**Mạng Máy Tính 101**

**Các khái niệm cơ bản**

* **Computer Network - Mạng máy tính**: Nhiều máy tính kết nối với nhau bằng phương tiện truyền dẫn, liên lạc và chia sẻ tài nguyên.
* **Internet – Mạng của mạng máy tính**: Có khả năng truy cập toàn cầu.
* **Một vài thiết bị mạng**:
  + Router: Kết nối các mạng
  + Switch: Kết nối các máy trong 1 mạng theo hình sao
  + Modem: Lấy kết nối từ Internet
* **Phân tier cấu trúc internet**:
  + Cấp quốc gia, quốc tế
  + Cấp vùng
  + Nhà cung cấp kết nối cho người dùng cuối
* **Khả năng của mạng máy tính**:
  + Liên lạc, trao đổi thông tin
  + Tự phát hiện và sửa lỗi
  + Tự tìm đường đi tối ưu đến đích
  + Chia sẻ tài nguyên
* **Phân loại mạng**:
  + **Theo địa hình**:
    - LAN: nhỏ, nhanh, ít lỗi, chi phí thấp
    - MAN: trung bình, chi phí cao
    - WAN: lớn, chi phí rất cao, nhiều lỗi
  + **Theo phạm vi hoạt động**:
    - Intranet: nội bộ
    - Extranet: cho phép bên ngoài truy cập theo chứng thực
    - Internet: cho phép bên ngoài truy cập
  + **Theo phương tiện truyền dẫn**:
    - Có dây
    - Không dây: Infrastructure – Ad-hoc
* **Lịch sử**:
  + Ý tưởng đầu tiên là của J.C.R Licklider
  + Khởi đầu là ARPANET năm 1969
  + 1990s năm bùng nổ của Internet
* **Các kiểu truyền**:
  + Unicast: 1 node to 1 node
  + Broadcast: 1 node to all
  + Multicast: 1 node to 1 group
  + Anycast: 1 node to 1 random node in 1 group
* **Protocol – Giao thức**: thống nhất các đối tượng khi trao đổi thông tin, qui định, qui tắc để trao đổi dữ liệu giữa các đối tượng trên mạng.
* **Bandwidth - Băng thông**: Lượng thông tin truyền đi trên 1 kết nối mạng trong 1 khoảng thời gian. Là kết nối lý tưởng.
* **Throughput - Thông lượng**: Băng thông thực tế, nhỏ hơn bandwidth.
* **Độ trễ gói tin**:
  + Trễ do tốc độ truyền (transmission delay)
  + Trễ trên đường truyền (propagation delay)
  + Xử lí tại nút (nodal processing)
  + Hàng đợi (queuing delay)
  + **Các công thức tính độ trễ tổng**:
    - TD: D = L/R = Chiều dài gói tin – bit / Băng thông đường truyền – bps
    - PD: D = D/c = Chiều dài đường truyền/ Tốc độ ánh sáng (3\*10^8 m/sec)
    - NP: Thời gian xử lý header 1 gói tin (kiểm lỗi bit, xác định đầu ra) – Rất nhỏ
    - DD: Thời gian đưa lên đường truyền – Phụ thuộc số lượng gói tin
* **Các lệnh để đo độ trễ**: PING, TRACERT, và PATHPING
* **Firewall – Tường lửa**: Bảo vệ hệ thống, Kiểm soát dữ liệu, Phần mềm/ Đĩa Cứng.
* **Proxy**: Có khả năng thay thế kết nối
* **Các thành phần mạng**:
  + Bên ngoài: Hosts, Phương tiện kết nối, Thiết bị liên mạng
  + Bên trong: Dịch vụ mạng, Giao thức, Phương thức truyền dữ liệu
* **Chuyển mạng mạch**: chia đường dây dẫn ra làm nhiều vùng khác nhau
* **Chuyển mạch gói**: chia sẻ thành các gói nhỏ
* **Network Topology – Đồ hình mạng**: cách bố trí các nút mạng: bus, star, ring, mesh
* **Ứng dụng mạng**: gia đình, thương mại, xã hội, hỗ trợ người di động
* **Vấn đề liên quan đến mạng**: tội phạm, bảo mật, tội phạm, …

**IP và Subnet**

* **Identifier – Địa chỉ mạng**: Định danh của 1 node mạng.
  + **Địa chỉ vật lý - MAC – Media Access Control**: do nhà sản xuất đặt cho card mạng.
  + **Địa chỉ logic IP – Internet Protocol:** do người dùng ấn định.
* **Địa chỉ IP nằm ở tầng 3 mô hình OSI**, có các version: IPv4, IPv5, và IPv6
* **Các đặc điểm của IPv4**:
  + 4 bytes – 32 bits
  + Mỗi bytes một số thập phân – octet
  + 2 octet cách nhau bằng 1 dấu chấm
  + VD: 172.29.1.0
  + Chia thành 2 phần: Network ID và Host ID
* **Subnet Mask**: Dùng xác địch Net ID và Host ID, dài 4 bytes, Network ID có các bit = 1; Host ID có các bit = 0
* **Net Address – Địa chỉ đường mạng**: Net ID giữ nguyên, Host ID về 0
* **Broadcast Address – Địa chỉ broadcast**: Net ID giữ nguyên, Host ID lên 1.
  + VD: 192.168.1.2/24
  + -> Net Address: 192.168.1.0
  + -> Broadcast Address: 192.168.1.255
* **Số địa chỉ host hợp lệ** trong 1 đường mạng: 2^m - 2 – m là số bit của Host ID

**Phân lớp Net ID**

|  |  |
| --- | --- |
| Class | Range |
| A | 1-126 |
| Ko được assigned, loopback | 127 |
| C | 128-191 |
| D | 192-223 |
| E | 224-239 |
| F | 240-255 |

* **Subnet mặc định**:
  + A - / 8 - 1 bytes;
  + B - /16 - 2 bytes;
  + C - /24 - 3 bytes
* **Phân loại**:
  + **Địa chỉ public**: dùng để trao đổi trên Internet, địa chỉ thật
  + **Địa chỉ private**: dùng cho các mạng LAN trong 1 tổ chức, địa chỉ ảo
* **Mục tiêu chia subnet**:
  + Giảm số node -> Tăng tốc độ mạng
  + Tăng bảo mật
  + Dễ quản trị, bảo trì
  + Tránh lãng phí IP
* **Cách chia subnet**:
  + Mượn bit đầu Host ID làm Net ID
  + Số subnet = 2^n (số bit vay của Host ID)
* **Địa chỉ MAC**:
  + Tầng 2 OSI
  + Gồm 6 bytes:
    - 3 bytes đầu: IEEE của mỗi nhà sản xuất
    - 3 bytes sau: nhà sản xuất ấn định cho mỗi card mạng

**DHCP & DNS**

* **DHCP – Dynamic Host Configuration Protocol**:
  + Có tiền thân là BOOTP.
  + **Ở Application Layer.**
  + **Chức năng**: cấp phát địa chỉ IP động.
  + Theo mô hình Server-Client:
    - Server: Port 67 – Cung cấp cấu hình TCP/IP cho các client
    - Client: Port 68 – Yêu cầu Server cung cấp TCP/IP
  + **Mô hình hoạt động**:
    - **Xin cấp mới**:
      * Discover: client tìm DHCP server
      * Offer: DHCP gợi ý một địa chỉ IP
      * Request: Client yêu cầu cấp 1 địa chỉ IP
      * Ack: Server xác nhận đồng ý và giải phóng IP
      * Nak: Server từ chối địa chỉ IP mà client yêu cầu
    - **Xin cấp lại**:
      * Request
      * Ack/ Nak
    - **Xin hủy thông tin được cấp**:
      * Release
  + **Một số lệnh**:
    - Kiểm tra thông tin cấu trúc IP: Ipconfig /all
    - Xin cấp một IP mới: Ipconfig /renew
    - Trả lại IP đang dùng: Ipconfig /release
* **DNS – Domain Name System:**
  + **Ở Application Layer** theo mô hình Server – Client.
  + **Chức năng**: Chuyển đổi tên miền sang <-> địa chỉ IP.
  + Gắn với tầng Transport:
    - UDP: truy vấn (query) – port 53
    - TCP: cập nhật thông tin
  + **Lưu trữ**:
    - Bằng file hosts.txt
    - Bằng CSDL phân tán và phân cấp (Name Server)
  + **Các thuật ngữ**:
    - **Name Space**: tập các tên miền
    - **Domain**: một nhánh trong Name Space
    - **Zone**: một vùng được quản lý trong DNS
      * Primary Zone
      * Secondary Zone
      * Stub Zone
    - **FQDN** (Fully Qualified Domain Name): tên đầy đủ
    - **Các kiểu phân giải truy vấn**:
      * Tuần tự
      * Đệ qui
      * **Trong quá trình truy vấn các DNS Server chia làm**:
        + Authoritative DNS Server: Server quản lý tên miền được phân giải
        + Non-Authoritative DNS Server: Ko quản lý, có được câu trả lời bằng caching/ forwarder
  + **Caching**:
    - Lưu tạm kết quả đã truy vấn trong cache
    - Lợi ích:
      * Giảm thời gian truy vấn
      * Giảm lưu lượng mạng
    - DNS: IP -> Domain Name
      * Nhu cầu: Cho địa chỉ IP, hỏi tên Domain
      * Sử dụng nhánh phân cấp
  + Nhận xét:
    - Khi phân giải tên miền ngoài zone mà DNS server quản lý -> DNS Server phải liên hệ với các DNS server để lấy thông tin
    - DNS xử lý nhiều + tốn băng thông
  + Dùng forwarder:
    - Standard forwarder
    - Conditional forwarder

**OSI – TCP/ IP**:

* **Cấu trúc**:
  + Xem mạng như 1 chồng các layer
  + Lớp N cung cấp cho lớp N+1
  + Lớp N sử dụng dịch vụ lớp N-1
  + Mỗi lớp trao đổi với nhau theo 1 giao thức
* **Ưu điểm**:
  + Mỗi lớp có 1 chức năng, giảm độ phức tạp
  + Dễ quản lý
  + Dễ mở rộng, dễ phát triển
  + Đơn giản
* **Mô hình OSI**: Do ISO đề xuất, công bố 1984, là khung sườn biểu diễn cách thông tin di chuyển trên mạng.
  + **Gồm các tầng**:
    - **Application Layer**
      * **Application**: Cung cấp dịch vụ mạng – Cho phép người sử dụng tương tác với nó
      * **Presentation**: Chuyển đổi dữ liệu – Mã hóa, giải mã và nén các dữ liệu
      * **Session**: Quản lí các phiên ứng dụng, kiểm soát việc giao tiếp của các tầng dưới
    - **Data Flow Layer** 
      * **Transport**: End to End, vận chuyển dữ liệu cho người dùng đầu cuối
      * **Network**: Host to Host, định tuyến
      * **Data Link**: Link to Link, truyền giữa các thực thể mạng, theo định chỉ MAC
      * **Physical**: Nhị phân, truyền quá các loại tín hiệu
  + **Mô hình TCP/ IP - Transmission Control Protocol/ Internet Protocol**:
  + **Lịch** sử: Do Cerf và Kahn định nghĩa năm 1974
  + **Gồm các tầng**:
    - Application
    - Transport
    - Internet
    - Host to Network
  + **Đóng gói và Phân rã dữ liệu**:
    - **Application**: Message
    - **Transport**: Segment – TCP (UDP) Header
    - **Network**: Packet – IP Header
    - **Data Link**: Frame – LLC Header and MAC Header
    - **Physical**: Bits

**Tầng ứng dụng**:

* **Chức** năng: Cung cấp dịch vụ mạng cho người dùng cuối.
* **Các ứng dụng phổ biến**: Email, Web, Instant Message, Telnet, etc.
* **Process – Tiến trình**: chương trình chạy trên máy, theo nhiều luồng công việc thread, có địa chỉ là **IP và Port**.
* **Liên lạc giữa các tiến trình**:
  + **Cùng máy**: Hệ điều hành chia sẻ bộ nhớ và truyền thông điệp giữa các tiến trình
  + **Trên các máy khác nhau**: truyền dữ liệu qua đường mạng, chạy trên end-system, liên lạc với nhau qua mạng, kiến trúc p-to-p
    - **Server – Client 101**:
      * **Server**: luôn sống, cố định địa chỉ IP, nhận yêu cầu từ client
      * **Client**: liên lạc và gởi yêu cầu cho Server, có thể dùng IP động, 2 client ko thể liên lạc với nhau
    - **P2P 101**: Ứng dụng có cả 2 chức năng của client và server, các client liên lạc trực tiếp, dùng địa chỉ động, khó quả lý
* **Giao thức Application Layer**: Do người cài đặt xây dựng; VD: HTTP, FTP.
* **Những yêu cầu dịch vụ của Application Layer**:
  + Đáng tin cậy
  + Thời gian nhanh
  + Băng thông tốt
  + Bảo mật dữ liệu
* **Những dịch vụ tầng transport cung cấp**:
  + **TCP**: Dịch vụ hướng kết nối, truyền dữ liệu tin cậy
  + **UDP**: Dịch vụ hướng không kết nối, truyền dữ liệu không đáng tin cậy
* **Port**: Ứng với mỗi services của Application Layer sẽ có một port để liên kết với Transport Layer.
* **Socket**: Cách cửa để giao tiếp với Transport Layer (TCP, UDP), cung cấp interface để lập trình ở tầng Transport, một socket là một end-point liên kết giữa 2 ứng dụng.
* **Lập trình kiến trúc mạng**:
  + Client – Server, P2P
  + Giao thức transport: TCP, UDP
  + Các Port sử dụng ở Server và Client
  + Giao thức tầng ứng dụng khi trao đổi dữ liệu giữa 2 end-host
  + Lập trình

**Tầng vận chuyển**:

* **Nhiệm vụ**: Cung cấp truyền dữ liệu logic giữa 2 máy.
* **Đặc trưng**:
  + Ở end-system
  + Bên gởi: **Dồn kênh – Multiplexing**:
    - Nhận messages từ Application Layer
    - Phân đoạn thành các segment
    - Thu dữ liệu từ socket
    - Dán nhãn, đóng gói
    - Đẩy xuống Network Layer
  + Bên nhận: **Phân kênh – Demultiplexing**:
    - Nhận từ Network Layer
    - Phân ra lại các segment
    - Phân phối segment theo socket tương ứng
    - Convert segment lại thành messages
    - Đẩy lên Application Layer
* **Hỗ trợ**:
  + **Truyền tin cậy**: điều khiển luồng, giải quyết tắt nghẽn, duy trì kết nối
  + **Truyền không tin cậy**: truyền nhanh nhất
  + **Không hỗ trợ**: đảm bảo thời gian trễ, đảm bảo băng thông
* **Segment 101**:
  + 32 bits
  + Thêm vào SOURCE port, DEST port**,** và HEADER fields khác
* **Các cách truyền dữ liệu đáng tin cậy**:
  + Pipeline: Go-Back-N và gởi lại có chọn
  + Giải quyết lỗi bit: Gởi kèm theo thông tin kiểm lỗi, sử dụng checksum và parity check
  + Giải quyết mất gói: Gửi tín hiệu báo ACK, NAK
* **Nguyên tắc RTD**: Dừng và chờ
  + **Bên gửi**:
    - Bên gởi gửi gói tin kèm theo thông tin kiểm tra lỗi
    - Dừng và chờ đến khi gói tin được gửi an toàn, nhận gói ACK
    - Gởi lại khi có lỗi xảy ra
  + **Bên nhận**:
    - Kiểm tra trùng lắp dữ liệu
    - Gửi gói tin phản hồi
* **Nguyên lý Pipe Line**:
  + Cho phép gửi nhiều gói tin khi chưa nhận ACK
  + Sử dụng buffer để lưu các gói tin:
    - Bên gởi: lưu gói tin đã gởi nhưng chưa ack
    - Bên nhận: lưu gói tin đã nhận đúng nhưng chưa đúng thứ tự
  + Giải quyết mất gói:
    - Go back N
    - Selective Repeat (gởi lại có chọn)
* **RDT 3.0**:
  + Bên gửi đợi một khoảng thời gian hợp lí cho ACK
  + Gửi lại nếu không nhận được ACK trong khoảng thời gian này
  + Nếu gói tin (hay ACK) bị trễ (không mất)
    - Gửi lại có thể trùng, phải đánh số thứ tự
    - Bên nhận phải xác định thứ tự của gói tin đã ACK
  + Yêu cầu đếm thời gian
* **Go-Back-N**:
  + Số thứ tự: k-bit
  + Window = N -> Số gói tin được gửi liên tục không ACK
  + **Bên gửi**:
    - Sử dụng buffer (“window”) để lưu các gói tin đã gởi nhưng chưa nhận được ACK
    - Gởi nếu gói tin có thể đưa vào “window”
    - Thiết lập đồng hồ cho gói tin cũ nhất (gói tin ở đầu “window”)
    - Timeout: gửi lại tất cả các gói tin chưa ACK trong window
  + **Bên nhận**:
    - Chỉ gửi ACK cho gói tin đã nhận đúng với số thứ tự cao nhất (có thể trùng ACK)
    - Chỉ cần nhớ số thứ tự đang đợi
    - Gói tin không theo số thứ tự:
      * Loại bỏ: không có bộ đệm
      * Gửi lại ACK với số thứ tự lớn nhất
* **Gửi lại có chọn**:
  + **Bên nhận**:
    - Báo nhận riêng lẻ từng gói tin nhận đúng
    - Dùng bộ đệm để lưu các gói tin không đúng thứ tự
    - Nhận 1 gói tin không đúng thứ tự
  + **Bên gởi**:
    - Có đồng hồ cho mỗi gói tin chưa nhận được ACK
    - Time out: chỉ gửi những gói tin không nhận được ACK
* **TCP – Transport Control Protocol**:
  + **Đặc trưng**:
    - P2P
    - Full-duplex, truyền 2 chiều trên cùng kết nối
    - Handshaking trước khi gửi dữ liệu
    - Cung cấp theo kiểu dòng, sử dụng buffer gởi và nhận
    - Tin cậy, theo thứ tự
    - Pipeline
    - Kiểm soát luồng
    - Kiểm soát tắt nghẽn
  + **Các trường của TCP**:
    - Source & destination port: Port của nơi gởi và nơi nhận
    - Sequence number: Số thứ tự của byte đầu tiên trong phần data của gói tin
    - Acknowledgment number: Số thứ tự của byte đang mong chờ nhận tiếp theo
    - Window size: Thông báo có thể nhận bao nhiêu byte sau byte cuối cùng được xác nhận đã nhận
    - Checksum: Checksum TCP header
    - Urgent pointer: Chỉ đến dữ liệu khẩn trong trường dữ liệu
    - Cờ:
      * URG = trường urgent pointer valid
      * ACK = trường Acknowledge number valid
      * PSH = dữ liệu cần phân phối ngay
      * RST = chỉ định nối kết cần thiết lập lại (reset)
      * SYN = sử dụng để thiết lập kết nối
      * FIN = sử dụng để đóng kết nối
    - Seq: số thứ tự của byte đầu tiên trong vùng data
    - ACK: số thứ tự của byte chờ nhận tiếp theo
  + **TCP thực hiện kết nối bắt tay 3 lần**
  + **TCP thực hiện đóng kết nối bắt tay 2 lần**
  + **Các vấn đề**:
    - **Điều khiển luồng**: Bên gởi làm tràn bộ đệm của bên nhận khi gởi quá nhiều dữ liệu hoặc gởi quá nhanh -> Sử dụng trường “window size”
    - **Kiểm soát tắt nghẽn**: Vấn đề: 1 node có thể nhận dữ liệu từ nhiều nguồn, buffer đến giới hạn, gói tin đến ồ ạt gây mất gói
      * **Giải quyết**: Thiết lập tốc độ gởi dựa trên phản hồi từ bên nhận và Tốc độ gởi: có 2 pha
* **UDP - User Datagram Protocol**:
  + **Đặc điểm**:
    - Dịch vụ “nỗ lực” để truyền nhanh
    - Gói tin UDP có thể: mất và không đúng thứ tự
    - Không kết nối
  + UDP nhanh và header nhỏ -> Application Layer có thể khắc phục lỗi
  + Dùng cho Multimedia