêu cầu đáp ứng nhanh

* Dữ liệu dễ bị lấy hết