



GT.0000026743

GOC THẮNG (Chủ biên)  
LÊN TIẾN DŨNG

GIÁO TRÌNH

MÁY TÍNH

và MẠNG MÁY TÍNH



NGUYỄN  
C LIÊU



NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM





TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HUNG YÊN

---

TS. PHẠM NGỌC THẮNG (Chủ biên)  
NGUYỄN TIẾN DŨNG

# GIÁO TRÌNH MÁY TÍNH VÀ MẠNG MÁY TÍNH

NHÀ XUẤT BẢN GIÁO DỤC VIỆT NAM



# MỤC LỤC

Trang

LỜI NÓI ĐẦU .....	7
DANH MỤC CÁC TỪ VIẾT TẮT .....	9

## **Chương 1** **TỔNG QUAN HỆ THỐNG MÁY TÍNH**

• MỤC TIÊU .....	11
• NỘI DUNG .....	11
1.1. LỊCH SỬ PHÁT TRIỂN CỦA HỆ THỐNG MÁY TÍNH .....	11
1.2. CÁC THUẬT NGỮ VÀ KHÁI NIỆM CƠ BẢN .....	22
1.3. THÔNG TIN VÀ BIỂU DIỄN THÔNG TIN TRONG MÁY TÍNH .....	23
1.4. SƠ LƯỢC CÁC BỘ PHẬN CỦA HỆ THỐNG MÁY TÍNH .....	49
1.5. GIỚI THIỆU MỘT SỐ PHẦN MỀM THÔNG DỤNG DÙNG CHO MÁY TÍNH .....	53
CÂU HỎI VÀ BÀI TẬP .....	55

## **Chương 2** **KIẾN TRÚC CPU CỦA MÁY TÍNH**

• MỤC TIÊU .....	58
• NỘI DUNG .....	58
2.1. SỰ TIỀN HOÁ CỦA CÁC BỘ VI XỬ LÝ INTEL .....	58
2.2. CẤU TRÚC CPU .....	61
2.3. TẬP CÁC THANH GHI CỦA VI XỬ LÝ .....	66
2.4. CÁC PHƯƠNG PHÁP QUẢN LÝ BỘ NHỚ .....	70
2.5. CÁC PHƯƠNG PHÁP ĐỊNH ĐỊA CHỈ .....	72
2.6. MÃ HOÁ LỆNH .....	73
2.7. TẬP LỆNH VI XỬ LÝ 8086 .....	75
CÂU HỎI VÀ BÀI TẬP .....	86

## **Chương 3** **BỘ NHỚ VÀ CÁC THIẾT BỊ NGOẠI VI**

• MỤC TIÊU .....	88
• NỘI DUNG .....	88
3.1. KIẾN TRÚC HỆ THỐNG NHỚ MÁY VI TÍNH .....	88

3.2. HỆ THỐNG VÀO/RA .....	116
3.3. GHÉP NỐI THIẾT BỊ NGOẠI VI.....	126
3.4. MỘT SỐ THIẾT BỊ NGOẠI VI.....	139
CÂU HỎI VÀ BÀI TẬP.....	145

#### Chương 4

### GIỚI THIỆU VỀ MẠNG MÁY TÍNH VÀ MẠNG INTERNET

• MỤC TIÊU .....	147
• NỘI DUNG.....	147
4.1. KHÁI NIỆM VỀ MẠNG MÁY TÍNH.....	147
4.2. CÁC THÀNH PHẦN MẠNG.....	153
4.3. GIAO THỨC VÀ CÁC MÔ HÌNH DỊCH VỤ .....	169
4.4. CẤU TRÚC MẠNG INTERNET .....	186
CÂU HỎI VÀ BÀI TẬP.....	190

#### Chương 5

### CÁC ỨNG DỤNG MẠNG

• MỤC TIÊU .....	192
• NỘI DUNG.....	192
5.1. NGUYÊN LÝ HOẠT ĐỘNG CỦA CÁC GIAO THỨC LỚP ỨNG DỤNG.....	192
5.2. CÁC CHƯƠNG TRÌNH ỨNG DỤNG .....	202
CÂU HỎI VÀ BÀI TẬP.....	235

#### Chương 6

### LỚP TRUYỀN TẢI

• MỤC TIÊU .....	237
• NỘI DUNG.....	237
6.1. CÁC DỊCH VỤ VÀ NGUYÊN LÝ CƠ BẢN TRONG LỚP TRUYỀN TẢI.....	237
6.2. PHƯƠNG THỨC TRUYỀN DẪN KHÔNG KẾT NỐI UDP .....	243
6.3. PHƯƠNG THỨC TRUYỀN DẪN CÓ KẾT NỐI TCP.....	247
6.4. ĐIỀU KHIỂN LƯỜNG TRONG TCP.....	262
6.5. ĐIỀU KHIỂN LỖI .....	263
CÂU HỎI VÀ BÀI TẬP.....	264

**Chương 7**  
**LỚP MẠNG VÀ CÁC PHƯƠNG PHÁP ĐỊNH TUYẾN**

• MỤC TIÊU.....	265
• NỘI DUNG.....	265
7.1. CHỨC NĂNG LỚP MẠNG TRONG MÔ HÌNH PHẦN LỚP.....	265
7.2. GIAO THỨC LIÊN MẠNG IP.....	271
7.3. GIAO THỨC PHÂN GIẢI ĐỊA CHỈ.....	291
7.4. GIAO THỨC THÔNG DIỆP ĐIỀU KHIỂN INTERNET ICMP.....	293
7.5. GIAO THỨC ĐỊNH TUYẾN.....	294
7.6. ROUTER.....	317
CÂU HỎI VÀ BÀI TẬP.....	326

**Chương 8**  
**MẠNG CỤC BỘ – MẠNG LAN**

• MỤC TIÊU.....	329
• NỘI DUNG.....	329
8.1. ĐẶC TRƯNG CỦA MẠNG CỤC BỘ.....	329
8.2. KIẾN TRÚC MẠNG CỤC BỘ.....	331
8.3. CÁC PHƯƠNG PHÁP TRUY CẬP ĐƯỜNG TRUYỀN.....	335
8.4. CÔNG NGHỆ MẠNG CỤC BỘ – LAN.....	356
8.5. CÁC THIẾT BỊ MẠNG LAN.....	372
8.6. CÔNG NGHỆ MẠNG DIỆN RỘNG WAN.....	380
CÂU HỎI VÀ BÀI TẬP.....	397
DANH MỤC TÀI LIỆU THAM KHẢO.....	398





## Lời nói đầu

**M**ạng máy tính phát sinh từ nhu cầu muốn chia sẻ, dùng chung tài nguyên và cho phép giao tiếp trực tuyến (online) cũng như các ứng dụng đa phương tiện trên mạng. Tài nguyên gồm có tài nguyên phần mềm (dữ liệu, chương trình ứng dụng...) và tài nguyên phần cứng (máy in, máy quét, CD-ROM...). Giao tiếp trực tuyến bao gồm gửi và nhận thông điệp, thư điện tử... Các ứng dụng đa phương tiện có thể là phát thanh, truyền hình, điện thoại qua mạng, hội thảo trực tuyến, nghe nhạc, xem phim...

Sự kết hợp của máy tính với các hệ thống truyền thông đã tạo ra cuộc cách mạng trong vấn đề tổ chức khai thác và sử dụng hệ thống máy tính. Mô hình tập trung dựa trên máy tính lớn được thay thế bởi mô hình các máy tính đơn lẻ được kết nối lại để cùng thực hiện công việc, hình thành môi trường làm việc nhiều người sử dụng phân tán, cho phép nâng cao hiệu quả khai thác tài nguyên chung từ những vị trí địa lý khác nhau. Các hệ thống như thế được gọi là mạng máy tính.

Với sự phát triển của khoa học và kỹ thuật hiện nay, các hệ thống mạng máy tính đã phát triển một cách nhanh chóng và đa dạng cả về quy mô, hệ điều hành và ứng dụng. Do vậy, việc nghiên cứu để vận hành, khai thác chúng ngày càng trở nên phức tạp. Tuy nhiên, các mạng máy tính có cùng các điểm chung, thông qua đó chúng ta có thể khảo sát, phân loại và đánh giá chúng.

**Giáo trình Máy tính và Mạng máy tính** là một trong các cuốn sách của bộ sách **Điện – Điện tử** do tập thể giảng viên Khoa Điện – Điện tử, Trường Đại học Sư phạm Kỹ thuật Hưng Yên biên soạn dựa trên kinh nghiệm giảng dạy và nghiên cứu trong những năm qua. Nội dung chính của cuốn sách nhằm cung cấp cho sinh viên những kiến thức về kiến trúc máy tính, các kỹ năng thiết kế, cấu hình và quản trị hệ thống mạng LAN. Đây có thể xem là những kiến thức nền tảng, quan trọng cho các quản trị viên về hệ thống mạng máy tính. Chính vì vậy, cuốn sách còn là nguồn tài liệu tham khảo hữu ích cho các sinh viên chuyên ngành Công nghệ thông tin, Kỹ thuật máy tính và những người đang làm công tác kỹ thuật trong lĩnh vực Điện, Điện tử và Tin học.

Sách được biên soạn của sách này nhằm mục đích bám sát mục tiêu của học phần, nên sau mỗi phần lý thuyết đều có các ví dụ cụ thể để giúp người học hiểu rõ vấn đề, các hệ thống câu hỏi và bài tập cuối chương giúp người học hệ thống lại và nắm chắc kiến thức. Các ví dụ, bài tập đều đã được kiểm nghiệm thông qua các bài thực hành, thí nghiệm, sẽ rất có ích cho người học trong quá trình làm việc thực tế.

Nội dung của cuốn sách gồm 8 chương:

- Chương 1.** Tổng quan hệ thống máy tính.
- Chương 2.** Kiến trúc CPU của máy tính.
- Chương 3.** Bộ nhớ và các thiết bị ngoại vi.
- Chương 4.** Giới thiệu về mạng máy tính và Internet.
- Chương 5.** Các ứng dụng mạng.
- Chương 6.** Lớp truyền tải.
- Chương 7.** Lớp mạng và các phương pháp định tuyến.
- Chương 8.** Mạng cục bộ – mạng LAN.

Trong quá trình biên soạn cuốn sách này, chúng tôi đã tham khảo một số lượng lớn tài liệu, kết quả nghiên cứu của nhiều tác giả trong và ngoài nước. Chúng tôi xin chân thành cảm ơn các cơ quan, tổ chức, các tác giả với những thông tin đã được trích dẫn trong cuốn sách. Tuy nhiên, do điều kiện thời gian gấp rút nên nhóm tác giả không thể gặp hết các tác giả để xin phép. Rất mong được các tác giả lượng thứ.

Chúng tôi xin chân thành cảm ơn các nhà khoa học, các giảng viên, các đồng nghiệp ở trong và ngoài trường đã có nhiều ý kiến đóng góp quý báu cho nhóm tác giả trong quá trình biên soạn.

Nội dung cuốn sách đã được dùng làm tài liệu giảng dạy cho các lớp Đại học và Cao đẳng trong nhiều năm và đã qua nhiều lần cập nhật, chỉnh sửa. Tuy vậy, cuốn sách khó tránh khỏi những thiếu sót. Chúng tôi rất mong nhận được những ý kiến đóng góp của các đồng nghiệp và bạn đọc để kịp thời chỉnh sửa, bổ sung cho lần tái bản sau. Mọi ý kiến xin gửi về Khoa Điện – Điện tử, Trường Đại học Sư phạm Kỹ thuật Hưng Yên.

CÁC TÁC GIẢ

## DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Chú thích	Ý nghĩa
Acc User	Account User	Tài khoản người dùng
ADSL	Asymmetric Digital Subscriber Line	Đường thuê bao bất đối xứng – kết nối băng thông rộng
ASCII	American Standard Code for Information Interchange	Bộ mã chuẩn của Mỹ dùng để trao đổi thông tin
BCD	Binary Coded Decimal	Bảng mã số BCD
BIOS	Basic Input/Output	Hệ thống nhập/xuất cơ sở
BIU	Bus Interface Unit	Đơn vị giao tiếp BUS
CMD	Command	Dòng lệnh để thực hiện một chương trình nào đó
CPU	Central Processing Unit	Đơn vị xử lý trung tâm
DNS	Domain Name System	Hệ thống phân giải tên miền thành IP và ngược lại
E-Mail	Electronic Mail	Hệ thống thư điện tử
FTP	File Transfer Protocol	Giao thức truyền tải file.
HDD	Har Disk Driver	Ổ đĩa cứng
HTTP	Hyper Text Transfer Protocol	Giao thức truyền tải file dưới dạng siêu văn bản
I/O	Input/Output	Cổng suất nhập
IC	Intergrated Circuit	Mạch tích hợp
IDE	Integrated Development Enviroment	Chuẩn tích hợp mạch điều khiển ổ đĩa
IEEE	Institute of Electrical and Electronics Engineers	Viện Kỹ nghệ điện và điện tử
LAN	Local Area Network	Mạng máy tính cục bộ

Từ viết tắt	Chú thích	Ý nghĩa
MAC	Media Access Control	Khả năng kết nối ở tầng vật lý
NIC	Network Interface Card	Card giao tiếp mạng
OSI	Open System Interconnection	Mô hình liên kết hệ thống mở – chuẩn hoá quốc tế
PC	Program counter	Bộ đếm chương trình
POP	Post Office Protocol	Giao thức văn phòng, dùng để nhận Mail từ Mail Server
RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
ROM	Read Only Memory	Bộ nhớ chỉ đọc
SMTP	Simple Mail Transfer Protocol	Giao thức truyền file đơn giản dùng để nhận Mail từ Mail Client đến Mail Server
TCP/IP	Transmission Control Protocol/Internet Protocol	Giao thức mạng
URL	Uniform Resource Locator	Dùng để định nghĩa một Website, là đích của một liên kết
USB	Universal Serial Bus	Chuẩn truyền dữ liệu cho BUS (thiết bị) ngoại vi.
UTP	Unshielded Twisted Pair)	Cáp xoắn đôi – dùng để kết nối mạng thông qua đầu nối RJ45
VLSM	Variable-Length Subnet Mask	Mặt nạ mạng con có chiều dài thay đổi
WAN	Wide Area Network	Mạng máy tính diện rộng
Wi-Fi	Wireless Fidelity	Kỹ thuật mạng không dây
www	World Wide Web	Hệ thống Web diện rộng toàn cầu



# Chương 1

## TỔNG QUAN HỆ THỐNG MÁY TÍNH

### MỤC TIÊU

Sau khi học xong chương này, người học:

- ✓ Hiểu được lịch sử phát triển của hệ thống máy tính, cũng như xu hướng phát triển của hệ thống máy tính. Nắm vững các khái niệm cơ bản liên quan đến các hệ thống số được dùng trong máy tính. Thành thạo các thao tác biến đổi số giữa các hệ thống số.
- ✓ Nắm được cấu trúc của hệ thống máy tính, các thuật ngữ và khái niệm cơ bản của hệ thống máy tính.
- ✓ Biết được cách biểu diễn thông tin trong máy tính và ý nghĩa của một số phần mềm thông dụng.
- ✓ Nhận biết, kiểm tra, sửa chữa hỏng hóc một số bộ phận của hệ thống máy tính và cài đặt hệ điều hành Window XP.

### NỘI DUNG

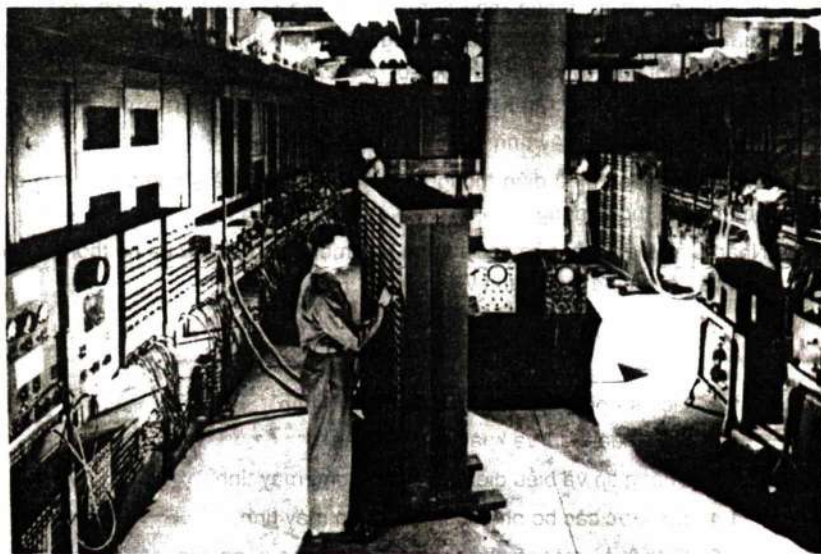
- 1.1. Lịch sử phát triển của hệ thống máy tính.
  - 1.2. Các thuật ngữ và khái niệm cơ bản.
  - 1.3. Thông tin và biểu diễn thông tin trong máy tính.
  - 1.4. Sơ lược các bộ phận của hệ thống máy tính.
  - 1.5. Giới thiệu một số phần mềm thông dụng dùng cho máy tính.
- Câu hỏi và bài tập.

## 1.1. LỊCH SỬ PHÁT TRIỂN CỦA HỆ THỐNG MÁY TÍNH

Sự phát triển của máy tính được mô tả dựa trên sự tiến bộ của các công nghệ chế tạo các linh kiện cơ bản của máy tính như: vi xử lý, bộ nhớ, ngoại vi...

Máy tính điện tử số trải qua năm thế hệ liên tiếp. Việc chuyển từ thế hệ trước sang thế hệ sau được đặc trưng bằng sự thay đổi cơ bản về công nghệ.

- **Thế hệ 1:** Máy tính dùng đèn chân không (*Vacuum Tube*) (1940 – 1956).
- **Thế hệ 2:** Máy tính dùng Transistor (1956 – 1963).
- **Thế hệ 3:** Máy tính dùng mạch tích hợp IC (*Integrated Circuit*) (1963 – 1971).
- **Thế hệ 4:** Máy tính dùng mạch tích hợp cực lớn VLSI (*Very Large Scale Intergration*) (1971 đến nay).
- **Thế hệ 5:** Máy tính sử dụng trí tuệ nhân tạo (từ ngày nay đến sau này).



Hình 1.1. Máy tính ENIAC – *Electronic Numerical Integrator and Computer* (Máy tính thế hệ 1)

### 1.1.1. Máy tính thế hệ 1

#### a) Máy tính ENIAC (*Electronic Numerical Integrator and Computer*)

Đề xuất năm 1943 và hoàn thành năm 1946, được sử dụng đến năm 1956, bởi thầy trò Eckert và Mauchly – Trường đại học Pennsylvania của Mỹ.



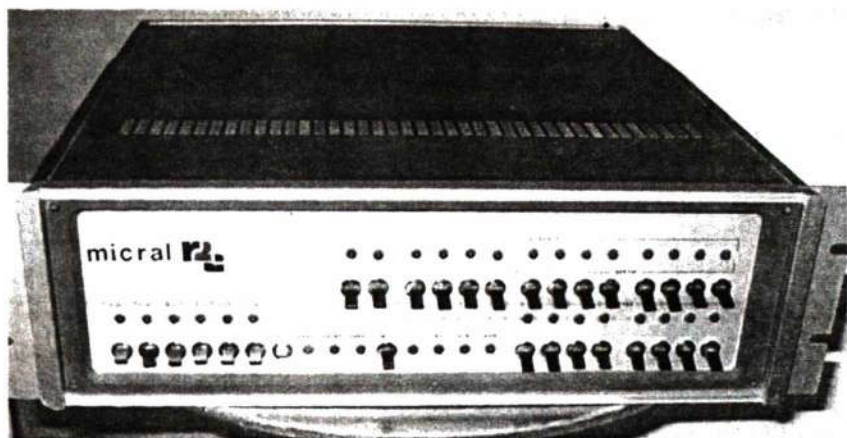
Đặc điểm chính:

- Năng 30 tấn, chiếm diện tích khoảng 150m<sup>2</sup> và sử dụng 140KW/h.
- 5000 nghìn phép cộng trên giây.
- Sử dụng hệ thập phân.
- Lập trình bằng công tắc.
- Sử dụng 18000 bóng đèn điện tử (*vacumm tubes*).

### **b) Máy tính IAS (Institute for Advanced Studies)**

Xây dựng dựa trên ý tưởng của Turring (Mỹ) và Von Neumann (Anh). Thiết kế năm 1947 và hoàn thành năm 1952.

Vào giữa những năm 50 của thế kỷ XX, khi những thế hệ máy tính đầu tiên được đưa vào hoạt động thực tế, với những bóng đèn điện tử thì chúng có kích thước rất cồng kềnh và tốn nhiều năng lượng. Hồi đó việc nhập dữ liệu vào các máy tính được thông qua các tấm bia mà người viết chương trình đã đục lỗ sẵn. Mỗi tấm bia tương đương với một dòng lệnh mà mỗi một cột của nó có chứa tất cả các ký tự cần thiết mà người viết chương trình phải đục lỗ vào ký tự mình lựa chọn. Các tấm bia được đưa vào một "thiết bị" gọi là thiết bị đục bia mà qua đó các thông tin được đưa vào máy tính (hay còn gọi là trung tâm xử lý), và sau khi tính toán kết quả sẽ được đưa ra máy in.



Hình 1.2. Micral (Pháp) máy vi tính lắp ráp hoàn toàn đầu tiên

Như vậy, các thiết bị đọc bìa và máy in được thể hiện như các thiết bị vào/ra (I/O) đối với máy tính. Sau một thời gian các thế hệ máy mới được đưa vào hoạt động, trong đó một máy tính trung tâm có thể được nối với nhiều thiết bị vào/ra mà qua đó nó có thể thực hiện liên tục hết chương trình này đến chương trình khác.

### 1.1.2. Máy tính thế hệ 2

– Lab Bell phát minh ra transistor vào năm 1947, các đèn điện tử được thay thế bằng đèn điện tử lưỡng cực.

– Đến cuối năm 1950, máy tính thương mại dùng transistor xuất hiện trên thị trường.

– Kích thước máy tính giảm, rẻ tiền hơn, tiêu tốn năng lượng ít hơn.

– Vào thời điểm này mạch in và bộ nhớ bằng vòng xuyên từ được dùng.

– Ngôn ngữ cấp cao xuất hiện (như FORTRAN năm 1956, COBOL năm 1959, ALGOL năm 1960) và hệ điều hành kiểu tuần tự (*Batch processing*) được dùng.

### 1.1.3. Máy tính thế hệ 3

– Xuất hiện các mạch tích hợp IC (*Integrated Circuit*).

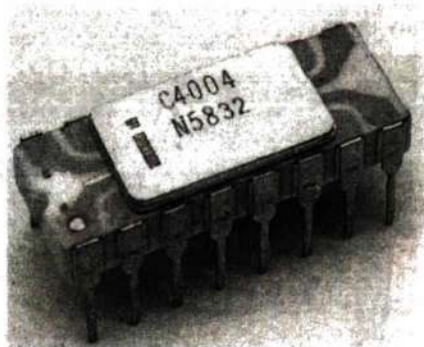
– Các mạch tích hợp mật độ thấp (*SSI: Small Scale Integration*) có thể chứa vài chục linh kiện.

– Mạch tích hợp mật độ trung bình (*MSI: Medium Scale Integration*) chứa hàng trăm linh kiện trên mạch tích hợp.

– Mạch in nhiều lớp xuất hiện.

– Bộ nhớ bán dẫn bắt đầu thay thế bộ nhớ bằng xuyên từ.

– Máy tính đa chương trình và hệ điều hành chia thời gian được dùng.



Hình 1.3. Chip VXL intel 4004

#### 1.1.4. Máy tính thế hệ 4

– IC có mật độ tích hợp cao (*LSI: Large Scale Integration*) có thể chứa hàng ngàn linh kiện.

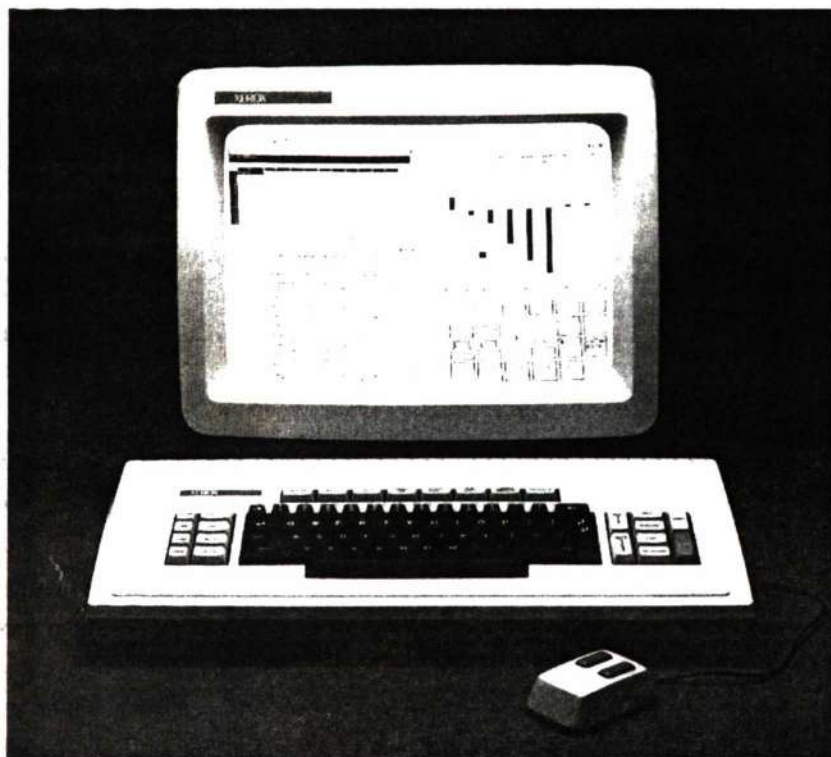
– IC mật độ tích hợp rất cao (*VLSI: Very Large Scale Integration*) có thể chứa hơn 10 ngàn linh kiện trên mạch (Hiện nay, các chip VLSI chứa hàng triệu linh kiện).

– Xuất hiện bộ vi xử lý (*microprocessor*) chứa cả phần thực hiện và phần điều khiển của một bộ xử lý.

– Bắt đầu xuất hiện các thế hệ máy tính cá nhân.

– Bộ nhớ bán dẫn, bộ nhớ cache, bộ nhớ ảo được dùng rộng rãi.

– Dùng kỹ thuật ống dẫn, kỹ thuật vô hướng, xử lý song song...



Hình 1.4. IBM/PC đầu tiên khởi đầu cho dòng máy PC phổ biến ngày nay

### 1.1.5. Máy tính thế hệ 5

Máy tính thế hệ 5 đang trong giai đoạn phát triển, hoạt động dựa trên trí thông minh nhân tạo. Mục tiêu của máy tính thế hệ 5 là xây dựng các thiết bị tính toán có đầu vào là ngôn ngữ tự nhiên, thiết bị có khả năng học tập và tự tổ chức. Tính toán lượng tử, tế bào và công nghệ nano sẽ được sử dụng trong thế hệ này. Đặc điểm của máy tính thế hệ này là:

- Sử dụng chip có mật độ tích hợp siêu lớn;
- Có trí tuệ nhân tạo;
- Có thể nhận biết hình ảnh, đồ hoạ;
- Máy tính thế hệ 5 có thể giải quyết những vấn đề phức tạp, bao gồm việc ra quyết định, lập luận lôgic;
- Máy tính thế hệ 5 có thể sử dụng nhiều hơn một CPU nên tốc độ xử lý nhanh hơn;
- Máy tính thế hệ 5 được dự định làm việc với ngôn ngữ tự nhiên.

### 1.1.6. Xu hướng phát triển của các hệ thống máy tính

Việc chuyển từ thế hệ 4 sang thế hệ 5 còn chưa rõ ràng. Người Nhật đã và đang đi tiên phong trong các chương trình nghiên cứu để cho ra đời thế hệ 5 của máy tính, thế hệ của những máy tính thông minh, dựa trên các ngôn ngữ trí tuệ nhân tạo như LISP và PROLOG... và những giao diện người – máy thông minh.

Đến thời điểm này, các nghiên cứu đã cho ra các sản phẩm bước đầu và gần đây (2004) là sự ra mắt sản phẩm người máy thông minh gần giống với con người nhất: ASIMO (*Advanced Step Innovative Mobility – Bước chân tiên tiến của đổi mới và chuyển động*). Với hàng trăm nghìn máy móc điện tử tối tân được đặt trong cơ thể, ASIMO có thể lên/xuống cầu thang một cách uyển chuyển, nhận diện người, các cử chỉ hành động, giọng nói và đáp ứng một số mệnh lệnh của con người. Thậm chí, nó có thể bắt chước cử động, gọi tên người và cung cấp thông tin ngay sau khi bạn hỏi, rất gần gũi và thân thiện.



Hiện nay có nhiều công ty, viện nghiên cứu của Nhật thuê ASIMO tiếp khách và hướng dẫn khách tham quan như: Viện Bảo tàng Khoa học năng lượng và Đổi mới quốc gia, hãng IBM Nhật Bản, Công ty Điện lực Tokyo. Hãng Honda bắt đầu nghiên cứu ASIMO từ năm 1986, dựa vào nguyên lý chuyển động bằng hai chân. Cho tới nay, hãng đã chế tạo được 50 robot ASIMO. Các tiến bộ liên tục về mật độ tích hợp trong VLSI đã cho phép thực hiện các mạch vi xử lý ngày càng mạnh (8 bit, 16 bit, 32 bit và 64 bit với việc xuất hiện các bộ xử lý RISC năm 1986 và các bộ xử lý siêu vô hướng năm 1990). Chính các bộ xử lý này giúp thực hiện các máy tính song song với từ vài bộ xử lý đến vài ngàn bộ xử lý. Điều này làm các chuyên gia về kiến trúc máy tính tiên đoán máy tính thế hệ 5 là thế hệ các máy tính xử lý song song.

#### *a) Máy tính thông minh*

– Máy tính thông minh dựa trên các ngôn ngữ trí tuệ nhân tạo như LISP, PROLOG...

– Máy tính song song với từ vài bộ xử lý đến vài ngàn bộ xử lý.

– Máy tính đứng thứ hai thế giới hiện nay thuộc dự án INCITE (2009) của Bộ Năng lượng Mỹ với khả năng xử lý 1,75 petaflop (triệu tỷ phép tính mỗi giây).

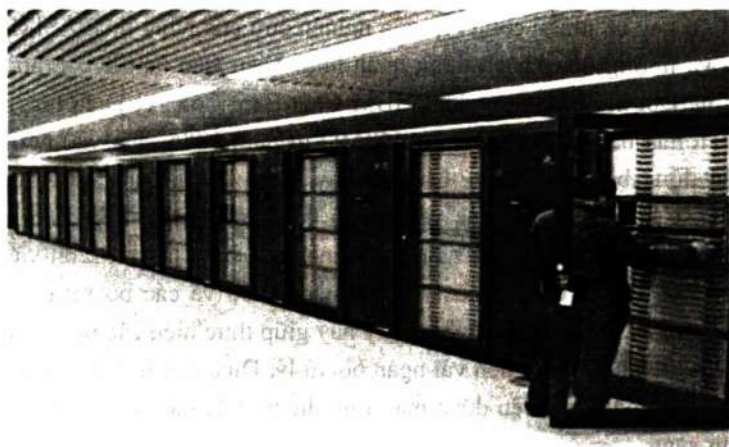
– Đứng thứ ba là Nebulae (2009) thuộc Trung tâm Máy tính siêu quốc gia ở Thẩm Quyển (Trung Quốc) có tốc độ 1,27 petaflop.

– Đứng thứ tư là Tasubame 2.0 (2010) của Nhật với tốc độ 1,2 petaflop và được đặt tại Viện Công nghệ Tokyo (Tokyo Tech).

– Hopper (2010) được đặt theo tên Admiral Grace Hopper – người tiên phong trong việc phát triển phần mềm và ngôn ngữ lập trình, là siêu máy tính mạnh thứ năm thế giới với tốc độ 1,05 petaflop, hiện được đặt tại trung tâm điện toán NERSC thuộc Bộ Năng lượng Mỹ.

– Blue Gene/Q "Mira" được IBM lên kế hoạch hoạt động từ năm 2012, với tốc độ tới 10 petaflop, gấp 4 lần Tianhe-1A.

– Exa – Scale Computer (2018 – 2020) là tham vọng lớn của Mỹ trong việc xây dựng một siêu máy tính nhanh gấp 1.000 lần máy tính nhanh nhất hiện nay, và có thể đếm mọi ngôi sao trong vũ trụ chỉ trong 20 phút.



*Hình 1.5. Siêu máy tính Tianhe-1A (2010) của Trung Quốc đạt tốc độ 2,5 petaflop*

- Deep Rybka 3 (2011) sẽ là máy tính chơi cờ vua giỏi nhất thế giới.
- Stratus có tốc độ 0,07 petaflop thuộc Trung tâm Dự báo khí tượng Mỹ, được coi là siêu máy tính dự đoán khí hậu và thời tiết chuẩn nhất thế giới.
- Watson (2010) là máy tính chơi game Jeopardy giỏi nhất và nó tự đưa ra câu trả lời mà không cần người điều khiển.

***b) Giao diện người – máy thông minh ASIMO (Advanced Step Innovative Mobility – Bước chân tiên tiến của đôi mới và chuyển động)***



*Hình 1.6. Người máy thông minh ASIMO*





Hình 1.7. Robot HRP-4C có khả năng trình diễn thời trang tại Nhật Bản



Hình 1.8. HRP-4C thể hiện các dạng vẻ mặt

Robot người mẫu được đặt tên là HRP-4C này có 30 motor trong cơ thể, cho phép "cô" bước và chuyển động cánh tay một cách mềm mại. 8 motor trên mặt giúp HRP-4C thể hiện sự giận dữ và ngạc nhiên. Người máy đặc biệt này có chiều cao 1,6m và nặng 40kg (thông số trung bình của phụ nữ Nhật Bản).

### c) Hệ thống nhúng

Sau sự phát triển của máy tính lớn và mini (*mainframe* và *mini computer*) giai đoạn 1960 – 1980, PC – Internet giai đoạn 1980 – 2000, thì hiện nay chúng ta đang ở thời đại hậu PC. Giai đoạn hậu PC – Internet này được dự đoán từ năm 2000 đến năm 2020 là giai đoạn của môi trường thông

minh mà hệ thống nhúng là cốt lõi, và đang làm nên làn sóng đổi mới thứ ba trong sự phát triển của Công nghệ thông tin. Sức đẩy của công nghệ đưa công nghệ vi điện tử, các công nghệ vi cơ điện, công nghệ sinh học hội tụ, tạo nên các chip của công nghệ nano, là nền tảng cho những thay đổi cơ bản trong công nghệ thông tin và truyền thông. Sức kéo của thị trường đòi hỏi các thiết bị phải có nhiều chức năng thân thiện với người dùng, có mức độ thông minh ngày càng cải thiện, đưa đến vai trò và tầm quan trọng của các hệ thống nhúng ngày càng cao.

Thời kỳ hậu PC là thời kỳ có những chuyển biến lớn trong công nghệ, đó là:

- Chip sẽ chuyển từ chip linh kiện sang chip hệ thống, có khả năng kết nối mạng và phần mềm mạnh.

- Chuyển dịch từ trọng tâm là PC sang trọng tâm là truyền thông và tính toán chuyên ngành. Trọng tâm thiết kế chuyển sang thiết kế các sản phẩm tiêu dùng có khả năng kết nối không dây, hội thoại với người dùng theo ngôn ngữ tự nhiên, ngôn ngữ hình ảnh và dáng điệu...



*Hình 1.9. Các hệ thống nhúng*

- Giá trị dịch vụ sẽ chiếm tỷ lệ chủ chốt so với giá thành thiết bị kết nối. Do các chuyển biến này mà các thiết bị nhúng đòi hỏi phải có những đặc tính nổi trội sau:

- Giá thành sản phẩm rẻ và tiêu tốn ít năng lượng.
- Có khả năng tái cấu hình qua mạng và có tính chuyên dụng cao.

Bảng 1.1. Các thế hệ máy tính

Thế hệ	Năm	Kỹ thuật	Sản phẩm mới	Hãng sản xuất và máy tính
1	1940 – 1956	Đèn điện tử	Máy tính điện tử tung ra thị trường	IBM 701. UNIVAC
2	1956 – 1963	Transistor	Máy tính rẻ tiền	Burroughs 6500. NCR, CDC 6600, Honeywell
3	1963 – 1971	Mạch IC	Máy tính mini	50 Hãng mới: DEC PDP – 11, Data General, Nova
4	1971 đến nay	LSI – VLSI	Máy tính cá nhân và trạm làm việc	Apple II, IBM – PC, Appolo DN 300, Sun 2...
5	Hiện tại đến tương lai	Tri tuệ nhân tạo	Máy tính đa xử lý, đa máy tính, trí tuệ nhân tạo, tế bào, công nghệ nano	Sequent. Thinking machine Inc, Honda, Casio

### 1.1.7. Phân loại máy tính

Thông thường máy tính được phân loại theo tính năng kỹ thuật và giá tiền.

- **Siêu máy tính** (*Super Computer*)

Là các máy tính đắt tiền nhất và tính năng kỹ thuật cao nhất. Giá bán một siêu máy tính từ vài triệu USD. Các siêu máy tính thường là các máy tính vectơ hay các máy tính dùng kỹ thuật vô hướng, được thiết kế để tính toán khoa học, mô phỏng các hiện tượng. Các siêu máy tính được thiết kế với kỹ thuật xử lý song song, với rất nhiều bộ xử lý (hàng ngàn đến hàng trăm ngàn bộ xử lý trong một siêu máy tính).

- **Máy tính lớn** (*Mainframe*)

Là loại máy tính đa dụng, nó có thể dùng cho các ứng dụng quản lý cũng như các tính toán khoa học. Dùng kỹ thuật xử lý song song và có hệ

thống vào/ra mạnh. Giá một máy tính lớn có thể từ vài trăm ngàn USD đến hàng triệu USD.

- **Máy tính mini** (*Minicomputer*)

Là loại máy cỡ trung bình, giá một máy tính mini có thể từ vài chục USD đến vài trăm ngàn USD.

- **Máy vi tính** (*Microcomputer*)

Là loại máy tính dùng bộ vi xử lý, giá một máy vi tính có thể từ vài trăm USD đến vài ngàn USD.

## 1.2. CÁC THUẬT NGỮ VÀ KHÁI NIỆM CƠ BẢN

Với sự phát triển của khoa học và kỹ thuật, hiện nay các mạng máy tính đã phát triển một cách nhanh chóng và đa dạng cả về quy mô, hệ điều hành và ứng dụng. Do vậy, việc nghiên cứu chúng ngày càng trở nên phức tạp. Tuy nhiên, các mạng máy tính cũng có cùng các điểm chung, thông qua đó chúng ta có thể đánh giá và phân loại chúng.

- **Máy tính** (*Computer*)

Là thiết bị điện tử thực hiện các công việc:

- Nhận thông tin vào.
- Xử lý thông tin theo chương trình sẵn bên trong bộ nhớ máy tính.
- Đưa thông tin ra.

- **Chương trình** (*Program*)

– Chương trình là dãy các câu lệnh nằm trong bộ nhớ, nhằm mục đích hướng dẫn máy tính thực hiện một công việc cụ thể nào đó.

- Máy tính thực hiện theo chương trình.

- **Phần mềm** (*Software*)

Bao gồm chương trình và dữ liệu.

- **Phần cứng** (*Hardware*)

Bao gồm tất cả các thành phần vật lý cấu thành nên hệ thống máy tính.

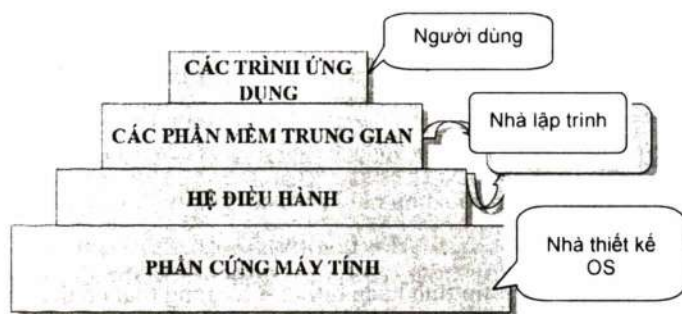
- **Phần dẻo** (*Firmware*)

Là thành phần chứa cả hai phần trên.

- **Kiến trúc máy tính** (*Computer Architecture*)

Đề cập đến các thuộc tính của hệ thống máy tính dưới cái nhìn của người lập trình, hay nói cách khác, là những thuộc tính ảnh hưởng trực tiếp đến quá trình thực hiện logic của chương trình. Bao gồm: tập lệnh, biểu diễn dữ liệu, các cơ chế vào/ra, kỹ thuật đánh địa chỉ...

- **Cấu trúc máy tính** (*Computer structure*)



Hình 1.10. Mô hình phân lớp hệ thống máy tính

Cấu trúc máy tính là những thành phần của máy tính, và liên kết giữa các thành phần. Ở mức cao nhất máy tính gồm 4 thành phần:

- Bộ xử lý: Điều khiển và xử lý số liệu;
- Bộ nhớ: Chứa các chương trình và dữ liệu;
- Hệ thống vào/ra: Trao đổi thông tin giữa máy tính với bên ngoài;
- Liên kết giữa các hệ thống (BUS): Liên kết các thành phần của máy tính lại với nhau.

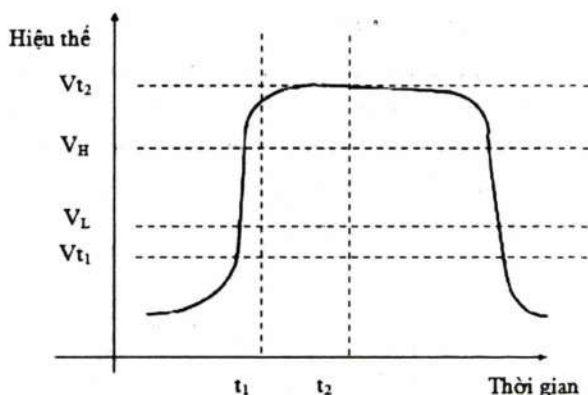
### 1.3. THÔNG TIN VÀ BIỂU DIỄN THÔNG TIN TRONG MÁY TÍNH

#### 1.3.1. Lượng thông tin và sự mã hoá thông tin

- **Trạng thái có ý nghĩa**

Trạng thái thấp khi có hiệu điện thế thấp hơn  $V_L$  và trạng thái cao khi hiệu điện thế cao hơn  $V_H$ . Để có thông tin, ta phải xác định thời điểm ta nhìn trạng thái của tín hiệu. Ví dụ, tại thời điểm  $t_1$  thì tín hiệu ở trạng thái thấp và tại thời điểm  $t_2$  thì tín hiệu lại ở trạng thái cao. Đơn vị đo lường thông tin là bit.





Hình 1.11. Thông tin về hai trạng thái có ý nghĩa của hiệu điện thế

- **Lượng thông tin** được định nghĩa bởi công thức:

$$I = \text{Log}_2(N),$$

trong đó:  $I$  – lượng thông tin tính bằng bit;  $N$  – số trạng thái có thể có.

Như vậy, lượng thông tin là số con số nhị phân cần thiết để biểu diễn số trạng thái có thể có. Do vậy, một con số nhị phân được gọi là 1 bit. Một từ  $n$  bit có thể tương trưng một trạng thái trong tổng số  $2^n$  trạng thái mà từ đó có thể tương trưng. Một từ  $n$  bit tương ứng với một lượng thông tin  $n$  bit.

### 1.3.2. Biểu diễn thông tin trong máy tính

#### a) Nguyên tắc chung

Mọi dữ liệu đưa vào máy tính đều được mã hoá thành số nhị phân.

#### • Các loại dữ liệu

- Dữ liệu nhân tạo: Do con người quy ước.
- Dữ liệu tự nhiên: Tồn tại khách quan với con người.
- Mã hoá dữ liệu nhân tạo.
- Dữ liệu số nguyên: Mã hoá theo một số chuẩn đã quy ước.
- Dữ liệu số thực: Mã hoá bằng số dấu phẩy động.
- Dữ liệu phi số (ký tự): Mã hoá theo các bộ mã hoá ký tự hiện hành như: ASCII, Unicode...





Hình 1.12. Mô hình mã hoá và tái tạo tín hiệu vật lý

• Các tín hiệu thông dụng

- Hình ảnh.
- Âm thanh.

• Thứ tự lưu trữ các byte trong máy tính

- Bộ nhớ chính tổ chức lưu trữ dữ liệu theo đơn vị byte.
- Độ dài từ dữ liệu có thể chiếm từ 1 đến 4 byte. Vì vậy, cần phải biết thứ tự chúng lưu trữ trong bộ nhớ chính đối với các dữ liệu nhiều byte.
- Có hai cách lưu trữ được đưa ra:
  - + Little endian (đầu nhỏ): Byte có ý nghĩa thấp hơn được lưu trữ trong bộ nhớ ở vị trí có địa chỉ nhỏ hơn.
  - + Big endian (đầu to): Byte có ý nghĩa cao hơn được lưu trữ trong bộ nhớ ở vị trí có địa chỉ lớn hơn.

Ví dụ: Lưu trữ một từ dữ liệu 32 bit.

$$\underbrace{0001\ 1010}_{1A} \quad \underbrace{0010\ 1011}_{2B} \quad \underbrace{0011\ 1100}_{3C} \quad \underbrace{0100\ 1101}_{4D} \quad \text{(B)}$$

$$\hspace{10em} \text{(H)}$$

Biểu diễn trong ngăn nhớ:

**Bảng 1.2. Little endian**

300	4D
301	3C
302	2B
303	1A

**Bảng 1.3. Big endian**

300	1A
301	2B
302	3C
303	4D

- **Lưu trữ của các bộ vi xử lý điển hình**

- Loại máy Intel: 80x86, Pentium → Little endian.
- Motorola 680x0 và các bộ xử lý RISC → Big endian.
- Power PC & Itanium: Tích hợp cả hai cách trên.

### b) Biểu diễn số

- **Khái niệm hệ thống số**

– Cơ sở của một hệ thống số định nghĩa phạm vi các giá trị có thể có của một chữ số.

*Ví dụ:* Trong hệ thập phân, một chữ số có giá trị từ 0 đến 9; trong hệ nhị phân, một chữ số (một bit) chỉ có hai giá trị là 0 hoặc 1.

- Dạng tổng quát để biểu diễn giá trị của một số

$$V_k = \sum_{i=m}^{n-1} b_i k^i,$$

trong đó:  $V_k$  – số cần biểu diễn giá trị;  $m$  – số thứ tự của chữ số phần lẻ (phần lẻ của số có  $m$  chữ số được đánh số thứ tự từ  $-m$  đến  $-1$ );  $n-1$  – số thứ tự của chữ số phần nguyên (phần nguyên của số có  $n$  chữ số, có thứ tự từ 0 đến  $n-1$ );  $b_i$  – giá trị của chữ số thứ  $i$ ;  $k$  – hệ số ( $k=10$ : hệ thập phân,  $k=2$ : hệ nhị phân...).

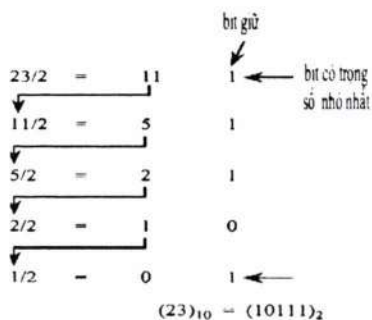
*Ví dụ:* Biểu diễn số  $541.25_{10}$ . Ta có

$$541.25_{10} = 5 \cdot 10^2 + 4 \cdot 10^1 + 1 \cdot 10^0 + 2 \cdot 10^{-1} + 5 \cdot 10^{-2} \\ = (500)_{10} + (40)_{10} + (1)_{10} + (2/10)_{10} + (5/100)_{10}.$$

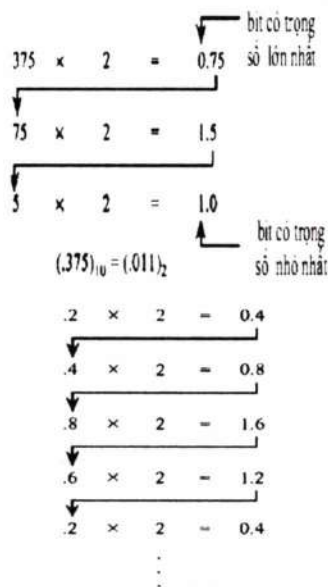
Một máy tính chủ yếu được cấu tạo bằng các mạch điện tử có hai trạng thái. Vì vậy rất tiện lợi khi dùng các số nhị phân để biểu diễn số trạng thái của các mạch điện hoặc để mã hoá các ký tự, các số cần thiết cho vận hành của máy tính. Để biến đổi một số hệ thập phân sang hệ nhị phân, ta có hai phương thức biến đổi:

- *Phương thức số dư:* Biến đổi phần nguyên của số thập phân sang nhị phân.

*Ví dụ:* Đổi  $(23.375)_{10}$  sang nhị phân. Để làm điều đó, chuyển đổi phần nguyên, dùng phương thức số dư như sau:



– Phương thức nhân để biến đổi phần lẻ của số thập phân sang nhị phân



Kết quả cuối cùng nhận được:  $23.375_{10} = 10111.011_2$ .

Tuy nhiên, trong việc biến đổi phần lẻ của một số thập phân sang số nhị phân theo phương thức nhân có một số trường hợp việc biến đổi số lặp lại vô hạn.

*Ví dụ:* Trường hợp biến đổi số nhị phân sang các hệ thống số khác ta có thể nhóm một số các số nhị phân để biểu diễn cho số trong hệ thống số tương ứng.

**Bảng 1.4.** Một số hệ thống số tương ứng với hệ nhị phân

Binary (Bas 2)	Octal (Base 8)	Decimal (Base 10)	Hexadecimal (Base 16)
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

Thông thường, người ta nhóm 4 bit trong hệ nhị phân để biểu diễn số dưới dạng thập lục phân (Hexadecimal).

Như vậy, dựa vào cách biến đổi số trong bảng nêu trên, ta có ví dụ về cách biến đổi các số trong các hệ thống số khác nhau theo hệ nhị phân:

$$1011_2 = (10_2)(11_2) = 23_4$$

$$23_4 = (2_4)(3_4) = (10_2)(11_2) = 1011_2$$

$$101010_2 = (101_2)(010_2) = 52_8$$

$$01101101_2 = (0110_2)(1101_2) = 6D_{16}$$

• **Một từ n bit** có thể biểu diễn tất cả các số dương từ 0 đến  $2^n - 1$ . Nếu  $d_i$  là một số thập phân thứ i, một từ n bit tương ứng với một số nguyên thập phân là:

$$N = \sum_{i=0}^{n-1} d_i 2^i .$$

• **Một Byte** (gồm 8 bit) có thể biểu diễn các số từ 0 đến 255 và một từ 32 bit cho phép biểu diễn các số từ 0 đến 4294967295.

• **Biểu diễn số nguyên** chia thành hai loại:

- *Số nguyên không dấu:*

Giả sử dùng n bit để biểu diễn số nguyên không dấu. Dải giá trị biểu diễn là:  $0 \div (2^n - 1)$ .

*Ví dụ:* n = 8: dải giá trị từ 0 ÷ 255;

n = 16: dải giá trị từ 0 ÷ 65535.

- *Số nguyên có dấu:*

• Số bù 1 và số bù 2: Cho một số nhị phân N được biểu diễn bởi n bit.

Ta có:

• Số bù 1 của N bằng  $(2^n - 1) - N$ ;

• Số bù 2 của N bằng  $2^n - N$ .

*Ví dụ:* Cho số  $N = 0001\ 0001_2$  được biểu diễn bởi n = 8 bit. Xác định số bù 1 và bù 2 của N.

Áp dụng công thức: 
$$\begin{array}{r} \underline{\quad 1111\ 1111} \quad (2^n - 1) \\ \quad \underline{0001\ 0001} \quad (N) \end{array}$$

Số bù 1 của N là:  $1110\ 1110$

*Nhận xét:*

• Số bù 1 của một số N được xác định bằng cách đảo các bit trong N.

• Số bù 2 của một số N được xác định bằng cách lấy số bù 1 của N cộng thêm 1:

$$\text{Số bù 2 của } N = (\text{Số bù 1 của } N) + 1.$$

+ Giả sử dùng n bit để biểu diễn số nguyên có dấu, dải mà n bit biểu diễn được là  $(-2^{n-1} \dots, -1, 0, \dots, 2^{n-1} - 1)$ . Giá trị của số nguyên đó được tính theo hai phần riêng biệt:

+ Phần giá trị dương:  $0 \div (2^{n-1} - 1)$ .

+ Phần giá trị âm:  $-2^{n-1} \div -1$ .

Trong đó bit có trọng số cao nhất (hay bit ngoài cùng bên trái của dãy nhị phân được máy tính sử dụng để biểu diễn dấu của giá trị):

• Nếu bằng 0 thì số nhị phân cần tính có giá trị là số dương và có dạng tổng quát là  $0a_{n-2}a_{n-3}...a_0$ ;

• Nếu bằng 1 thì số nhị phân cần tính có giá trị là số âm và có dạng tổng quát là  $1a_{n-2}a_{n-3}...a_0$ .

### • Biểu diễn số thực

Ví dụ:

Khối lượng Mặt Trời:

19900g;

Khối lượng điện tử:

0.000910956g.

Biểu diễn số trên trong máy tính nếu sử dụng cách biểu diễn số có dấu phẩy tĩnh rất bất tiện, vì vậy sử dụng số có dấu phẩy động.

Biểu diễn cho số thực (Floating Point Number) có dạng

$$X = M \cdot R^E,$$

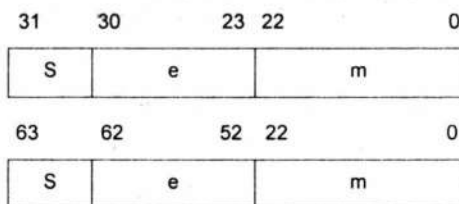
trong đó: X – số thực; M – phần định trị; R – cơ số (2); E – phần mũ.

– Chuẩn IEEE754/85, gồm:

+ Số có dấu phẩy động chính xác đơn 32 bit.

+ Số có dấu phẩy động chính xác kép 64 bit.

+ Số có độ chính xác mở rộng dài 128 bit (quadruple).



**Hình 1.13.** Chuẩn IEEE754/85



Trong đó: S – bit dấu;  $e = E + 127 \Rightarrow E = e - 127$ ; m – phần lẻ của phần định trị M;  $M = 1.m$ .

Giá trị số thực là:  $X = (-1)^S e * 1.m * 2^{127}$ .

– Cách biểu diễn bằng số thừa K: Trong cách này, số dương của một số N có được bằng cách "cộng thêm vào" số thừa K được chọn sao cho tổng của K và một số âm bất kỳ luôn luôn dương. Số âm -N của số N có được bằng cách lấy K - N (hay lấy bù 2 của số vừa xác định).

*Ví dụ:* Số thừa K = 128, số "cộng thêm vào" 128 là một số nguyên dương. Số âm là số lấy bù 2 số vừa tính, bỏ qua số giữ bit cao nhất:

$$+2510 = 10011001_2 - 2510 = 01100111_2.$$

Dùng 1 Byte (8 bit) để biểu diễn một số có dấu lớn nhất là +127 và số nhỏ nhất là -128.

Chỉ có một giá trị 0:  $+0 = 10000000_2$ ,  $-0 = 00000000_2$ .

– Cách biểu diễn số nguyên có dấu bằng số bù 2 được dùng rộng rãi cho các phép tính số nguyên. Nó có lợi là không cần thuật toán đặc biệt nào cho các phép tính cộng và tính trừ, và giúp phát hiện dễ dàng các trường hợp bị tràn.

– Các cách biểu diễn bằng "dấu, trị tuyệt đối" hoặc bằng "số bù 1" dẫn đến việc dùng các thuật toán phức tạp và bất lợi vì luôn có hai cách biểu diễn của số không. Cách biểu diễn bằng "dấu, trị tuyệt đối" được dùng cho phép nhân của số có dấu phẩy động.

Cách biểu diễn bằng số thừa K được dùng cho số mũ của các số có dấu phẩy động.

Cách này làm cho việc so sánh các số mũ có dấu khác nhau trở thành việc so sánh các số nguyên dương.

### • Các phép toán với số dấu phẩy động

– Tăng số mũ của số có số mũ nhỏ hơn bằng với số mũ của số có số mũ lớn.

– Cộng hoặc trừ phần định trị.

– Chuẩn hoá lại kết quả.

Trước khi đi vào cách biểu diễn số với dấu phẩy động, ta xét cách biểu diễn một số dưới dạng dấu phẩy tĩnh.

*Ví dụ:* Trong hệ thập phân số  $254_{10}$  có thể biểu diễn dưới các dạng sau:

$$254 * 10^0; 25.4 * 10^1; 2.54 * 10^2; 0.254 * 10^3; 0.0254 * 10^4; \dots$$

Trong hệ nhị phân, số  $(0.00011)_2$  tương đương với số  $(0.09376)_{10}$  có thể biểu diễn dưới các dạng:

$$0.00011; 0.00011 * 2^0; 0.0011 * 2^{-1}; 0.011 * 2^{-2}; 0.11 * 2^{-3}; 1.1 * 2^{-4}.$$

Các cách biểu diễn này gây khó khăn trong một số phép so sánh các số. Để dễ dàng trong các phép tính, các số được chuẩn hoá về một dạng biểu diễn:

$$\pm 1.f\text{ff}...f \times 2^{\pm E},$$

trong đó:  $f$  – phần lẻ;  $E$  – phần mũ.

Số dấu phẩy động được chuẩn hoá cho phép biểu diễn gần đúng các số thập phân rất lớn hay rất nhỏ dưới dạng một số nhị phân theo một dạng quy ước. Thành phần của số phẩy động bao gồm: phần dấu, phần mũ và phần định trị.

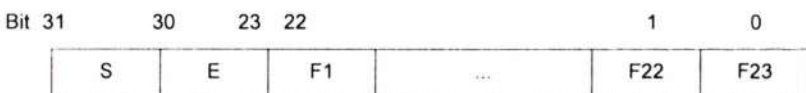
Như vậy, cách này cho phép biểu diễn gần đúng các số thực, tất cả các số đều có cùng cách biểu diễn. Có nhiều cách biểu diễn dấu phẩy động, trong đó cách biểu diễn theo chuẩn IEEE754 được dùng rộng rãi trong khoa học máy tính hiện nay. Trong cách biểu diễn này, phần định trị có dạng  $1, f$  với số 1 ẩn tăng và  $f$  là phần số lẻ.

Chuẩn IEEE 754 định nghĩa hai dạng biểu diễn số dấu phẩy động:

– Số phẩy động chính xác đơn với định dạng được định nghĩa: Chiều dài số 32 bit được chia thành các trường: dấu  $S$  (Sign bit – 1 bit), mũ  $E$  (Exponent – 8 bit), phần lẻ  $F$  (Fraction – 23 bit).

Số này tương ứng với số thực:

$$(-1)^S * (1, f_1 f_2 \dots f_{23}) * 2^{(E-127)}.$$



**Hình 1.14.** Biểu diễn số có dấu phẩy động chính xác đơn với 32 bit



– Số dấu phẩy động chính xác kép với định dạng được định nghĩa: Chiều dài số 64 bit được chia thành các trường: dấu S (Sign bit – 1 bit), mũ E (Exponent – 11 bit), phần lẻ F (Fraction – 52 bit).

$$\text{Số này tương ứng với số thực } (-1)^S * (1, f_1 f_2 \dots f_{52}) * 2^{(E-1023)}$$

Bit 63 62 52 51 1 0



**Hình 1.15.** Biểu diễn số có dấu phẩy động chính xác kép với 64 bit

Để thuận lợi trong một số phép tính toán, IEEE định nghĩa một số dạng mở rộng của chuẩn IEEE 754.

**Bảng 1.5.** Một số dạng mở rộng của chuẩn IEEE 754

Tham số	Chính xác đơn	Mở rộng chính xác đơn	Chính xác kép	Mở rộng chính xác kép
Chiều dài bit	32	≥ 43	64	≥ 79
Chiều dài trường mũ (E)	8	≥ 11	11	≥ 15
Số thừa	127	–	1023	≥ 15
Giá trị mũ tối đa	127	≥ 1023	1023	≥ 16383
Giá trị mũ tối thiểu	-126	≤ -1022	-1022	≤ -16382
Chiều dài trường lẻ (F)	23	≥ 31	52	≥ 63

Chuẩn IEEE 754 cho phép biểu diễn các số chuẩn hoá (các bit của E không cùng lúc bằng 0 hoặc bằng 1), các số không chuẩn hoá (các bit của E không cùng lúc bằng 0 và phần số lẻ f1 f2... khác không), trị số 0 (các bit của E không cùng lúc bằng 0 và phần số lẻ bằng không), và các ký tự đặc biệt (các bit của E không cùng lúc bằng 1 và phần lẻ khác 0).

**c) Biểu diễn ký tự**

• **BCD (Binary Coded Decimal):** Mã hoá các số thập phân bằng các ký hiệu nhị phân.

**Bảng 1.6.** Các ký hiệu nhị phân mã hoá các số thập phân

Thập phân	BCD	Thập phân	BCD
0	0000	8	1000
1	0001	9	1001
2	0010	Cấm	1010
3	0011	Cấm	1011
4	0100	Cấm	1100
5	0101	Cấm	1101
6	0110	Cấm	1110
7	0111	Cấm	1111

• Mã ASCII (American Standard Code for Information Interchange)

Biểu diễn các ký tự hiển thị được và ký tự điều khiển:

**Bảng 1.7.** Bảng mã ASCII tiêu chuẩn

Decimal	Hex	ASCII	Decimal	Hex	ASCII	Decimal	Hex	ASCII	Decimal	Hex	ASCII
0	0	NUL	32	20		64	40	@	96	60	·
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	\$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(	72	48	H	104	68	h
9	9	HT	41	29	)	73	49	I	105	69	i
10	A	LF	42	2A	*	74	4A	J	106	6A	j
11	B	VT	43	2B	+	75	4B	K	107	6B	k
12	C	FF	44	2C	,	76	4C	L	108	6C	l
13	D	CR	45	2D	-	77	4D	M	109	6D	m
14	E	SOH	46	2E	.	78	4E	N	110	6E	n
15	F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	□

Bảng 1.8. Bảng mã ASCII mở rộng

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ā	160	A0	ā	192	C0	Ł	224	E0	α
129	81	ū	161	A1	ī	193	C1	ł	225	E1	β
130	82	ē	162	A2	ō	194	C2	ŧ	226	E2	Γ
131	83	ā	163	A3	ū	195	C3	ł	227	E3	π
132	84	ā	164	A4	ñ	196	C4	—	228	E4	Σ
133	85	ā	165	A5	Ñ	197	C5	+	229	E5	σ
134	86	ā	166	A6	*	198	C6	ł	230	E6	μ
135	87	ċ	167	A7	◊	199	C7	ł	231	E7	ι
136	88	ē	168	A8	ċ	200	C8	ł	232	E8	ϕ
137	89	e	169	A9	ƒ	201	C9	ƒ	233	E9	θ
138	8A	e	170	AA	→	202	CA	ł	234	EA	Ω
139	8B	i	171	AB	½	203	CB	ƒ	235	EB	δ
140	8C	i	172	AC	¼	204	CC	ł	236	EC	∞
141	8D	i	173	AD	ı	205	CD	=	237	ED	∞
142	8E	À	174	AE	«	206	CE	ł	238	EE	ε
143	8F	Ā	175	AF	»	207	CF	ł	239	EF	∩
144	90	Ē	176	B0	░	208	DO	ł	240	FO	≡
145	91	æ	177	B1	▒	209	D1	ƒ	241	F1	±
146	92	Æ	178	B2	▓	210	D2	ƒ	242	F2	≥
147	93	ó	179	B3		211	D3	ł	243	F3	≤
148	94	o	180	B4	┆	212	D4	ł	244	F4	[
149	95	o	181	B5	┆	213	D5	ƒ	245	F5	]
150	96	ù	182	B6	ł	214	D6	ƒ	246	F6	-
151	97	ù	183	B7	ŋ	215	D7	ł	247	F7	~
152	98	y	184	B8	ŋ	216	D8	+	248	F8	*
153	99	Ō	185	B9	ł	217	D9	ł	249	F9	.
154	9A	U	186	BA	ł	218	DA	ƒ	250	FA	.
155	9B	◊	187	BB	ŋ	219	DB	▓	251	FB	√
156	9C	Ł	188	BC	ł	220	DC	▓	252	FC	~
157	9D	Ÿ	189	BD	ł	221	DD		253	FD	ε
158	9E	Ł	190	BE	ł	222	DE		254	FE	▓
159	9F	ł	191	BF	ł	223	DF	▓	255	FF	□

#### d) Các phép toán luận lý

Ba phép toán thông thường trong nhóm của các phép toán luận lý đó là AND, OR và EXCLUSIVE OR (XOR). Chúng tương tự như phép cộng và trừ với hai toán hạng và trả về một kết quả duy nhất (Trái lại có một số phép toán mà giá trị trả về của nó sẽ cho ra hai số khác nhau như là phép lấy căn bậc hai, ví dụ như 4 khi lấy căn bậc hai sẽ cho hai kết quả là 2 và -2). Bây giờ ta sẽ xem qua một số phép toán như sau:

### • Phép AND

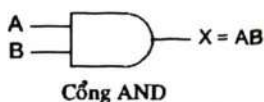
Nếu hai biến logic A và B được kết hợp qua cổng AND, kết quả là:

$$X = A.B.$$

Bảng giá trị chân lý của phép AND (phép nhân) hai biến A và B như bảng 1.9.

**Bảng 1.9.** Bảng giá trị chân lý của phép AND 2 đầu vào

A	B	$X = A.B$
0	0	0
0	1	0
1	0	0
1	1	1



**Hình 1.16.** Ký hiệu cổng AND 2 đầu vào

*Kết luận:*

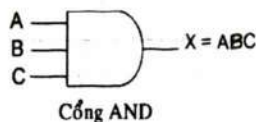
– Phép toán AND có kết quả bằng 0 nếu một hay nhiều biến đầu vào bằng 0.

– Cổng AND chỉ có 1 đầu ra và có nhiều hơn 2 đầu vào.

*Ví dụ:* Cổng AND 3 đầu vào có bản sự thật như sau:

**Bảng 1.10.** Bảng giá trị chân lý của phép AND 3 biến

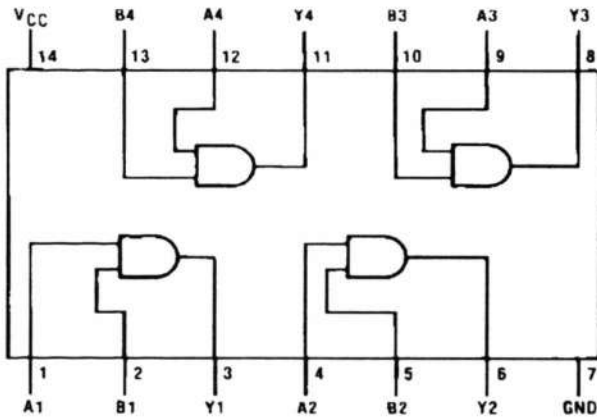
A	B	C	$X = ABC$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



**Hình 1.17.** Cổng AND 3 đầu vào

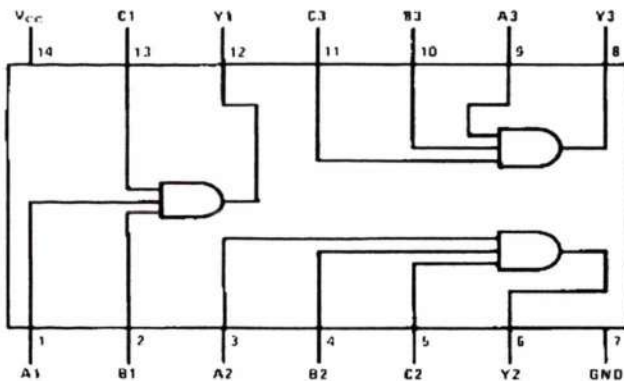
Các IC số:

– DM 7408: IC AND 2 đầu vào.



Hình 1.18. Sơ đồ của IC DM7408

– DM74LS11: IC AND 3 đầu vào.



Hình 1.19. Sơ đồ của IC DM74LS11

• Phép OR

A và B là hai biến độc lập. Khi A và B kết hợp qua phép OR, kết quả X được mô tả như sau:

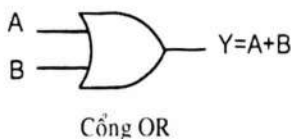
$$X = A + B.$$



Trong biểu thức này dấu "+" không có nghĩa là phép cộng thuần túy. Nó là phép OR, kết quả của phép OR được cho trong bảng giá trị chân lý như bảng 1.11.

**Bảng 1.11.** Bảng giá trị chân lý phép OR 2 đầu vào

A	B	$X = A + B$
0	0	0
0	1	1
1	0	1
1	1	1



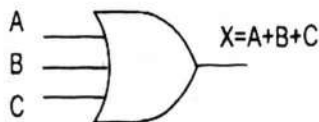
**Hình 1.20.** Ký hiệu cổng OR 2 đầu vào

*Kết luận:*

- Phép toán OR sẽ có kết quả bằng 1 nếu một hay nhiều biến đầu vào bằng 1.
- Cổng OR chỉ có 1 đầu ra và có thể có nhiều hơn 2 đầu vào.
- Ký hiệu và bảng giá trị chân lý của cổng OR 3 đầu vào như hình 1.21.

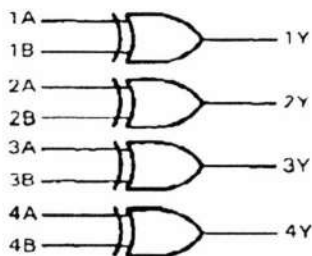
**Bảng 1.12.** Bảng giá trị chân lý phép OR 3 đầu vào

A	B	C	$X = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



**Hình 1.21.** Cổng OR 3 đầu vào

- IC DM74LS32 (OR 2 đầu vào)



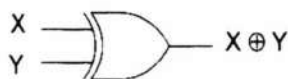
Hình 1.22. Sơ đồ chân của IC DM4LS32

• Phép EXCLUSIVE OR (XOR)

Phép toán XOR (ký hiệu  $\oplus$ ) có bảng giá trị chân lý bảng 1.13.

Bảng 1.13. Bảng giá trị chân lý phép XOR 2 đầu vào

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

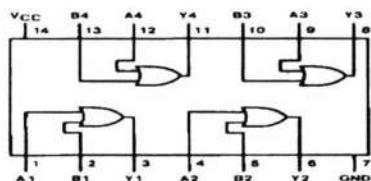
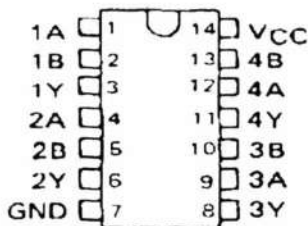


Hình 1.23. Cổng XOR 2 đầu vào

- Từ bảng giá trị chân lý ta nhận thấy rằng:  $X \oplus Y = 1$  khi  $X \neq Y$  và  $X \oplus Y = 0$  khi  $X = Y$ .

- Biểu thức toán của phép toán XOR:  $X \oplus Y = \overline{X}Y + X\overline{Y}$ .

- IC SN74136



Hình 1.24. Sơ đồ chân của IC SN7413

### • Các phép chuyển dịch và quay

Các phép toán thuộc lớp các phép toán như phép quay (rotation) và phép dịch chuyển (shift), đều có ý nghĩa biến đổi các bit trong một thanh ghi và thường được sử dụng để giải quyết các bài toán thực hiện trên bit. Ví dụ như biến đổi một byte theo một yêu cầu nào đó bằng cách sử dụng mặt nạ, hoặc thao tác trên phần định trị của các số biểu diễn ở dạng dấu phẩy động. Những phép toán này được phân chia tùy theo hướng di chuyển của các dãy bit (sang trái hay sang phải).

Cho một byte gồm 8 bit, nếu ta thực hiện phép toán SHIFT cho dãy bit của nó sang hướng trái hay phải thì bit đầu tiên của byte (là bit cao nhất nếu dịch chuyển sang phải, hay bit thấp nhất khi dịch chuyển sang trái) sẽ bị chuyển đi, và bit cuối cùng của nó (là bit cao nhất nếu dịch chuyển sang trái, hay bit thấp nhất khi dịch chuyển sang phải) sẽ được đặt là 0.

*Ví dụ:* Cho một byte có giá trị là 10001110, khi SHIFT trái một lần sẽ là 00011100 hoặc SHIFT phải kết quả là 01000111.

Đối với phép toán quay, cũng giống như phép SHIFT. Nhưng bit cuối sẽ được chuyển vào bit đầu tiên.

*Ví dụ:* Cho một byte có giá trị là 10001110, khi ta quay trái một bit thì kết quả sẽ là 00011101; quay phải một bit thì kết quả sẽ là 01000111.

Phép toán SHIFT thường được sử dụng cho các phép nhân hay chia cho 2, đối với SHIFT trái chính là nhân cho 2, và SHIFT phải là chia cho 2. Do đó phép toán này ta gọi là phép chuyển số học (arithmetic Shifts).

### 1.3.3. Bộ cộng, bộ trừ

#### a) Bộ cộng, trừ số nhị phân

##### • Bộ cộng (Thực hiện cộng hai số nhị phân)

– *Bộ cộng nửa* (Half adder – HA):

Gọi A là số được cộng; B là số cộng; S là tổng của A và B; C là số nhớ ra từ phép cộng.

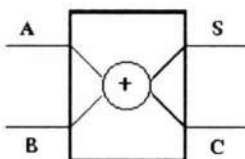
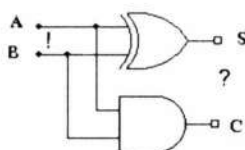
Xét mạch cộng nhị phân 1 bit: Với tổ hợp 4 trạng thái logic của A và B thì trạng thái logic đầu ra S và C.

Rút gọn biểu thức logic ta có:  $S = A \oplus B$ ;  $C = AB$ .

S là đầu ra của cổng XOR 2 đầu vào A và B; còn C là đầu ra của cổng AND.

**Bảng 1.14.** Bảng giá trị chân lý của phép cộng nửa

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

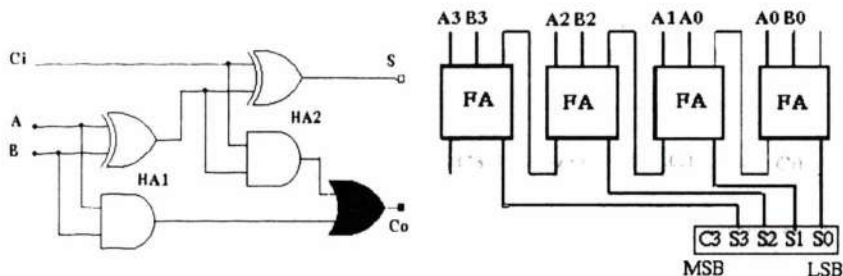


**Hình 1.25.** Mạch cộng nửa

Cấu trúc logic của mạch: Đây là mạch cộng nửa hay cộng chưa đủ vì chưa có số nhớ ban đầu của phép cộng trước đó.

– Bộ cộng đủ (Full adder – FA):

Giả sử mạch đã thực hiện phép cộng lần đầu rồi nên được tổng là  $S_0$  và số nhớ  $C_0$ , nếu tiếp tục cộng lần 2 khi trạng thái logic của A và B thay đổi thì S không chỉ là tổng của A và B mà gồm cả  $C_0$  trước đó. Mạch cộng đủ như hình 1.26.



**Hình 1.26.** Mạch cộng đủ

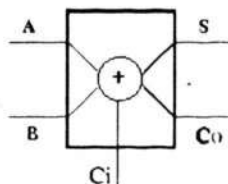
Rút gọn biểu thức ta được:

$$S = C_0 \oplus (A \oplus B);$$

$$C = AB + (A \oplus B).$$

**Bảng 1.15.** Bảng giá trị chân lý của phép cộng đủ

A	B	C <sub>i</sub>	S	C <sub>o</sub>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



**Hình 1.27.** Mạch cộng đủ

Cấu trúc của mạch logic của mạch cộng đủ như hình 1.27.

FA thực ra bao gồm trong nó HA và cổng OR ở đầu ra cho số nhớ C. Mạch HA ở trên mới chỉ làm phép cộng 1 bit.

– Bộ cộng nhiều bit song song:

Nếu 2 số cộng có nhiều bit hơn thì cách cộng cũng tương tự, trước hết cộng 2 bit LSB để cho bit tổng (LSB). Số nhớ được đưa tới để cộng chung với 2 bit kế tiếp bit LSB để cho bit tổng ở hàng kế tiếp cho đến phép cộng cuối cùng giữa 2 bit MSB để được bit tổng hàng đó, số nhớ bây giờ trở thành bit LSB của tổng. Đây là một mạch cộng song song, vì các hàng được cộng cùng một lúc, tuy nhiên như cấu trúc mạch ở trên thì các bit ra của tổng không phải là đồng thời, bởi vì các phép cộng ở các bit cao thì chậm hơn do phải chờ bit nhớ ở phép cộng trước đưa tới. Tức là đã có trì hoãn làm giảm tính đồng bộ của mạch. Nếu thêm vào mạch cho phép cung cấp sẵn các bit nhớ để phục vụ cho các phép cộng ở các hàng được cùng lúc thì sẽ khắc phục được nhược điểm này.

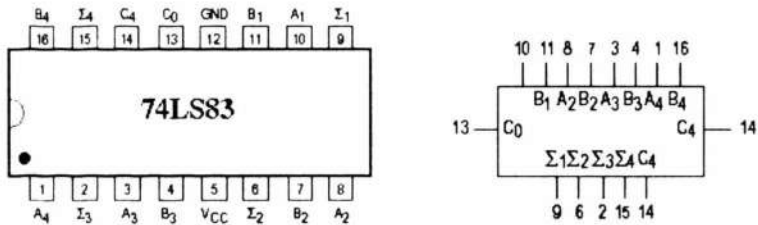


Với công nghệ tích hợp cao, việc thêm mạch cung cấp sẵn các bit nhớ trở nên dễ dàng hơn, khi đó mạch trở thành mạch cộng có số nhớ nhìn trước.

Một số IC làm phép cộng là:

7480 (1 bit), 7482 (2 bit), 7483/LS83/283 (4 bit).

Ví dụ: Xét qua IC 74LS83 (hình 1.28).



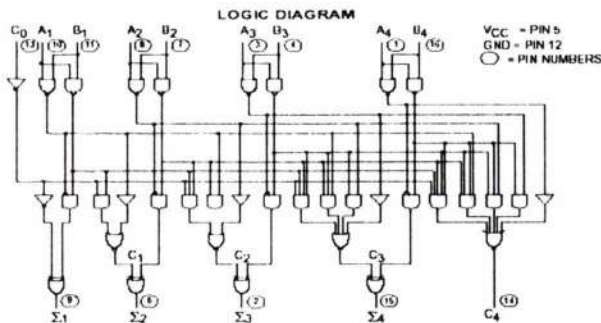
Hình 1.28. Ký hiệu khối và chân ra 74LS83

Trong đó : 2 số 4 bit vào là  $A_4A_3A_2A_1$  và  $B_4B_3B_2B_1$ ; số nhớ ban đầu là  $C_0$ .

Vậy, tổng ra sẽ là  $C_4S_4S_3S_2S_1$ , với  $C_4$  là số nhớ của phép cộng.

Bảng 1.16. Bảng giá trị chân lý của mạch cộng 4 bit 74LS83

Số nhớ đầu $C(n-1)$	Số cộng		Tổng $\Sigma_n$	Số nhớ sau $C_n$
	$A_n$	$B_n$		
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Hình 1.29. Mạch logic của 74LS83

Có thể nối chồng IC cộng lại với nhau để cho số bit gấp đôi. Khi đó bit MSB ( $C_4$ ) của tầng đầu được nối tới đầu vào nhớ ban đầu ( $C_0$ ) của tầng sau.

– *Bộ cộng nhiều bit nối tiếp* (Bộ cộng hai số nhị phân 4 bit):

Ngoài cách cộng song song còn có một dạng mạch cộng số nhiều bit nữa gọi là mạch cộng nối tiếp. Khi đó 2 bit LSB của các số được cộng trước, bit LSB của tầng được đưa ra 1 ghi dịch, còn số nhớ sẽ quay trở về cộng chung với 2 bit kế tiếp bit LSB và cứ vậy cho đến 2 bit cuối cùng được cộng. Mạch ghi dịch đầu ra dịch chuyển sang phải qua mỗi lần cộng sẽ cho ra kết quả cộng số nhớ cuối cùng trở thành bit MSB của tổng ra.

Mạch thực hiện phép tính chậm hơn so với cộng song song, nó cũng cần 1 xung nhịp để giữ cho các mạch làm việc đồng bộ.

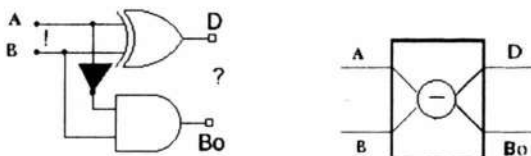
#### • Bộ trừ

– *Mạch trừ nửa và trừ đủ:*

Mạch này cũng gồm 2 loại mạch trừ nửa hay bán phần (half subtractor – HS) và trừ đủ hay toàn phần (full subtractor – FS, khi đó cần bit mượn  $B_1$  trước tham gia vào phép tính).

**Bảng 1.17.** Bảng giá trị chân lý của phép trừ nửa

A	B	D	Bo
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

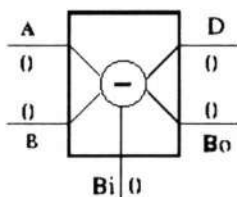


**Hình 1.30.** Mạch trừ nửa

Phép trừ thực ra là phép cộng với số âm. Để có số âm của một số ta lấy bù 1 của số đó, còn khi thực hiện mạch thì đó là cổng đảo.

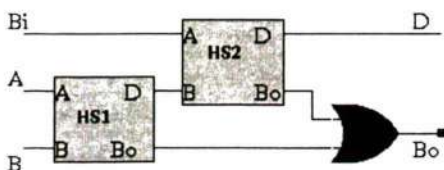
**Bảng 1.18.** Bảng giá trị chân lý của phép trừ đừ

A	B	B <sub>i</sub>	D	B <sub>o</sub>
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



**Hình 1.31.** Mạch trừ đừ

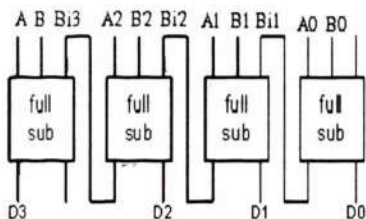
Cấu trúc mạch trừ cũng tương tự mạch cộng, chỉ khác là số bị trừ B cần phải qua cổng đảo khi thực hiện AND với số trừ A để cho số mượn R. Mạch trừ FS cũng gồm 2 mạch trừ HS và cổng OR ở đầu ra cho số mượn B<sub>o</sub>.



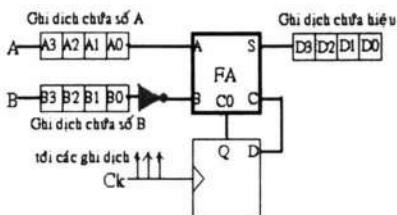
**Hình 1.32.** Mạch trừ FS

– Mạch trừ nhiều bit:

Với mạch trừ nhị phân nhiều bit cũng có thể thực hiện song song các mạch cộng FS từng bit, nhưng các bit của số bị trừ cần được đảo, số nhớ của tầng cuối cần đem về bit nhớ ban đầu của tầng đầu.



**Hình 1.33.** Mạch trừ 4 bit song song



**Hình 1.34.** Mạch trừ 4 bit nối tiếp

### • Mạch cộng, trừ kết hợp

Nếu thêm vào một số công logic cần thiết ta đã có một mạch có thể cộng hay trừ tùy theo đầu vào điều khiển CT.

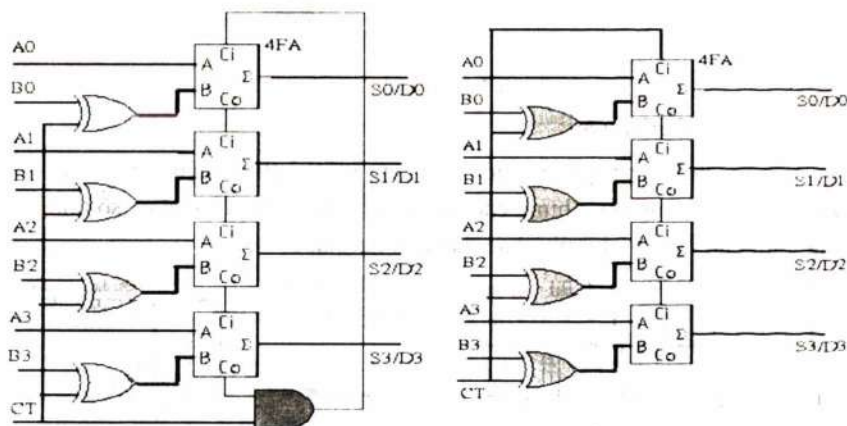
Khi  $CT = 0$ , các cổng EXOR có 1 ngõ ở thấp nên cho số B qua không bị đảo, tức là mạch thực hiện phép cộng.

Khi  $CT = 1$ , các cổng EXOR có 1 ngõ ở cao nên hoạt động như một cổng NOT, số B bị đảo, khi đó mạch thực hiện phép cộng  $A + (-B)$ , tức là phép trừ.

$C_0,3$  là bit LSB của tổng được vòng trở lại (qua cổng AND) về  $C_0$ ; sẽ cho phép cộng nhiều bit.

Ngoài cách dùng bù 1 cũng có thể dùng bù 2 (lấy bù 1 rồi cộng thêm 1) để thực hiện phép toán trừ nhị phân kể cả số có dấu. Cách này được sử dụng phổ biến ở VXL và máy tính.

Mạch khá giống như nó ở cách dùng bù 1, nhưng bit nhớ ra cuối cùng không cần đem về tầng đầu. Tổng hay hiệu ra ở dạng bù 2, muốn lấy đúng kết quả thì phải chuyển trở lại. Khi đó mạch cộng, trừ nhị phân 4 bit dùng bù 2 như hình 1.35.



Hình 1.35. Mạch cộng, trừ dùng 1 bit và 2 bit

**b) Cộng, trừ BCD**

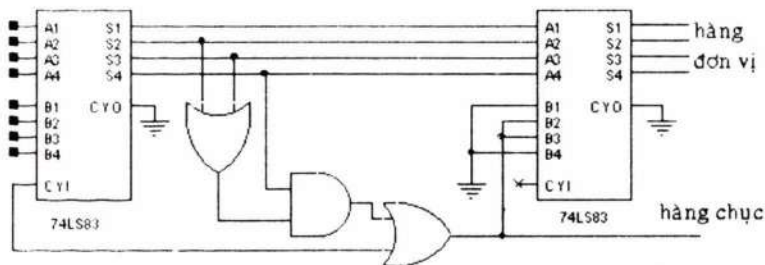
**• Cộng hai số BCD**

Số BCD thực ra cũng là số nhị phân n bit nhưng chỉ có 10 tổ hợp trạng thái từ 0000 đến 1001 (biểu thị số thập phân tương ứng là từ 0 đến 9) nên cách cộng cũng tương tự như cộng số nhị phân nhiều bit.

0011 (3)	0110 (6)
+ 0110 (5)	+ 1000 (8)
1001 (9)	1110 (14)
số BCD	+ 0110 (6)
	0001 0100 (1 4)
	số BCD 2 hàng

Tuy nhiên, khi tổng vượt quá 1001 tức là tổng đó không còn là số BCD nữa, do đó ta phải cộng tổng với 0110 (số 6) để cho tổng mới là số BCD, đồng thời số nhớ chính là hàng cao hơn của tổng.

*Ví dụ:* Cộng hai số BCD là A3A2A1A0 và B3B2B1B0 cho tổng là C3S4S3S2S1 (C3 là hàng chục nếu có).



**Hình 1.36. Mạch cộng hai số BCD 1 bit**

Nếu tổng đầu vượt quá 9 (từ 10 đến 18) thì các cổng logic sẽ cho phép xác định hàng chục, đồng thời tổng này phải được cộng thêm 6 ở tầng 74LS83 thứ hai để cho tổng cuối cùng ở dạng BCD.

Nếu tổng không vượt quá 9 (vẫn là số BCD) thì tổng hàng chục không có nên 74LS83 thứ hai sẽ cộng tổng này với 0, tổng ra không có gì thay đổi.

Ta có thể ghép nhiều mạch cộng ở trên để có mạch cộng hai số BCD nhiều bit, khi đó chỉ việc nối đầu ra hàng chục của tầng đầu tới đầu vào số nhớ C<sub>i</sub> của tầng sau là được.

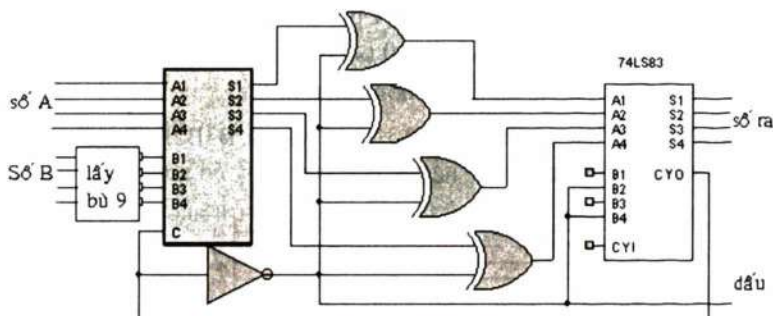
**• Trừ hai số BCD**

Với phép trừ BCD, ta phải lấy bù 9 của số trừ rồi mới làm phép cộng lại với số bị trừ. Lấy bù 9 của 1 số tức là lấy 9 trừ đi số đó.



Ví dụ: Phép trừ BCD: 9 – 5 và 2 – 6

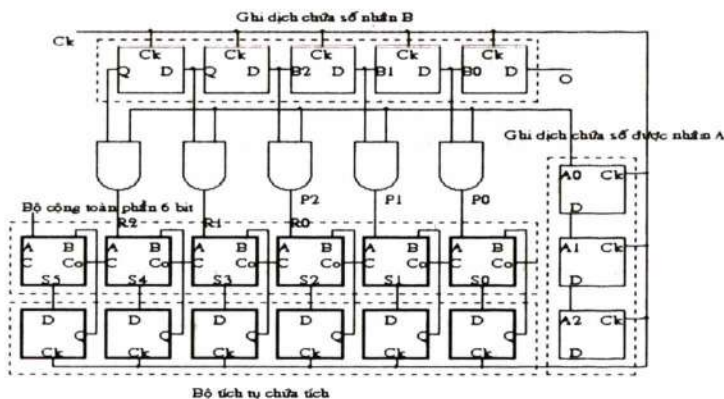
	1001 (9)		0010 (2)
	+ 0100 (bù 9 của 5)		+ 0001 (bù 9 của 8)
	1101 (# BCD)		0011 (3)
số nhũ trên 1	0110 (6)		bù 9 của 3 là 6 tức 0011
	+ 1		
	0100 (4)		do đã lấy bù 9 nên
	số BCD		kết quả cuối cùng phải là -6



Hình 1.37. Mạch trừ hai số BCD 1 bit

**c) Mạch nhân hai số nhị phân**

Về nguyên lý, đó là phép cộng nhiều lần. Mạch ở hình 1.38 minh hoạ cho mạch nhân hai số nhị phân 4 bit A và B.



Hình 1.38. Cấu trúc mạch nhân hai số nhị phân

Mạch gồm ghi dịch 4 bit để chứa số được nhân A, ghi dịch 5 bit để chứa số nhân B, ghi dịch 6 bit để chứa kết quả nhân (còn gọi là bộ tích lũy: accumulator). 5 cổng AND sẽ tạo tích từng phần của các cặp bit và 6 bộ cộng đủ để tạo tích cuối và số nhớ tương ứng.

*Ví dụ:* Phép nhân hai số  $A = 1101$  và  $B = 1011$ .

Số nhân (A): 1101 ( $13_{10}$ )

Số bị nhân B: 1011 ( $11_{10}$ )

Tích 1101

từng 1101

phần 0000

1101

Tích cuối: 10001111 ( $143_{10}$ ).

## 1.4. SƠ LƯỢC CÁC BỘ PHẦN CỦA HỆ THỐNG MÁY TÍNH

### 1.4.1. Cấu trúc chung của máy vi tính

Máy vi tính là một hệ thống được ghép nhiều thành phần tạo nên. Do đó để máy tính có thể hoạt động được ta phải lắp ghép các thành phần của nó một cách hợp lý và khai báo với các thành phần khác. Ngày nay ngành Tin học dựa trên các máy tính hiện đang phát triển trên cơ sở hai phần:

- **Phần cứng**

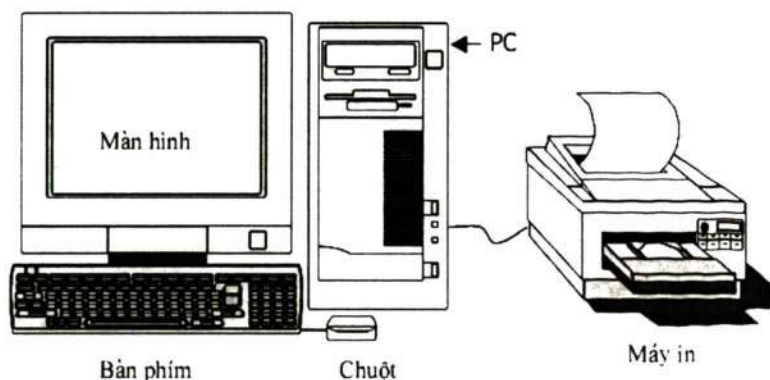
Phần cứng là những đối tượng vật lý hữu hình như vi mạch, bản mạch in, dây cáp nối mạch điện, bộ nhớ, màn hình, máy in, thiết bị đầu - cuối, nguồn nuôi... Phần cứng thực hiện các chức năng xử lý thông tin cơ bản ở mức thấp nhất, tức là tín hiệu nhị phân.

- **Phần mềm**

Phần mềm là các chương trình (Program) điều phối các hoạt động phần cứng của máy vi tính và chỉ đạo việc xử lý số liệu. Phần mềm của máy vi tính có thể chia làm hai loại phần mềm hệ thống (System Software) và phần mềm ứng dụng (Application software). Phần mềm hệ thống khi được đưa vào bộ nhớ chính, nó điều khiển máy tính thực hiện các công việc. Phần

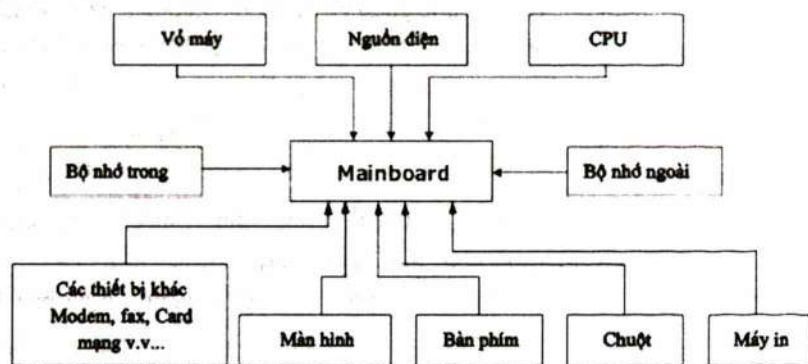
mềm ứng dụng là các chương trình thiết kế để giải quyết một bài toán hay một vấn đề cụ thể để đáp ứng một nhu cầu riêng trong một số lĩnh vực.

Máy tính cá nhân PC (Personal Computer): Theo đúng tên gọi của nó là máy tính có thể được sử dụng bởi riêng một người.



Hình 1.39. Hệ thống máy vi tính thường được sử dụng

#### 1.4.2. Các thành phần cơ bản của máy vi tính



Hình 1.40. Sơ đồ tổng quan về cấu trúc máy tính

- **Vỏ máy:** Nơi để gắn các thành phần của máy tính thành khối như nguồn, Mainboard, card... có tác dụng bảo vệ máy tính.

- **Nguồn điện:** Cung cấp hầu hết hệ thống điện cho các thiết bị bên trong máy tính.

- **Mainboard:** Có chức năng liên kết tất cả các thành phần tạo nên máy tính và là bang mạch lớn nhất trên máy tính.

- **CPU (Central Processing unit):** Bộ vi xử lý chính của máy tính.

- **Bộ nhớ trong (ROM, RAM):** Nơi lưu trữ dữ liệu và chương trình phục vụ trực tiếp cho việc xử lý của CPU, nó giao tiếp với CPU thông qua một thiết bị trung gian.

- **Bộ nhớ ngoài:** Nơi lưu trữ dữ liệu, chương trình một cách gián tiếp cho việc xử lý của CPU, bao gồm các loại đĩa cứng, đĩa mềm, CDROM... Khi giao tiếp với CPU nó phải thông qua thiết bị trung gian (thường là RAM) hay gọi tắt là ngắt.

- **Màn hình:** Thiết bị đưa thông tin ra ngoài giao diện trực tiếp với người dùng. Đây là thiết bị xuất chuẩn của máy vi tính hay còn gọi là bộ trực (Monitor).

- **Bàn phím (Keyboard):** Thiết bị nhập tin vào giao diện trực tiếp với người dùng. Đây là thiết bị nhập chuẩn của máy vi tính.

- **Chuột (Mouse):** Thiết bị điều khiển trong môi trường đồ họa giao diện trực tiếp với người sử dụng.

- **Máy in (Printer):** Thiết bị xuất thông tin ra giấy thông dụng nhất.

- **Các thiết bị** như Card mạng, Modem, máy Fax... phục vụ cho việc lắp đặt mạng máy và các chức năng khác.

- **Ổ đĩa cứng:** Bộ nhớ dữ liệu chính của máy tính cá nhân, các thành quả của một quá trình làm việc có thể được lưu trữ trên ổ đĩa cứng trước khi có các hành động sao lưu dự phòng trên các dạng bộ nhớ khác.

- **Ổ đĩa quang (CD, DVD):** Bộ nhớ dùng cho xuất/nhập dữ liệu với dung lượng lớn hoặc trao đổi dữ liệu, phần mềm với những máy tính khác. Sử dụng sao lưu dữ liệu và các mục đích khác. Đây không phải là thiết bị bắt buộc đối với hệ thống phần cứng máy tính cá nhân.

- **Ổ đĩa mềm:** Bộ nhớ dùng cho xuất/nhập dữ liệu với dung lượng thấp (phụ thuộc từng loại đĩa mềm). Đây không phải thiết bị bắt buộc phải có.

- **Bo mạch đồ họa:** Thiết bị có chức năng xuất hình ảnh ra màn hình, giúp người sử dụng giao tiếp với máy tính.

- **Bo mạch âm thanh:** Thiết bị có chức năng xuất tín hiệu âm thanh ra các thiết bị phát âm thanh (loa). Đây không phải thiết bị bắt buộc phải có.

- **Bo mạch mạng:** Thiết bị có chức năng kết nối các máy tính với nhau thành một mạng máy tính, giúp máy tính có thể trao đổi thông tin với các máy tính khác trên phạm vi rộng (có thể đến toàn thế giới). Đây không phải thiết bị bắt buộc phải có.

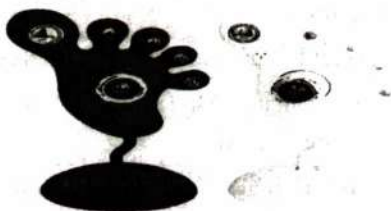
### 1.4.3. Các thiết bị ngoại vi thường sử dụng

- **Modem**



Hình 1.41. Modem wifi

- **Webcam**



Hình 1.42. Webcam

- **Loa**



Hình 1.43. Loa máy tính



- Máy in



*Hình 1.44. Máy in*

- Máy quét



*Hình 1.45. Máy quét ảnh*

- Micro



*Hình 1.46. Micro để bàn không dây*

## 1.5. GIỚI THIỆU MỘT SỐ PHẦN MỀM THÔNG DỤNG DÙNG CHO MÁY TÍNH

### 1.5.1. Phần mềm trình duyệt Web

- Firefox.
- Internet Explorer.
- Google Chrome.
- Opera.

### 1.5.2. Phần mềm văn phòng thông dụng

- Office.
- Phần mềm kế toán (EMASS, SAS INOVA 8.0 HRM, Fast Financial, AccountingOnline.vn – Chương trình kế toán trực tuyến...).

### 1.5.3. Phần mềm xử lý đồ hoạ

- Adobe Photoshop của hãng Adobe sản xuất, là phần mềm đồ hoạ chuyên xử lý ảnh dành cho giới chuyên nghiệp (thợ ảnh, kiến trúc sư, quảng cáo...). Được dùng để thiết kế ảnh, phục hồi ảnh cũ, vẽ kiểu...
- 3D Studio Max là phần mềm đồ hoạ vi tính 3 chiều của Công ty Autodesk Media & Entertainment, hoạt động trên hệ điều hành Window Win 32 hoặc Win 64.

### 1.5.4. Phần mềm lập trình

• **Một số phần mềm lập trình C thông dụng** (Turbo C, Bloodshed Dev-C++, C-Free):

– Bloodshed Dev-C++: Chương trình hỗ trợ đầy đủ các chức năng và một IDE chuyên nghiệp, từ việc soạn thảo trợ giúp đến Debug và hỗ trợ thư viện cho các lập trình viên C/C++ (kể cả viết chương trình cho DOS cũng như viết chương trình cho Windows).

– C-Free: Chương trình ứng dụng được thiết kế dành riêng cho ngôn ngữ lập trình C. Việc viết code trở nên dễ dàng (chỉ cần gõ từ đầu tiên của từ khoá chương trình sẽ tự động điền các từ khoá thích hợp, tương tự như vậy chương trình còn tự động gọi tên hàm).

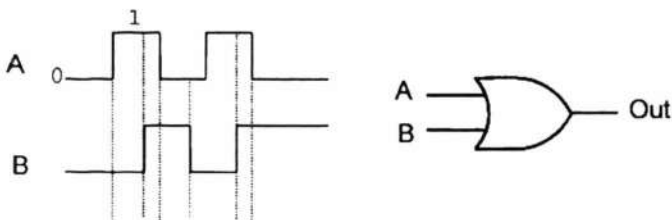
• **Microsoft Visual Studio:** Bộ công cụ lập trình toàn diện hỗ trợ đa ngôn ngữ, cho phép xây dựng và tích hợp các ứng dụng và dịch vụ Web, tăng hiệu suất làm việc. Được thiết kế tích hợp chặt chẽ với các chuẩn và giao thức Internet như XML và SOAP, Visual studio. Net đã đơn giản hoá đáng kể quy trình phát triển ứng dụng.

• **Eclipse:** Phần mềm miễn phí, được các nhà phát triển sử dụng để xây dựng những ứng dụng J2EE. Sử dụng Eclipse nhà phát triển có thể tích hợp

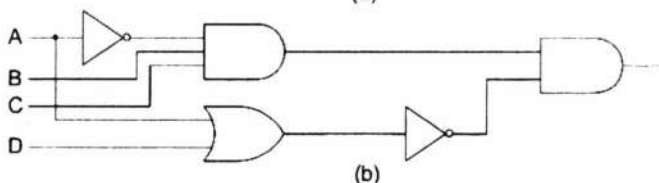
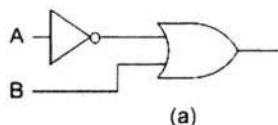
với nhiều công cụ hỗ trợ khác để có một bộ công cụ hoàn chỉnh mà không cần dùng đến phần mềm riêng nào khác. Eclipse SDK gồm 3 phần chính là: Platform, Java Development Toolkit (JDT), Plug-in Development Environment (PDE). Với JDT, Eclipse được xem như là một môi trường hỗ trợ phát triển Java mạnh mẽ. PDE hỗ trợ việc mở rộng Eclipse, tích hợp các Plug-in vào Eclipse Platform. Eclipse Platform là nền tảng của toàn bộ phần mềm Eclipse, mục đích của nó là cung cấp những dịch vụ cần thiết cho việc tích hợp những bộ công cụ phát triển phần mềm khác dưới dạng Plug-in, bản thân JDT cũng có thể được coi như là một Plug-in làm cho Eclipse như là một Java IDE (Integrated Development Environment).

## CÂU HỎI VÀ BÀI TẬP

1. Dựa vào tiêu chuẩn nào người ta chia máy tính thành các thế hệ?
2. Đặc trưng cơ bản của các thế hệ máy tính thứ 1, thứ 2, thứ 3 và thứ 4.
3. Khuynh hướng phát triển của hệ thống máy tính ngày nay là gì?
4. Việc phân loại máy tính dựa trên tiêu chuẩn nào?
5. Lượng thông tin là gì?
6. Trong máy tính, các số âm được lưu trữ và xử lý như thế nào?
7. Điểm chung nhất trong các cách biểu diễn một số nguyên n bit có dấu là gì?
8. Nêu cấu trúc của một hệ thống máy tính đơn giản.
9. Kể tên các phần mềm cài đặt cho máy tính mà em biết.
10. Xác định dạng sóng đầu ra cổng OR khi đầu vào A, B thay đổi theo giản đồ sau:



11. Xác định hàm đầu ra của các mạch sau:



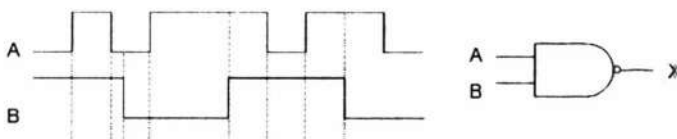
12. Vẽ sơ đồ thực hiện các biểu thức sau:

a)  $x = AB + BC$ ;

b)  $x = ABC(A + D)$ ;

c)  $y = AC + BC + ABC$ .

13. Xác định dạng sóng đầu ra của cổng NAND ứng với đầu vào như sau:



14. Đổi các số hệ thập phân sau sang hệ nhị phân:

a) 128

b) 4096

c) 256

d) 65536

e) 254

f) 9

g) 1024

h) 15

i) 344

j) 998

15. Đổi các số hệ nhị phân sau sang hệ thập phân, hệ thập lục phân:

a) 1001 1001

b) 1001 1101

c) 1100 0011

d) 0000 1001

e) 1111 1111

f) 0000 1111

g) 0111 1111

h) 1010 0101

i) 0100 0101

j) 0101 1010

k) 1111 0000

l) 1011 1101

m) 1100 0010

n) 0111 1110

o) 1110 1111.

16. Đổi các số hệ thập lục phân sau sang hệ nhị phân:

a) 0ABCD

b) 1024

c) 0DEAD

d) 0ADE

e) 0BEEF

f) 8

g) 05AAF

h) 0FFFF

i) 0ACDB

j) 0CDBA.

17. Biểu diễn số thực  $(37.76)_{10}$  dưới dạng số có dấu phẩy động chính xác đến 32 bit.
18. Thực hiện các phép toán AND các cặp giá trị ở hệ thập lục phân sau:
- |                |                 |               |
|----------------|-----------------|---------------|
| a) 0FF00, 0FF0 | b) 0F00F, 1234  | c) 4321, 1234 |
| d) 2341, 3241  | e) 0FFFF, 0EDCB | f) 1111, 5789 |
| g) 0FABA, 4322 | h) 5523, 0F572  | i) 2355, 7466 |
- (Gợi ý: Chuyển các giá trị đó sang hệ nhị phân, sau đó thực hiện tính toán, rồi lại đổi lại giá trị ở dạng thập lục phân).
19. Cho hai số  $A = 0.010 * 2^3$ ;  $B = 0.11 * 2^2$ .
- Hãy thực hiện các phép tính sau:
- |              |            |            |
|--------------|------------|------------|
| a) $A + B$   | b) $A - B$ | c) $A * B$ |
| d) $A / B$ . |            |            |
20. Thực hiện các phép toán cộng, trừ các số BCD sau dưới dạng nhị phân:
- |                |              |              |
|----------------|--------------|--------------|
| a) $24 + 19$   | b) $15 - 19$ | c) $32 + 46$ |
| d) $57 - 24$ . |              |              |



## Chương 2

# KIẾN TRÚC CPU CỦA MÁY TÍNH

### MỤC TIÊU

*Sau khi học xong chương này, người học:*

- ✓ Hiểu được sự tiến bộ của các bộ vi xử lý Intel; phân tích được cấu trúc của CPU cũng như biết được những thành phần cơ bản của một máy tính, tập các thanh ghi của vi xử lý.
- ✓ Nắm được các phương pháp quản lý bộ nhớ; phương pháp định địa chỉ và mã hoá lệnh.
- ✓ Nắm được cách phân loại lệnh của vi xử lý 8086; biết cách sử dụng lệnh và ứng dụng của các lệnh vào giải quyết các yêu cầu bài toán đặt ra.
- ✓ Sử dụng Card test main để sửa chữa hỏng hóc của Mainboard.

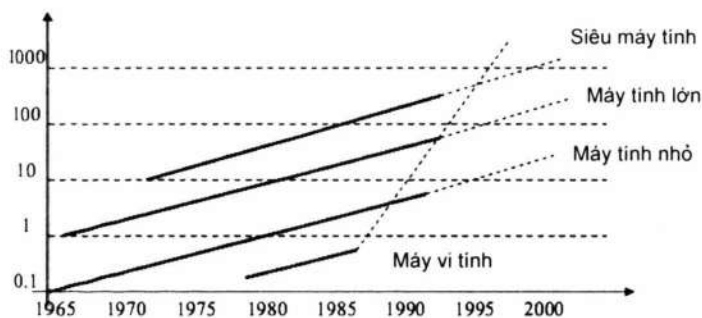
### NỘI DUNG

- 2.1. Sự tiến hoá của các bộ vi xử lý Intel.
  - 2.2. Cấu trúc CPU.
  - 2.3. Tập các thanh ghi của vi xử lý.
  - 2.4. Các phương pháp quản lý bộ nhớ.
  - 2.5. Các phương pháp định địa chỉ.
  - 2.6. Mã hoá lệnh.
  - 2.7. Tập lệnh 8086.
- Câu hỏi và bài tập.

## 2.1. SỰ TIẾN HOÁ CỦA CÁC BỘ VI XỬ LÝ INTEL

Hình 2.1 cho thấy diễn biến hiệu năng tối đa của máy tính. Hiệu năng này tăng theo hàm số mũ, độ tăng trưởng của các máy tính là 35% mỗi năm;

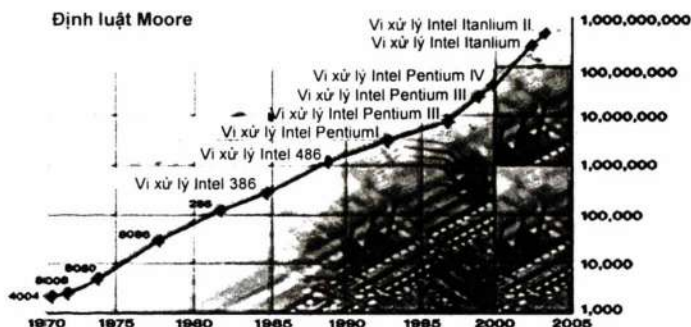
còn đối với các loại máy tính khác, độ tăng trưởng là 20% mỗi năm. Điều này cho thấy tính năng của các máy tính hiện nay đã vượt qua các loại máy tính khác vào đầu thập niên 90 của thế kỷ XX.



Hình 2.1. Đánh giá hiệu năng của máy tính

Máy tính dùng nhiều bộ xử lý song song rất thích hợp khi phải tính toán lớn. Sự tăng trưởng theo hàm số mũ của công nghệ chế tạo transistor MOS là nguồn gốc của hiệu năng các máy tính.

Hình 2.2 và 2.3 cho thấy sự tăng trưởng về tần số xung nhịp của các bộ vi xử lý MOS. Độ tăng trưởng của tần số xung nhịp bộ xử lý tăng gấp đôi sau mỗi thế hệ và độ trì hoãn trên mỗi cổng/xung nhịp giảm 25% cho mỗi năm. Sự phát triển của công nghệ máy tính, đặc biệt là sự phát triển của bộ vi xử lý của các máy vi tính làm cho các máy vi tính có tốc độ vượt qua tốc độ bộ xử lý của các máy tính lớn hơn.



Hình 2.2. Sự phát triển của bộ xử lý Intel dựa vào số lượng transistor trong một mạch tích hợp theo quy luật Moore

<i>Bộ xử lý Intel</i>	<i>Năm SX</i>	<i>Số lượng transistor tích hợp</i>
4004	1971	2,250
8008	1972	2.500
8080	1974	5,000
8086	1978	29,000
286	1982	120,000
Intel386™ processor	1985	275,000
Intel486™ processor	1989	1,180,000
Intel® Pentium® processor	1993	3,100,000
Intel® Pentium® II processor	1997	7,500,000
Intel® Pentium® III processor	1999	24,000,000
Intel® Pentium® 4 processor	2000	42,000,000
Intel® Itanium® processor	2002	220,000,000
Intel® Itanium® 2 processor	2003	410,000,000

*Hình 2.3. Các thông số về sự phát triển của bộ xử lý Intel dựa vào số lượng transistor trong một mạch tích hợp theo quy luật Moore*

Từ năm 1965, G. Moore (đồng sáng lập công ty Intel) quan sát và nhận thấy số transistor trong mỗi mạch tích hợp có thể tăng gấp đôi sau mỗi năm. G. Moore đưa ra dự đoán: Khả năng của máy tính sẽ tăng lên gấp đôi sau 18 tháng với giá thành là như nhau.

Kết quả của quy luật Moore là:

- Chi phí cho máy tính giảm.
- Giảm kích thước các linh kiện, máy tính sẽ giảm kích thước.
- Hệ thống kết nối bên trong mạch ngắn: tăng độ tin cậy, tăng tốc độ.
- Tiết kiệm năng lượng cung cấp, toả nhiệt thấp.
- Các IC thay thế cho các linh kiện rời.

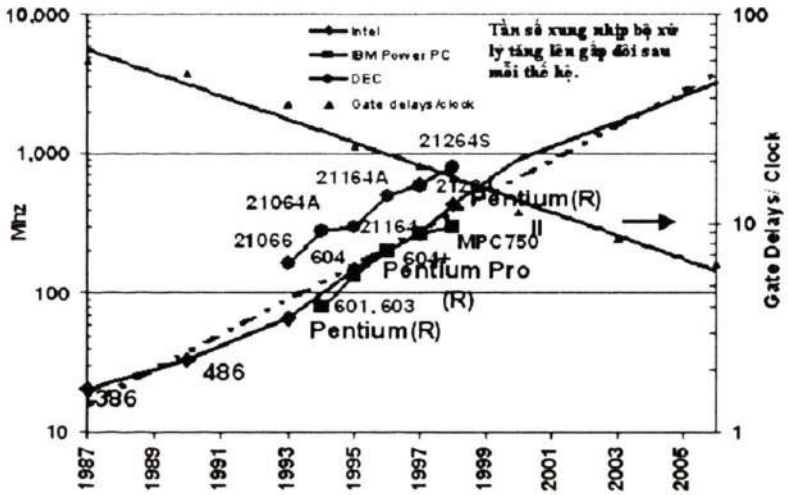
Một số khái niệm liên quan:

– *Mật độ tích hợp* là số linh kiện tích hợp trên một diện tích bề mặt tấm silicon cho sẵn, cho biết nhiệm vụ mà mạch có thực hiện.

– *Tần số xung nhịp bộ vi xử lý* cho biết tần số thực hiện nhiệm vụ.

– *Tốc độ xử lý của máy tính trong một giây* (hay công suất tính toán của mỗi mạch):

- + Được tính bằng tích của mật độ tích hợp và tần số xung nhịp;
- + Công suất này cũng tăng theo hàm mũ đối với thời gian.



Hình 2.4. Xung nhịp các bộ vi xử lý CMOS

## 2.2. CẤU TRÚC CPU

CPU là bộ não của máy tính, nó điều khiển mọi hoạt động của máy tính. CPU liên hệ với các thiết bị khác qua Mainboard và hệ thống cáp của thiết bị. CPU giao tiếp trực tiếp với bộ nhớ RAM và ROM, còn các thiết bị khác được liên hệ thông qua một vùng nhớ (địa chỉ vào/ra) và một ngắt thường gọi chung là cổng.

Khi một thiết bị cần giao tiếp với CPU nó sẽ gửi yêu cầu ngắt (*Interrupt Request – IRQ*) và CPU sẽ gọi chương trình xử lý ngắt tương ứng và giao tiếp với thiết bị thông qua vùng địa chỉ được quy định trước. Chính điều này dẫn đến khi khai báo hai thiết bị có cùng địa chỉ vào/ra và cùng ngắt giao tiếp sẽ lỗi hệ thống (xung đột ngắt – *IRQ Conflict*) có thể làm treo máy.

Ngày nay, với các thế hệ CPU mới có khả năng làm việc với tốc độ cao và BUS dữ liệu rộng giúp cho việc xây dựng chương trình đa năng ngày càng dễ dàng hơn.

Để đánh giá các CPU, người ta thường căn cứ vào các thông số của CPU như tốc độ, độ rộng của BUS, độ lớn của cache và tập lệnh được CPU hỗ trợ. Tuy nhiên, rất khó để có thể đánh giá chính xác các thông số này, do



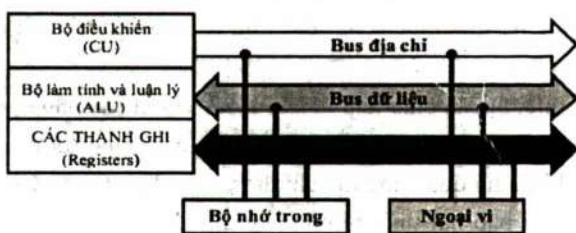
đó người ta vẫn thường dùng các chương trình thử trên cùng một hệ thống có các CPU khác nhau để đánh giá các CPU.

Thành phần cơ bản của một bộ máy tính gồm: bộ xử lý trung tâm (CPU: *Central Processing Unit*), bộ nhớ trong, các bộ phận xuất/nhập thông tin. Các bộ phận trên được kết nối với nhau thông qua các hệ thống bus.

Hệ thống bus bao gồm: bus địa chỉ, bus dữ liệu và bus điều khiển. Bus địa chỉ và bus dữ liệu dùng trong việc chuyển dữ liệu giữa các bộ phận trong máy tính. Bus điều khiển làm cho sự trao đổi thông tin giữa các bộ phận được đồng bộ.

Thông thường người ta phân biệt một bus hệ thống dùng trao đổi thông tin giữa CPU và bộ nhớ trong (thông qua cache), và một bus vào/ra dùng trao đổi thông tin giữa các bộ phận vào/ra và bộ nhớ trong.

Bộ xử lý trung tâm (CPU): Một chương trình sẽ được sao chép từ đĩa cứng vào bộ nhớ trong cùng với các thông tin cần thiết cho chương trình hoạt động, các thông tin này được nạp vào bộ nhớ trong từ các bộ phận cung cấp thông tin (ví dụ như bàn phím hay đĩa từ). Bộ xử lý trung tâm sẽ đọc các lệnh và dữ liệu từ bộ nhớ, thực hiện các lệnh và lưu các kết quả trở lại bộ nhớ trong hay cho xuất kết quả ra bộ phận xuất thông tin (màn hình hay máy in).



Hình 2.5. Cấu trúc một hệ máy tính đơn giản

## Thành phần cơ bản của một máy tính:

### a) Bộ nhớ trong

Đây là một tập hợp các ô nhớ, mỗi ô nhớ có một số bit nhất định và chứa một thông tin được mã hoá thành số nhị phân mà không quan tâm đến kiểu của dữ liệu mà nó đang chứa. Các thông tin này là các lệnh hay số liệu.



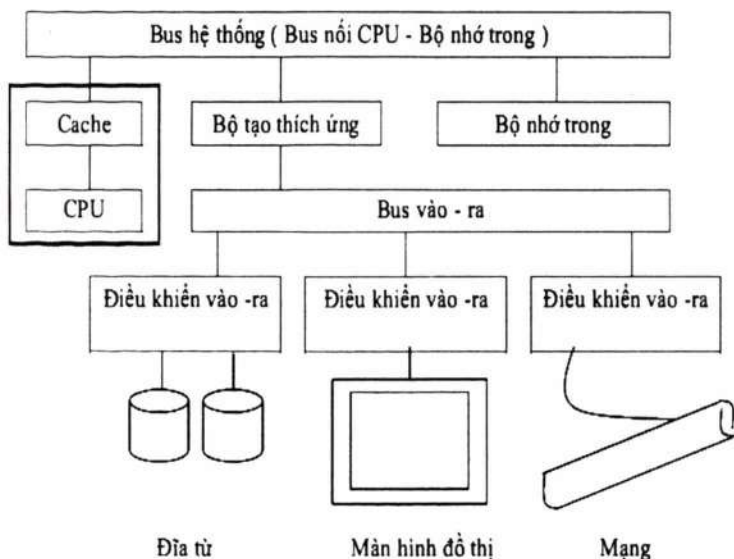
Mỗi ô nhớ của bộ nhớ trong đều có một địa chỉ. Thời gian thâm nhập vào một ô nhớ bất kỳ trong bộ nhớ là như nhau. Vì vậy, bộ nhớ trong còn được gọi là bộ nhớ truy cập ngẫu nhiên (RAM: *Random Access Memory*). Độ dài của một từ máy tính (*Computer Word*) là 32 bit (hay 4 byte), tuy nhiên dung lượng một ô nhớ thông thường là 8 bit (1 byte).

**b) Bộ xử lý trung tâm (CPU)**

Đây là bộ phận thi hành lệnh. CPU lấy lệnh từ bộ nhớ trong và lấy các số liệu mà lệnh đó xử lý. Bộ xử lý trung tâm gồm có hai phần: phần thi hành lệnh và phần điều khiển. Phần thi hành lệnh bao gồm bộ làm toán và luận lý (ALU: *Arithmetic And Logic Unit*) và các thanh ghi, có nhiệm vụ làm các phép toán trên số liệu. Phần điều khiển có nhiệm vụ đảm bảo thi hành các lệnh một cách tuần tự và tác động các mạch chức năng để thi hành các lệnh.

**c) Bộ phận vào/ra**

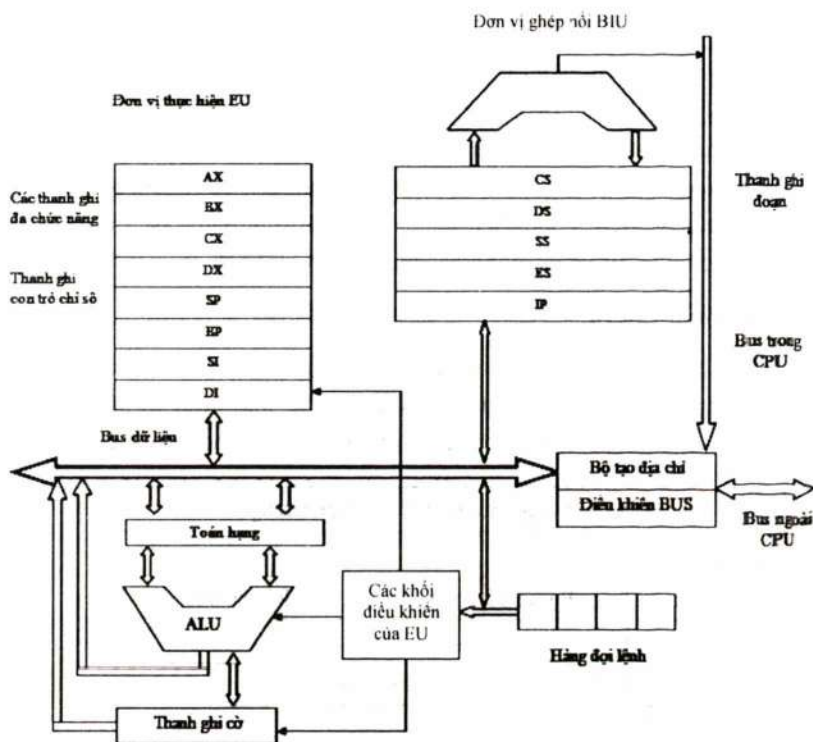
Đây là bộ phận xuất/nhập thông tin, bộ phận này thực hiện sự giao tiếp giữa máy tính với người sử dụng, hay giữa các máy tính trong hệ thống mạng (đối với các máy tính được kết nối thành một hệ thống mạng).



Hình 2.6. Sơ đồ mô tả hoạt động điển hình của một máy tính

Các bộ phận xuất/nhập thường gặp là: bộ lưu trữ ngoài, màn hình, máy in, bàn phím, chuột, máy quét, các giao diện mạng cục bộ hay mạng diện rộng... Bộ tạo thích ứng là một vi mạch tổng hợp (*chipset*) kết nối giữa các hệ thống bus có các tốc độ dữ liệu khác nhau.

Sơ đồ cấu trúc tổng quát của hệ vi xử lý 8086/8088 như hình 2.7.



Hình 2.7. Sơ đồ cấu trúc tổng quát của hệ vi xử lý 8086/8088

- BIU: Đơn vị giao tiếp BUS.
- EU: Đơn vị thực hiện lệnh (CU và ALU).

#### • Đơn vị giao tiếp BUS – BIU

Chức năng:

- Nhận trước các lệnh cất tạm vào hàng đợi lệnh.
- Đảm nhiệm chức năng điều khiển BUS.

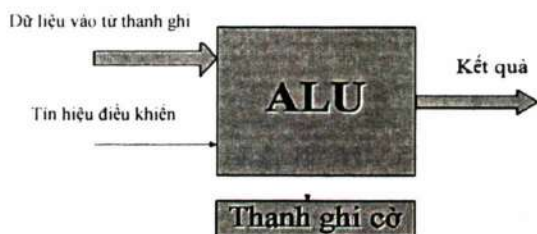
- Giao tiếp qua các BUS:
- + Phát các địa chỉ đến bộ nhớ và các cổng I/O thông qua A-BUS.
- + Đọc mã lệnh từ bộ nhớ thông qua D-Bus.
- + Đọc dữ liệu từ bộ nhớ thông qua D-Bus.
- + Ghi dữ liệu từ bộ nhớ thông qua D-Bus.
- + Đọc các dữ liệu từ các cổng Input thông qua D-Bus.
- + Ghi dữ liệu ra các cổng Output thông qua D-Bus.

### • Đơn vị số học và logic – ALU

*Chức năng:* Thực hiện các phép toán số học và logic.

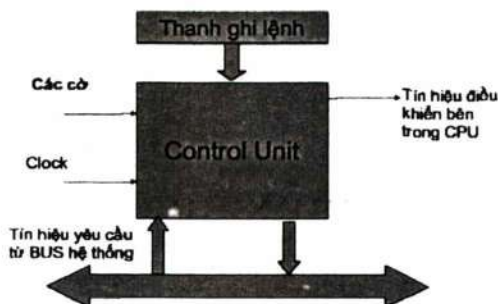
- Phép toán số học: cộng, trừ, nhân, chia, tăng, giảm, đảo...
- Phép toán logic: AND, OR, XOR, NOT, dịch bit...

### • Mô hình kết nối ALU



Hình 2.8. Mô hình kết nối ALU

### • Đơn vị điều khiển – CU



Hình 2.9. Mô hình kết nối đơn vị điều khiển

*Chức năng:*

- Nhận lệnh từ bộ nhớ đưa vào thanh ghi lệnh IP.
- Tăng nội dung thanh ghi PC mỗi khi nhận lệnh song.
- Giải mã lệnh và xác định thao tác mà lệnh yêu cầu.
- Phát tín hiệu điều khiển thực thi lệnh.
- Nhận các tín hiệu yêu cầu từ BUS hệ thống và giải quyết các yêu cầu đó.

- **Các thông tin đến CU**

- Clock: Tín hiệu xung nhịp từ mạch tạo dao động.
- Mã lệnh từ thanh ghi lệnh đưa đến CU giải mã.
- Các trạng thái cờ đưa đến cho biết trạng thái của CPU cũng như trạng thái thực hiện các phép toán trong ALU.
- Các tín hiệu điều khiển từ BUS điều khiển.
- Các tín hiệu điều khiển bên trong CPU: điều khiển thanh ghi, ALU.
- Các tín hiệu điều khiển bên ngoài CPU (bộ nhớ hay cổng vào/ra).

## 2.3. TẬP CÁC THANH GHI CỦA VI XỬ LÝ

Tập các thanh ghi của vi xử lý thực chất là vùng nhớ được các CPU nhận biết qua tên thanh ghi và có tốc độ truy xuất cực nhanh.

### 2.3.1. Chức năng

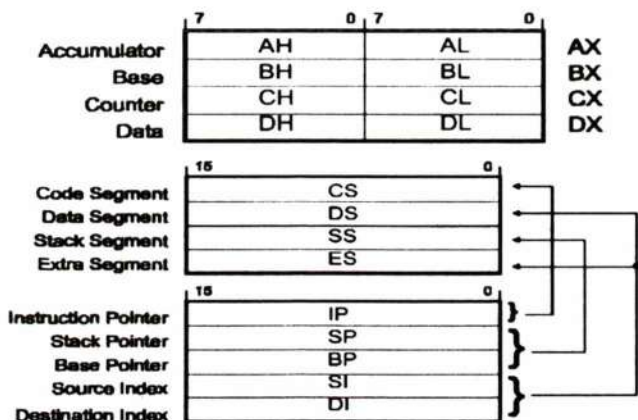
- Chứa thông tin tạm thời phục vụ cho hoạt động ở thời điểm hiện tại của CPU.
- Số lượng thanh ghi phụ thuộc vào bộ vi xử lý cụ thể: tăng hiệu năng CPU.
- Thanh ghi được chia là hai loại: loại lập trình được và loại không lập trình được.

### 2.3.2. Phân loại thanh ghi theo chức năng

- **Thanh ghi địa chỉ:** Thanh ghi được sử dụng để quản lý địa chỉ của cổng nhớ hay cổng vào/ra.

- **Thanh ghi dữ liệu:** Thanh ghi dùng để lưu trữ dữ liệu tạm thời (AX, BX, CX, DX).
- **Thanh ghi đa năng:** Thanh ghi có thể chứa dữ liệu hoặc địa chỉ đều được.
- **Thanh ghi điều khiển/trạng thái:** Thanh ghi chứa thông tin về trạng thái CPU.
- **Thanh ghi lệnh:** Thanh ghi chứa lệnh đang được thực hiện.

### 2.3.3. Cấu trúc của thanh ghi 8086/8088



Hình 2.10. Cấu trúc của thanh ghi 8086/8088

### 2.3.4. Các thanh ghi đa năng

Bảng 2.1. Các thanh ghi đa năng

AH	AL
BH	BL
CH	CL
DH	DL

(8 bit cao)

(8 bit thấp)



• **Thanh ghi AX (Accumulator – Acc):** Thanh ghi tích lũy, là thanh ghi dùng để thực hiện các phép toán số học, các lệnh xuất/nhập công. Thanh ghi này thường được dùng để lưu (tích lũy) các kết quả tính toán (cộng, trừ, nhân, chia...). Nếu kết quả là 8 bit thì thanh ghi AL được coi là thanh ghi Acc.

• **Thanh ghi BX (Base):** Thanh ghi cơ sở, thường dùng để chỉ địa chỉ cơ sở của một vùng nhớ.

• **Thanh ghi CX (Count):** Thanh ghi đếm, dùng để chứa số lần lặp của vòng lặp, phép dịch, phép quay.

• **Thanh ghi DX (Data):** Thanh ghi dữ liệu, thường dùng để lưu kết quả của các phép tính nhân, chia và định địa chỉ công trong các lệnh xuất/nhập công.

#### a) Thanh ghi con trỏ và chỉ số

Instruction Pointer	IP
Stack Pointer	SP
Base Pointer	BP
Source Index	SI
Destination Index	DI

Hình 2.11. Các thanh ghi con trỏ

Thanh ghi này chứa các giá trị offset (độ lệch) cho các phần tử định địa chỉ trong đoạn (Segmen).

• **SP:** Thanh ghi con trỏ ngăn xếp, chứa địa chỉ đỉnh ngăn xếp, cho phép truy xuất đến các địa chỉ đỉnh ngăn xếp.

• **SS:** Thanh ghi đoạn ngăn xếp, chứa địa chỉ hiện tại của lệnh.

• **BP:** Thanh ghi con trỏ cơ sở, mô tả offset tính từ SS nhưng còn được sử dụng để truy nhập dữ liệu từ bên trong ngăn xếp.

• **SI, DI:** Lưu trữ địa chỉ offset đối với lệnh truy cập dữ liệu cất trong đoạn dữ liệu DS.

• **Bộ đếm chương trình PC (Program counter)** còn gọi là con trỏ lệnh IP (*Intruction pointer*): Lưu giữ địa chỉ lệnh tiếp theo sẽ được nhận vào. Sau khi một lệnh được nhận vào, nội dung PC tự động tăng để trỏ sang lệnh kế tiếp.

#### b) Thanh ghi đoạn

Bộ nhớ CPU được chia thành các đoạn logic 64 Kbyte: Địa chỉ các đoạn được chứa trong các thanh ghi đoạn.

- **CS** (*Code segment*): Thanh ghi đoạn mã, nhận diện địa chỉ bắt đầu của đoạn chương trình hiện hành trong bộ nhớ.
- **DS** (*Data segment*): Nhận địa chỉ bắt đầu của đoạn số liệu hiện hành trong bộ nhớ (nơi chứa các biến chương trình).
- **SS** (*Stack segment*): Thanh ghi ngăn xếp, chứa địa chỉ bắt đầu của đoạn ngăn xếp trong bộ nhớ.
- **ES**: Thanh ghi đoạn mở rộng, lưu trữ các dữ liệu chuỗi.

**c) Thanh ghi cờ Flag Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cờ					OF	DF	IF	TF	SF	ZF		AF		PF		CF

*Ý nghĩa các cờ:* Các cờ OF, DF, IF, TF, SF, ZF phản ánh trạng thái của kết quả sau một thao tác nào đó. Các cờ AF, PF, CF là cờ điều khiển được lập hoặc xoá bằng các lệnh riêng.

**Bảng 2.2.** Thanh ghi và cờ Flag Register

TT	Cờ	Các cờ	Ý nghĩa	Nhóm
1	AF	Auxiliary Flag	Cờ phụ = 1 nếu số nhớ vượt 4 LSB	6 bit cờ trạng thái
2	CF	Carry Flag	Cờ nhớ = 1 nếu số nhớ vượt quá bit MSB	
3	OF	Overflow Flag	Cờ tràn = 1 nếu kết quả vượt thang	
4	SF	Sign Flag	Cờ dấu = 1 nếu kết quả âm (bảng giá trị bit cuối)	
5	PF	Parity Flag	Cờ chẵn = 1 nếu các số 1 kết quả chẵn (byte đầu)	
6	ZF	Zero Flag	Cờ zero = 1 nếu kết quả bằng 0	
7	DF	Direction Flag	Cờ hướng: Nếu DF = 1 thực hiện theo chiều giảm	3 bit cờ điều khiển
8	IF	Interrupt – Enable Flag	Cờ ngắt: Báo MP biết yêu cầu ngắt	
9	TF	Trap Flag	Cờ bẫy: Đặt MP vào chế độ chạy từng lệnh	

## 2.4. CÁC PHƯƠNG PHÁP QUẢN LÝ BỘ NHỚ

• **Thanh ghi 8086:** 20 bit địa chỉ, quản lý 1 Mbyte bộ nhớ. Độ lớn của các thanh ghi 16 bit, dẫn đến không đủ để quản lý bộ nhớ.

– *Giải pháp:* Chia bộ nhớ thành các đoạn 64 Kbyte.

– *Phương pháp quản lý bộ nhớ:* Sử dụng nhiều thanh ghi 16 bit để biểu diễn cho địa chỉ 20 bit.

• **Địa chỉ vật lý và địa chỉ luận lý (logic):**

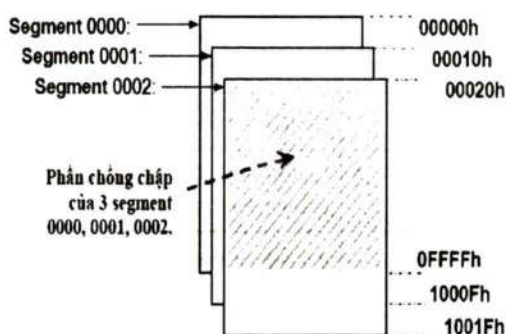
– Địa chỉ vật lý: Địa chỉ 20 bit, được sử dụng cho việc thiết kế các mạch giải mã địa chỉ cho bộ nhớ và I/O.

– Địa chỉ luận lý: Địa chỉ gồm có hai thành phần: địa chỉ đoạn (*segment*) và địa chỉ trong đoạn (*offset – độ lệch*).

Mỗi địa chỉ thành phần chỉ có 16 bit, cách tính địa chỉ vật lý từ địa chỉ logic như sau:

$$\text{Địa chỉ vật lý} = (\text{segment} \times 16) + \text{offset.}$$

Ví dụ: Địa chỉ vật lý ứng với địa chỉ luận lý B001:1234.

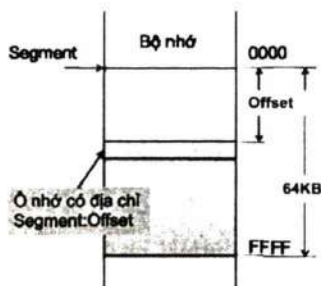


Hình 2.12. Quản lý bộ nhớ

+ Địa chỉ segment còn được gọi là địa chỉ nền của đoạn, điểm bắt đầu trong bộ nhớ.

+ Địa chỉ offset: Khoảng cách kể từ đầu đoạn của ô nhớ cần tham khảo, chiều dài tối đa của một đoạn là 64 Kbyte.

Trong mỗi đoạn, ô nhớ đầu tiên có offset là 0000h và có ô nhớ cuối cùng có offset là FFFF (h).



Hình 2.13. Quan hệ giữa địa chỉ vật lý và địa chỉ luận lý

• Địa chỉ luận lý và các thanh ghi:

- Địa chỉ luận lý được sử dụng trực tiếp hoặc thông qua các thanh ghi của nó.
- Các thanh ghi đoạn dùng để chứa địa chỉ đoạn segment.
- Các thanh ghi đa năng dùng để chứa địa chỉ trong đoạn offset.
- Để tham khảo đến địa chỉ luận lý có segment trong thanh ghi DS và offset trong thanh ghi BX, ta viết DS:BX.

Bảng 2.3. Định địa chỉ luận lý và các thanh ghi

Tham chiếu bộ nhớ	Đoạn mặc nhiên	Đoạn khác	Offset
Nhận lệnh	CS	Không	IP
Làm việc với Stack	SS	Không	IP
Dữ liệu tổng quát	DS	CS, ES,SS	Địa chỉ hiệu dụng
Nguồn của chuỗi (String)	DS	CS, ES,SS	SI
Đích của chuỗi (String)	ES	Không	DI
BX dùng làm con trỏ	DS	CS, ES,SS	Địa chỉ hiệu dụng
BP dùng làm con trỏ	SS	CS, ES,SS	Địa chỉ hiệu dụng

- Một số cặp thanh ghi luôn phải dùng chung với nhau bắt buộc:
  - CS:IP → lấy lệnh (địa chỉ sắp thi hành).
  - SS:SP → địa chỉ đỉnh chồng.
  - SS:BP → thông số trong chồng (dùng cho chương trình con).
  - DS:SI → địa chỉ chuỗi nguồn (chỉ có ý nghĩa trong các lệnh xử lý chuỗi).
  - ES:DI → địa chỉ chuỗi đích (chỉ có ý nghĩa trong các lệnh xử lý chuỗi).

## 2.5. CÁC PHƯƠNG PHÁP ĐỊNH ĐỊA CHỈ

### 2.5.1. Định địa chỉ thanh ghi

Toán hạng được chứa trong một thanh ghi, thanh ghi có tên trong trường địa chỉ toán hạng.

*Ví dụ:* MOS DX: CX

### 2.5.2. Định địa chỉ tức thời

- Toán hạng là dữ liệu 8 hoặc 16 bit nằm ngay trong lệnh.
- Toán hạng tức thời được lưu trong đoạn mã ngay sau mã toán hạng.
- Toán hạng được lấy ra cùng với lệnh và ghi vào hàng đợi lệnh và ghi vào hàng nhận lệnh trước.

*Ví dụ:* MOV: AL, 12h

**Bảng 2.4.** Cách định địa chỉ

Cách định địa chỉ	Mã đối tượng	Ví dụ		
		Từ gọi nhớ	Đoạn truy xuất	Hoạt động
Tức thời	B80010	MOV AC, 1000h	Mã	AH←10h, AL←00h
Thanh ghi	89CA	MOV DX, CX	VXL	DX←CX
Trực tiếp	8A260010	MOV AH, [1000h]	Dữ liệu	AH←[1000h]



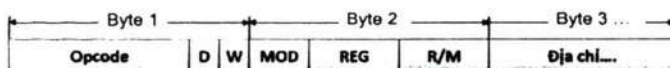
Cách định địa chỉ	Mã đối tượng	Vi dụ		
		Từ gọi nhớ	Đoạn truy xuất	Hoạt động
Gián tiếp thanh ghi	8B04	MOV AX, [SI]	Dữ liệu	AL←[SI], AH←[SI+1]
	FF25	JMP[DI]	Dữ liệu	IP←[DI+1:DI]
	FE 4600	INC BYTE PTR, [BP]	Stack	[BP]←[BP]+1
	FF0F	DEC WORD PTR, [BX]	Dữ liệu	[BX+1:BX]←[BX+1:BX]-1
Có chỉ số	8B4406	MOV AX, [SI+6]	Dữ liệu	AL←[SI+6], AH←[SI+7]
	FF6506	JMP [DI+6]	Dữ liệu	IP←[DI+6:DI+7]
Có nền	8B4602	MOV AX, [BP+2]	Stack	AL←[BP+2], AH←[BP+3]
	FF6702	JMP [BP+2]	Dữ liệu	IP←[BX+3:BX+6]
Có nền và có chỉ số	8B00	MOV AX, [BX+SI]	Dữ liệu	AL←[BX+SI], AH[BX+SI+1]

### 2.5.3. Định vị bộ nhớ

- Định vị trực tiếp: Toán hạng chứa địa chỉ trực tiếp nằm trong lệnh. Địa chỉ đoạn hiện tại chứa trong DS.
- Định vị gián tiếp thanh ghi: Địa chỉ hiệu dụng là nội dung các thanh ghi BX, BP, SI hoặc DI.
- Định vị cơ sở: Địa chỉ hiệu dụng là tổng của độ dịch chuyển và nội dung BX hoặc BP.
- Định vị chỉ số và cơ sở: Địa chỉ hiệu dụng là nội dung thanh ghi cơ sở, thanh ghi chỉ số và độ dịch chuyển.

### 2.6. MÃ HOÁ LỆNH

Mã hoá lệnh như hình 2.14.



Hình 2.14. Mã hoá lệnh

Trong đó:

- Opcode: Mã lệnh;
- D: Hướng truyền dữ liệu;
- D = 0: Dữ liệu đi từ thanh ghi quy định bởi 3 bit REG;
- D = 1: Dữ liệu đi tới thanh ghi quy định bởi 3 bit REG;
- W: Xác định truyền byte (W = 0) hay truyền word (W = 1);
- REG: Chọn thanh ghi sử dụng;
- MOD và R/M: Chọn chế độ địa chỉ cho toán hạng của lệnh.

Ví dụ: Lệnh MOV.

Mnemonic and Description	Instruction Code			
<b>DATA TRANSFER</b>				
<b>MOV = Move:</b>	76543210	76543210	76543210	76543210
Register/Memory to/from Register	100010dw	mod reg r/m		
Immediate to Register/Memory	1100011w	mod 000r/m	data	data if w = 1
Immediate to Register	1011w reg	data	data if w = 1	
Memory to Accumulator	1010000w	addr-low	addr-high	
Accumulator to Memory	1010001w	addr-low	addr-high	
Register/Memory to Segment Register	10001110	mod 0 reg r/m		
Segment Register to Register/Memory	10001100	mod 0 reg r/m		

Hình 2.15. Lệnh MOV

Bảng 2.5. REG xác định thanh ghi cho toán hạng thứ nhất

16 – Bit (W=1)		8 – Bit (W=0)		Segment	
000	AX	000	AL	00	ES
001	CX	001	CL	01	CS
010	DX	010	DL	10	SS
011	BX	011	BL	11	DS
100	SP	100	AH		
101	BP	101	CH		
110	SI	110	DH		
111	DI	111	BH		

Bảng 2.6. MOD và R/M cùng nhau xác định cho toán hạng thứ hai

R/M	MOD=00	MOD=01	MOD=10	MOD=11 W=1	MOD=11 W=0
000	[BX]+[SI]	[BX]+[SI]+ADD8	[BX]+[SI]+ADD16	AX	AL
001	[BX]+[DI]	[BX]+[DI]+ADD8	[BX]+[DI]+ADD16	CX	CL
010	[BP]+[SI]	[BP]+[SI]+ADD8	[BP]+[SI]+ADD16	DX	DL
011	[BP]+[DI]	[BP]+[DI]+ADD8	[BP]+[DI]+ADD16	BX	BL
100	[SI]	[SI]+ADD8	[SI]+ADD16	SP	AH
101	[DI]	[DI]+ADD8	[DI]+ADD16	BP	CH
110	ADD16	[BP]+ADD8	[BP]+ADD16	SI	DH
111	[BX]	[BX]+ADD8	[BX]+ADD16	DI	BH

## 2.7. TẬP LỆNH VI XỬ LÝ 8086

### 2.7.1. Nhóm lệnh di chuyển dữ liệu

#### a) Lệnh MOV

- Copy giá trị từ toán hạng nguồn sang toán hạng đích.
- Kích thước dữ liệu có thể là 8 hoặc 16 bit.

#### • Cú pháp

*MOV destination, source*

- Lệnh MOV không ảnh hưởng tới thanh ghi cờ.
- Không thể chuyển dữ liệu trực tiếp giữa các ô nhớ.
- Không thể chuyển dữ liệu tức thời vào thanh ghi đoạn mà phải dùng thanh ghi trung gian.
- Không thể chuyển trực tiếp giữa hai thanh ghi đoạn.

#### • Minh họa lệnh MOV

.DATA

Table DB 3,5,6,9,11,12,15

.CODE

; truy xuất phần tử của mảng

MOV AL,table; hay MOV AL,table[0]

MOV AL,table+1; hay MOV AL,table[1]

MOV AL,table+4; hay MOV AL,table[4]

MOV AX,DS:[100h]; chép nội dung 16 bit tại địa chỉ 100h trong đoạn DS vào thanh ghi AX.

### b) Lệnh PUSH, POP

#### • PUSH source

Cất một từ vào trong ngăn xếp bộ nhớ vào stack.

#### • PUSH

Giảm thanh ghi SP 2 đơn vị và chuyển nội dung 16 bit của toán hạng nguồn vào đỉnh stack. Đỉnh stack được xác định bởi cặp thanh ghi SS:SP, không tác động đến các cờ.

#### • POP destination:

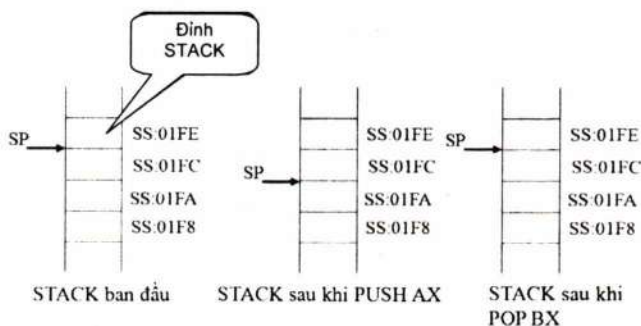
– Lấy dữ liệu 16 bit trong stack vào toán hạng đích.

– Toán hạng đích: Thanh ghi đa năng, thanh ghi đoạn (trừ thanh ghi đoạn mã CS) hoặc ô nhớ

$$SP \leftarrow SP + 2$$

– Không tác động đến các thanh ghi cờ.

#### • Mô tả lệnh PUSH, POP



Hình 2.16. Mô tả lệnh PUSH, POP

**PUSH AX; [SS:SP+1, SS:SP] ← AX**

**POP BX; BX ← [SS:SP+1, SS:SP]**

## 2.7.2. Nhóm lệnh chuyển địa chỉ

#### • LEA destination, source

– Nạp địa chỉ hiệu dụng vào thanh ghi.

– Đích thường là các thanh ghi: BX, CX, DX, BP, SI, DI.

– Nguồn là tên biến trong đoạn DS được chỉ rõ trong lệnh hoặc trong ô nhớ cụ thể.

– Mục đích tính địa chỉ lệch của biến hoặc địa chỉ của ô nhớ chọn làm gốc rồi nạp vào thanh ghi đã chọn.

– Lệnh không tác động đến cờ.

#### • **LDS**

Nạp một từ (từ bộ nhớ) vào thanh ghi trong lệnh và một từ tiếp theo vào DS.

#### • **LDS destination, source**

– Dịch là các thanh ghi AX.

– Gốc là ô nhớ trong đoạn DS được chỉ rõ trong lệnh.

– Dịch  $\leftarrow$  gốc, DS  $\leftarrow$  gốc + 2.

– Lệnh không tác động đến các cờ.

### 2.7.3. Nhóm lệnh chuyển thanh ghi cờ

#### • **LAHF**

– Nạp byte thấp của thanh ghi cờ vào AH.

– Kết hợp với lệnh PUSH AX để lưu thanh ghi cờ vào trong ngăn xếp.

#### • **SAHF**

– Cất thanh ghi AH vào byte thấp của thanh ghi cờ.

– Sau lệnh này, các bit cờ được cập nhật là: AF, CF, OF, PF, SF, ZF.

### 2.7.4. Nhóm lệnh chuyển dữ liệu qua cổng

#### • **IN Acc, port**

– Đọc dữ liệu từ cổng vào.

– Port là địa chỉ 8 bit của cổng: 00H-FFH, có thể đưa địa chỉ cổng vào thanh ghi DX (nếu địa chỉ cổng là 16 bit).

– Acc là thanh ghi tích lũy: Nếu Acc là AL thì dữ liệu 8 bit đưa vào cổng port; nếu Acc là AX thì dữ liệu đưa vào là 16 bit và được đưa vào từ cổng port và port + 1.

– Lệnh không tác động đến thanh ghi cờ.



- **OUT port, Acc**

- Đưa dữ liệu từ Acc ra cổng port.
- Nếu Acc là AL thì dữ liệu 8 bit được đưa ra cổng port; nếu Acc là AX thì dữ liệu 16 bit được đưa ra cổng port và port + 1.
- Có thể biểu diễn địa chỉ cổng bằng thanh ghi DX (DX chứa địa chỉ cổng): OUT DX, Acc.
- Lệnh không tác động đến thanh ghi cờ.

### 2.7.5. Lệnh so sánh

- **CMP đích, gốc**

- Toán hạng đích và gốc có thể là thanh ghi, bộ nhớ, hay hằng số, nhưng phải có cùng độ dài và không được phép là hai ô nhớ.
- Lệnh này ảnh hưởng đến giá trị các thanh ghi cờ, không lưu kết quả, thường dùng để tạo các cờ cho các lệnh nhảy có điều kiện.

**Bảng 2.7.** Ảnh hưởng của lệnh so sánh tới giá trị thanh ghi cờ

	CF	ZF
Đích = gốc	0	1
Đích > gốc	0	0
Đích < gốc	1	0

### 2.7.6. Nhóm lệnh chuyển điều khiển

- **JMP nhãn**

Lệnh nhảy không có điều kiện, lệnh mới được bắt đầu tại địa chỉ được mô tả trong lệnh.

**JMP Label**

**JMP near Label; nhảy gần (chiếm 3 byte)**

**JMP short Label; nhảy ngắn (chiếm 2 byte)**

**JMP far Label; nhảy xa (chiếm 5 byte)**

**JMP AX; IP←AX.**

### 2.7.7. Lệnh nhảy có điều kiện

• **J <điều\_kiện> short\_label**

– Lệnh nhảy có điều kiện, trước tiên kiểm tra điều kiện; nếu điều kiện thoả mãn thì nhảy đến nhãn short\_label; nếu không thì lệnh nhảy bị bỏ qua. Khoảng cách nhảy của lệnh nhảy có điều kiện từ -128 đến +128 (ngắn).

– Nhảy có điều kiện có dấu:

SYMBOL	Description	Condition for jump
JG/JNLE	Jump if greater than Jump if not less than or equal	ZF=0, SF=OF
JGE/JNL	Jump if greater than or equal Jump if not less than	SF=OFS
JL/JNGE	Jump if less than Jump if not greater or equal	SF<>OF
JLE/JNG	Jump if less than or equal Jump if not greater	ZF=1, SF<>OF

– Nhảy có điều kiện không dấu:

SYMBOL	Description	Condition for jump
JA/JNBE	Jump if above Jump if not below or equal	CF = 0, ZF = 0
JAE/JNB	Jump if above or equal Jump if not below	CF = 0
JB/JNAE	Jump if below Jump if not above or equal	CF = 1
JBE/JNA	Jump if below or equal Jump if not above	CF = 1, ZF = 1

– Nhảy một cờ:

SYMBOL	Description	Condition for jump
JE/JZ	Jump if equal Jump if zero	ZF = 1
JNE/JNZ	Jump if not equal Jump if not zero	ZF = 0
JNC	Jump if carry Jump if not carry	CF = 1 CF = 0
JO	Jump if overflow	OF = 1
JNO	Jump if not overflow	OF = 0
JS	Jump if signed Jump if negative	SF = 1
JNS	Jump if nonnegative sign	SF = 0
JP/JPE	Jump if parity even	PF = 1
JNP/JPO	Jump if parity odd	PF = 0

Chú ý về các lệnh nhảy:

*Ví dụ:*

```
AX = 7FFFH, BX = 8000H
CMP AX, BX
JA BELOW
```

Lệnh nhảy không thực hiện được (do 7FFFH < 8000H trong trường hợp không dấu).

```
MOV CX, AX
CMP BX, CX
JLE NEXT
MOV CX, BX
NEXT:
```

### 2.7.8. Nhóm lệnh lặp

- **LOOP short\_label**

Giảm nội dung CX một đơn vị và nhảy đến nhãn short\_label nếu thỏa mãn điều kiện  $CX \neq 0$ : Nếu  $CX = 0$  thì lệnh này bị bỏ qua.

- **Các lệnh LOOPE, LOOPP, LOOPZ...**

*Ví dụ:*

```
MOV CX, 16
LAP:
INC AL
LOOP LAP
```

*Ví dụ vòng lặp:* In ra màn hình 80 ký tự?

```
MOV CX, 80
MOV AH, 2; hàm xuất ký tự
MOV DL, '?'
TOP:
INT 21H
LOOP TOP
```

### 2.7.9. Nhóm lệnh tính toán số học

- Cộng không nhớ: **ADD đích, nguồn**

đích  $\leftarrow$  nguồn + đích

- Không thực hiện phép cộng đối với các thanh ghi đoạn.
- Toán hạng đích chỉ có thể là thanh ghi hoặc ô nhớ.

- Lệnh cộng có nhớ: **ADC đích, nguồn**

đích  $\leftarrow$  nguồn + đích + (CF)

- Lệnh tăng: **INC đích**

đích  $\leftarrow$  đích + 1

- Lệnh trừ không nhớ: **SUB đích, nguồn**

đích  $\leftarrow$  đích - nguồn

- Trừ có nhớ: **SBB đích, nguồn**

đích  $\leftarrow$  đích - nguồn - (CF)

- **Lệnh giảm: DEC đích**

đích  $\leftarrow$  đích - 1

- **Thay đổi các cờ AF, OF, PF, SF, ZF** (cờ CF không bị tác động).

- **NEG đích:** Đổi dấu toán hạng đích.

- **MUL nguồn:** Phép nhân không dấu

$AX \leftarrow (AL) * (\text{nguồn}_8 \text{ bit});$  kết quả 16 bit trong AX

$(DX:AX) \leftarrow (AX) * (\text{nguồn}_{16} \text{ bit});$  kết quả 32 bit

- **IMUL:** Nhân số có dấu.

- **DIV nguồn:** Chia số không dấu.

$(AL) \leftarrow (AX) / (\text{nguồn}_8 \text{ bit});$  thương số

$(AH) \leftarrow (AX) / (\text{nguồn}_8 \text{ bit});$  số dư

$(AX) \leftarrow (DX:AX) / (\text{nguồn}_{16} \text{ bit})$

$(DX) \leftarrow (DX:AX) / (\text{nguồn}_{16} \text{ bit})$

- **IDIV:** Lệnh chia có dấu.

*Ví dụ:*

- Giả sử hai biến A, B là hai biến. Thực hiện phép toán:

$$A = 12 \times A - 5 \times B.$$

CODE:

MOV AX, 12 ; AX=12

IMUL A; AX=Ax12

MOV A, AX ; A=12xA

MOV AX, 5 ; AX=5

IMUL B ; AX=Bx5

SUB A, AX ; A=12xA-5xB

- Tính N! Chứa N trên CX, giá trị trả về trên AX với giả thiết không có tràn.

MOV AX, 1

MOV CX, N

TOP:

MUL CX

LOOP TOP



### • Toán tử dùng trong Assembler

- Toán tử thực hiện tính toán các giá trị hằng khi dịch chương trình.
- Lệnh điều khiển sự tính toán các giá trị không xác định được khi dịch chương trình, chỉ khi thực hiện chương trình thì giá trị mới được xác định.

**Bảng 2.8.** Các toán tử dùng trong Assembler

Toán tử	Cú pháp	Công dụng
+	+Exp	Dấu dương
-	- Exp	Dấu âm
*	Exp1* Exp2	Nhân
/	Exp1/Exp2	Chia
Mod	Exp1 Mod Exp2	Phần dư
+	Exp1+ Exp 2	Cộng
	Exp1 - Exp 2	Trừ
Shl	Exp Shl n	Dịch trái n bit
Shr	Exp Shr n	Dịch phải n bit

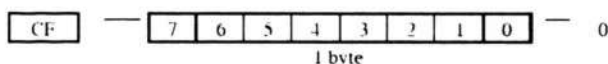
### 2.7.10. Nhóm lệnh dịch chuyển và quay vòng

- **SHL(SAL):** Dịch toán hạng đích sang trái

*Cú pháp:* SHL destination, CL

CL chứa số lần dịch.

- Sau mỗi lần dịch, một số 0 được thêm vào LSB.
- Các cờ bị ảnh hưởng:
  - + SF, PF, ZF phản ánh kết quả, AF không xác định.
  - + CF = bit cuối cùng được dịch ra.
  - + OF = 1 nếu kết quả đổi dấu ở lần dịch cuối cùng.

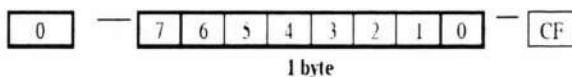


**Hình 2.17.** Mô tả lệnh dịch toán hạng sang trái

- **SHR(SAR):** Dịch toán hạng dịch sang phải.

*Cú pháp:* SHR destination, CL

- Sau mỗi lần dịch, một số 0 được thêm vào MSB.
- Các cờ ảnh hưởng giống SHL.
- SAR tương tự như SHR tuy nhiên giá trị MSB vẫn giữ nguyên giá trị nguyên thủy (Dấu không đổi).



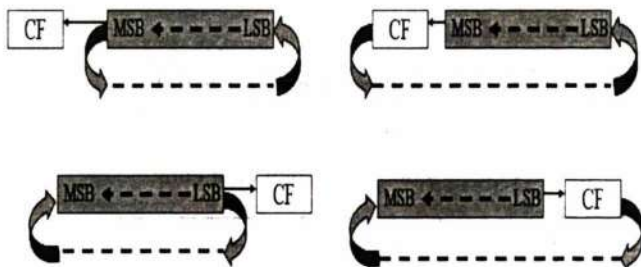
**Hình 2.18.** Mô tả lệnh dịch toán hạng sang phải

### • Lệnh quay

- ROL: Quay các bit sang trái. LSB được thay thế bằng MSB, CF = MSB.
- RCL: Quay trái qua cờ nhớ.
- ROR: Quay các bit sang phải. MSB được thay bằng LSB, CF = LSB.
- RCR: Quay phải qua cờ nhớ.

Cú pháp của lệnh:

- ROL destination, CL
- RCL destination, CL
- ROR destination, CL
- RCR destination, CL



**Hình 2.19.** Mô tả lệnh quay các lệnh quay

### 2.7.11. Nhóm lệnh thực hiện phép toán logic

- **AND destination, source**
- **OR destination, source**
- **XOR destination, source**

– Toán hạng đích là thanh ghi hoặc ô nhớ lưu kết quả; toán hạng nguồn có thể là hằng số, thanh ghi hoặc vị trí nhớ.

– Ảnh hưởng của các cờ SF, ZF, PF phản ánh kết quả. AF không xác định. CF = OF = 0.

- Sử dụng các mặt nạ để thay đổi các bit.

**Bảng 2.9.** Các phép toán logic

$a \text{ AND } 1 = a$	$a \text{ OR } 0 = a$	$a \text{ XOR } 0 = a$
$a \text{ AND } 0 = 0$	$a \text{ OR } 1 = 1$	$a \text{ XOR } 1 = \text{NOT } a$

**AND AL, 7FH; Xoá bit đầu AL**

**OR AL, 81H; Set 1 cho MSB và LSB của AL**

**XOR DX, 8000H; Thay đổi bit đầu của DX**

- Đổi một số ASCII thành một số.

+ Cách 1: SUB AL, 30H

+ Cách 2: AND AL, 0FH; xoá nửa byte cao của AL.

Đổi chữ thường thành chữ hoa.

+ Cách 1: SUB DL, 20H

+ Cách 2: AND DL, 0DFH; xoá bit thứ 5 của DL.

– Xoá một thanh ghi: MOV AX, 0 hoặc SUB AX, AX hoặc XOR AX, AX

– Lệnh NOT destination: Lấy bù đảo của toán hạng đích.

– Lệnh TEST destination, source: Thực hiện phép toán AND giữa toán hạng đích và toán hạng nguồn, nhưng không làm thay đổi toán hạng đích. Mục đích để set các cờ trạng thái:

ZF, SF, PF phản ánh kết quả, AF không xác định

CF=OF=0

– Lệnh TEST dùng để kiểm tra bit 1 trên toán hạng. Mặt nạ phải chứa bit 1 tại bit cần kiểm tra, các bit khác thì bằng 0: TEST destination,mask

*Ví dụ:* Kiểm tra một số là số chẵn hay lẻ?

**TEST AL, 1 ;AL chẵn?**

**JZ NEXT ; Đúng, nhảy đến nhãn NEXT**

## CÂU HỎI VÀ BÀI TẬP

1. Các thành phần của một hệ thống máy tính đơn giản là gì?
2. Nêu kết quả quy luật More về sự tiến hoá của bộ vi xử lý.
3. Nhiệm vụ của mỗi bus trong hệ thống bus của một hệ máy tính đơn giản. Tại sao trong thực tế cần có một hệ thống bus vào/ra?
4. Hãy nêu chức năng của đơn vị số học và logic – ALU.
5. Nêu chức năng của tập các thanh ghi của vi xử lý.
6. Người ta phân loại thanh ghi dựa trên yếu tố nào? Có bao nhiêu loại thanh ghi dựa vào cách phân loại ấy?
7. Có bao nhiêu phương pháp định địa chỉ?
8. Nhập một ký tự từ bàn phím và xuất ký tự vừa nhập ra màn hình tại hàng 11, cột 10.
9. Dùng hàm 02h của ngắt 21h để xuất chuỗi "Hello".
10. Di chuyển ký tự 'A' theo chiều ngang từ trái sang phải, bắt đầu tại hàng 1, cột 1; kết thúc tại hàng 1, cột 40.
11. Nhập một chuỗi từ bàn phím và xuất chuỗi ra màn hình bằng cách dùng hàm 02h của ngắt 21h.
12. Nhập một chuỗi từ bàn phím và xuất chuỗi ra màn hình bằng cách dùng hàm 09h của ngắt 21h.

(Lưu ý: Hàm 09h dùng để xuất chuỗi, trong đó chuỗi bắt buộc phải kết thúc bằng ký tự \$, nên phải thêm ký tự này vào cuối chuỗi trước khi xuất).

13. Viết chương trình điều khiển Port 0 sáng dần lên và tối dần đi.
14. Viết chương trình điều khiển 16 led sáng dần và tắt hết (dùng Port 0 và Port 1).
15. Viết chương trình điều khiển 8 led sáng hết và diêm sáng dịch chuyển tắt dần.



## **Chương 3**

# **BỘ NHỚ VÀ CÁC THIẾT BỊ NGOẠI VI**

### **MỤC TIÊU**

*Sau khi học xong chương này, người học:*

- ✓ Trình bày được kiến trúc của hệ thống nhớ máy vi tính. Hiểu về nguyên tắc hoạt động của các loại bộ nhớ.
- ✓ Trình bày được các phương pháp điều khiển vào/ra, các cổng vào/ra thông dụng, các kỹ thuật vào/ra.
- ✓ Hiểu được phương pháp ghép nối các thiết bị ngoại vi.

### **NỘI DUNG**

- 3.1.** Kiến trúc hệ thống nhớ máy vi tính.
  - 3.2.** Hệ thống vào/ra.
  - 3.3.** Ghép nối thiết bị ngoại vi.
  - 3.4.** Một số thiết bị ngoại vi.
- Câu hỏi và bài tập.

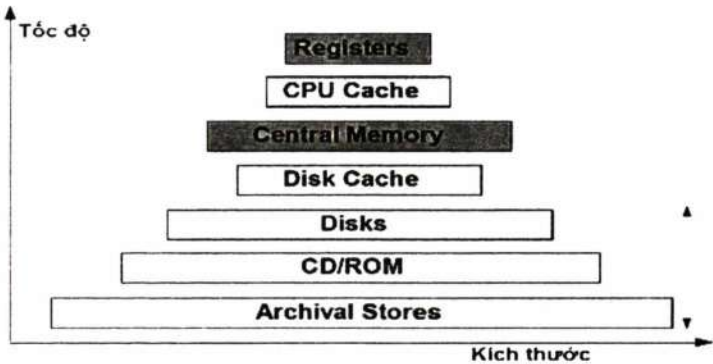
## **3.1. KIẾN TRÚC HỆ THỐNG NHỚ MÁY VI TÍNH**

### **3.1.1 Tổng quan hệ thống nhớ máy vi tính**

Bộ nhớ chứa chương trình nghĩa là chứa lệnh và dữ liệu. Người ta phân ra các loại bộ nhớ: Bộ nhớ trong RAM (Bộ nhớ vào/ra ngẫu nhiên) được chế tạo bằng chất bán dẫn, bộ nhớ chỉ đọc ROM cũng là loại bộ nhớ truy xuất ngẫu nhiên; và bộ nhớ ngoài bao gồm: đĩa cứng, đĩa mềm, băng từ, trống từ, các loại đĩa quang, các loại thẻ nhớ...

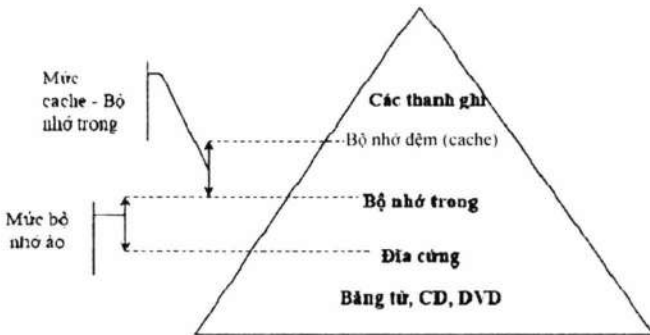
**Các cấp bộ nhớ:**

Các đặc tính như lượng thông tin lưu trữ, thời gian thâm nhập bộ nhớ, chu kỳ bộ nhớ, giá tiền mỗi bit nhớ khiến ta phải phân biệt các cấp bộ nhớ: các bộ nhớ nhanh với dung lượng ít đến các bộ nhớ chậm với dung lượng lớn.



Hình 3.1. Các cấp bộ nhớ

Các đặc tính chính của các cấp bộ nhớ dẫn đến hai mức chính là: mức cache – bộ nhớ trong và mức bộ nhớ ảo (bao gồm bộ nhớ trong và không gian cấp phát trên đĩa cứng). Cách tổ chức này trong suốt đối với người sử dụng. Người sử dụng chỉ thấy duy nhất một không gian định vị ô nhớ, độc lập với vị trí thực tế của các lệnh và dữ liệu cần thâm nhập.



Hình 3.2. Hai mức bộ nhớ

Các cấp bộ nhớ giúp ích cho người lập trình có một bộ nhớ thật nhanh với chi phí đầu tư giới hạn. Vì các bộ nhớ nhanh đắt tiền nên các bộ nhớ được tổ chức thành nhiều cấp, cấp có dung lượng ít thì nhanh, nhưng đắt tiền hơn cấp có dung lượng cao hơn. Mục tiêu của việc thiết lập các cấp bộ nhớ là người dùng có một hệ thống bộ nhớ rẻ tiền như cấp bộ nhớ thấp nhất và gần nhanh như cấp bộ nhớ cao nhất. Các cấp bộ nhớ thường được lồng vào nhau. Mọi dữ liệu trong một cấp thì được gấp lại trong cấp thấp hơn và có thể tiếp tục gấp lại trong cấp thấp nhất.

Có nhận xét rằng, mỗi cấp bộ nhớ có dung lượng lớn hơn cấp trên mình, ánh xạ một phần địa chỉ các ô nhớ của mình vào địa chỉ ô nhớ của cấp trên trực tiếp có tốc độ nhanh hơn, và các cấp bộ nhớ phải có cơ chế quản lý và kiểm tra các địa chỉ ánh xạ.

### 3.1.2. Bộ nhớ trong

**Bảng 3.1.** Các kiểu bộ nhớ trong

Kiểu bộ nhớ	Loại	Cơ chế xoá	Cơ chế ghi	Tình bay hơi
RAM	Đọc/ghi	Bằng điện, mức byte	Bằng điện	Có
ROM	Chỉ đọc	Không thể xoá	Mặt nạ	Không
Programmable ROM(PROM)			Bằng điện	
Erasable PROM Electrically Erasable				
Flash Memory	Hầu hết chỉ đọc	Tia cực tím, mức chip bằng điện, mức byte	Bằng điện	
		Bằng điện, mức khối		

#### 3.1.2.1. ROM (Read Only Memory)

Đây là bộ nhớ mà CPU chỉ có quyền đọc và thực hiện, chứ không có quyền thay đổi nội dung vùng nhớ. Loại này chỉ được ghi một lần với thiết bị ghi đặc biệt. Bộ nhớ chỉ đọc ROM cũng được chế tạo bằng công nghệ bán dẫn. Chương trình trong ROM được viết khi chế tạo nó.

- **ROM** thường được ghi các chương trình quan trọng như chương trình khởi động, chương trình kiểm tra thiết bị... Tiêu biểu trên Mainboard là ROM BIOS.

- **PROM** (*Programable ROM*): Chế tạo bằng các mối nối (cầu chì – có thể làm đứt bằng điện). Chương trình nằm trong PROM có thể được viết vào bởi người sử dụng bằng thiết bị đặc biệt và không thể xoá được.

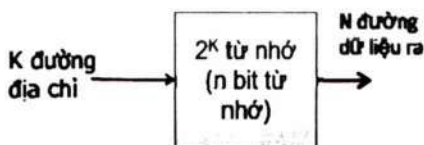
- **EPROM** (*Erasable Programable ROM*): Chế tạo bằng nguyên tắt phân cực tĩnh điện. Chương trình nằm trong ROM có thể được viết vào (bằng điện) và có thể xoá (bằng tia cực tím – trung hoà tĩnh điện) để viết lại bởi người sử dụng.

- **EEPROM** (*Electrically Erasable Programable ROM*): Chế tạo bằng công nghệ bán dẫn. Chương trình nằm trong ROM có thể được viết vào và có thể xoá (bằng điện) để viết lại bởi người sử dụng.

Hiện nay hầu hết các thiết bị đều có gắn ROM để phục vụ các chương trình cần thiết. ROM có nhiều loại với công nghệ khác nhau như: EPROM, FPROM...

- *Sử dụng để lưu các thông tin sau:*

- Thư viện các chương trình con.
- Các chương trình con điều khiển hệ thống (BIOS).
- Các bảng chức năng.

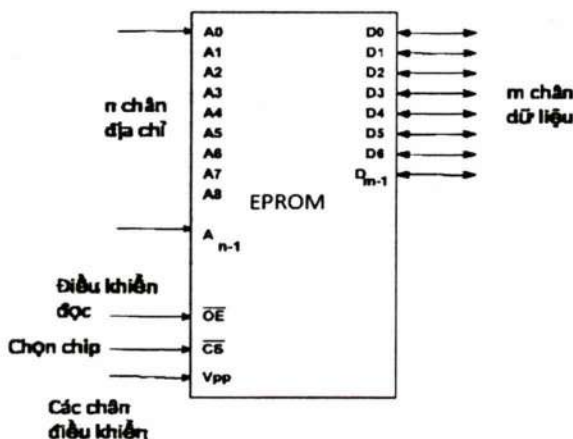


Hình 3.3. Mô hình bộ nhớ

- *Các kiểu ROM*

- ROM mật nạ: Thông tin được ghi khi sản xuất.
- PROM:
  - + Cần thiết bị chuyên dụng để ghi chương trình.
  - + Ghi được 1 lần.

- EPROM:
- + Giống PROM, nhưng ghi được nhiều lần.
- + Xoá bằng tia cực tím.
- EEPROM: Có thể ghi từ byte và xoá bằng điện.
- Plash Memory (*Bộ nhớ cực nhanh*): Ghi theo khối, xoá bằng điện.



Hình 3.4. Sơ đồ chân chip EPROM

### 3.1.2.2. RAM (Radom access Memmory)

Bộ nhớ RAM có đặc tính là các ô nhớ có thể đọc hoặc viết vào trong khoảng thời gian bằng nhau cho dù chúng ở bất kỳ vị trí nào trong bộ nhớ. Mỗi ô nhớ có một địa chỉ, thông thường mỗi ô nhớ là 1 byte (8 bit), nhưng hệ thống có thể đọc ra hay viết vào nhiều byte (2, 4 hay 8 byte). Bộ nhớ trong (RAM) được đặc trưng bằng dung lượng và tổ chức của nó (số ô nhớ và số bit cho mỗi ô nhớ), thời gian thâm nhập (thời gian từ lúc đưa ra địa chỉ ô nhớ đến lúc đọc được nội dung ô nhớ đó) và chu kỳ bộ nhớ (thời gian giữa hai lần liên tiếp thâm nhập bộ nhớ).

Nếu phân loại theo khe cắm trên Mainboard thì RAM có các loại sau:

- **SIMM** (*Single Inline Module Memory*): Là loại RAM giao tiếp 72 chân, được sử dụng nhiều ở các Mainboard cũ, dung lượng mỗi thanh có thể là: 4 MB, 8 MB, 16 MB, 32 MB...

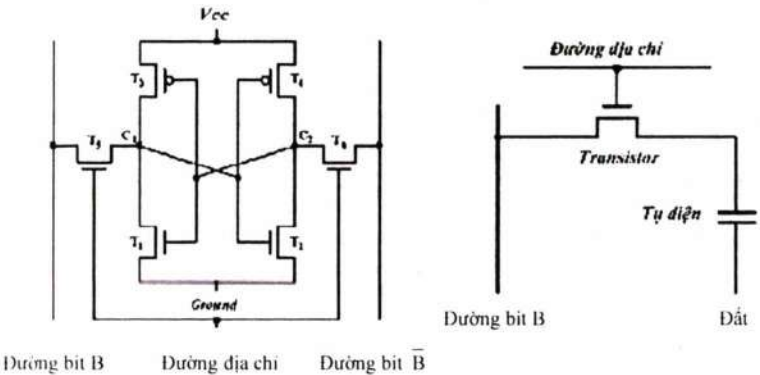


- **DIMM** (*Dual Inline Module Memory*): Là chuẩn thanh RAM 168 chân, có mặt ở các Mainboard mới, các thanh này có kích thước 8 MB trở lên và được cắm vào khe DIMM trên Mainboard.

- **SIPRAM** (*Single Inline Pin Random Access Memory*) và **DIPRAM** (*Dual Inline Pin Random Access Memory*): Là hai loại RAM thường được cắm sẵn trên Mainboard và thường có dung lượng nhỏ tính theo KB. Các Mainboard mới hiện không còn thấy loại này nữa.

Nếu phân loại theo công nghệ chế tạo, người ta phân biệt RAM tĩnh (*SRAM: Static RAM*) và RAM động (*Dynamic RAM*).

- **RAM tĩnh** được chế tạo theo công nghệ ECL (CMOS và BiCMOS). Mỗi bit nhớ gồm có các cổng logic với độ 6 transistor MOS, việc nhớ một dữ liệu là tồn tại nếu bộ nhớ được cung cấp điện. SRAM là bộ nhớ nhanh, việc đọc không làm huỷ nội dung của ô nhớ và thời gian thâm nhập bằng chu kỳ bộ nhớ.



Hình 3.5. SRAM và DRAM

- **RAM động** dùng kỹ thuật MOS. Mỗi bit nhớ gồm có một transistor và một tụ điện. Cũng như SRAM, việc nhớ một dữ liệu là tồn tại nếu bộ nhớ được cung cấp điện. Việc ghi nhớ dựa vào việc duy trì điện tích nạp vào tụ điện và như vậy việc đọc một bit nhớ làm nội dung bit này bị huỷ. Vậy, sau mỗi lần đọc một ô nhớ, bộ phận điều khiển bộ nhớ phải viết lại ô nhớ đó nội dung vừa đọc và do đó chu kỳ bộ nhớ động ít nhất là gấp đôi thời gian thâm nhập ô nhớ. Việc lưu giữ thông tin trong bit nhớ chỉ là tạm thời, vì tụ điện sẽ

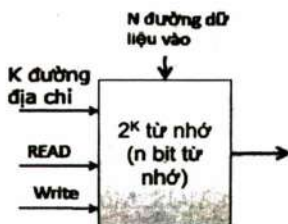


phóng hết điện tích đã nạp vào, và như vậy phải làm tươi bộ nhớ sau mỗi  $2\mu\text{s}$ . Làm tươi bộ nhớ là đọc ô nhớ và viết lại nội dung đó vào lại ô nhớ. Việc làm tươi được thực hiện với tất cả các ô nhớ trong bộ nhớ và được thực hiện tự động bởi một vi mạch bộ nhớ. Bộ nhớ DRAM chậm, nhưng rẻ tiền hơn SRAM.

- **SDRAM** (*Synchronous DRAM – DRAM đồng bộ*) – một dạng DRAM đồng bộ bus bộ nhớ. Tốc độ SDRAM đạt từ 66 – 133 MHz (thời gian thâm nhập bộ nhớ từ 75ns đến 150ns).

- **DDR SDRAM** (*Double Data Rate SDRAM*) là cải tiến của bộ nhớ SDRAM với tốc độ truyền tải gấp đôi SDRAM nhờ vào việc truyền tải hai lần trong một chu kỳ bộ nhớ. Tốc độ DDR SDRAM đạt từ 200 – 400 MHz.

- **RDRAM** (*Rambus RAM*) là một loại DRAM được thiết kế với kỹ thuật hoàn toàn mới so với kỹ thuật SDRAM. RDRAM hoạt động đồng bộ theo một hệ thống lập và truyền dữ liệu theo một hướng. Một kênh bộ nhớ RDRAM có thể hỗ trợ đến 32 chip DRAM. Mỗi chip được ghép nối tuần tự trên một module gọi là RIMM (*Rambus Inline Memory Module*), nhưng việc truyền dữ liệu giữa các mạch điều khiển và từng chip riêng biệt chứ không truyền giữa các chip với nhau. Bus bộ nhớ RDRAM là đường dẫn liên tục đi qua các chip và module trên bus, mỗi module có các chân vào và ra trên các đầu đối diện. Do đó, nếu các khe cắm không chứa RIMM sẽ phải gắn một module liên tục để đảm bảo đường truyền được nối liền. Tốc độ RDRAM đạt từ 400 – 800 MHz.



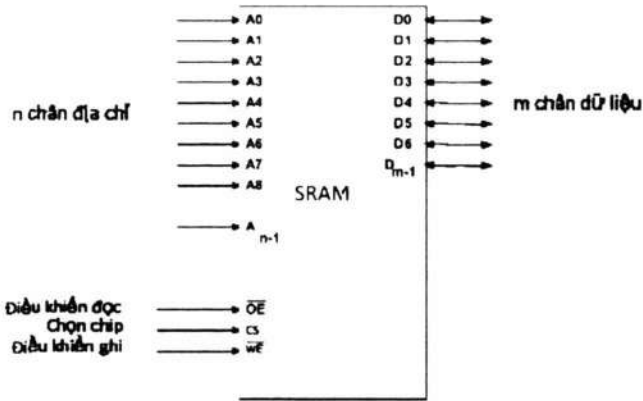
Hình 3.6. Mô hình bộ nhớ:

- **SRAM** (*Static RAM*)

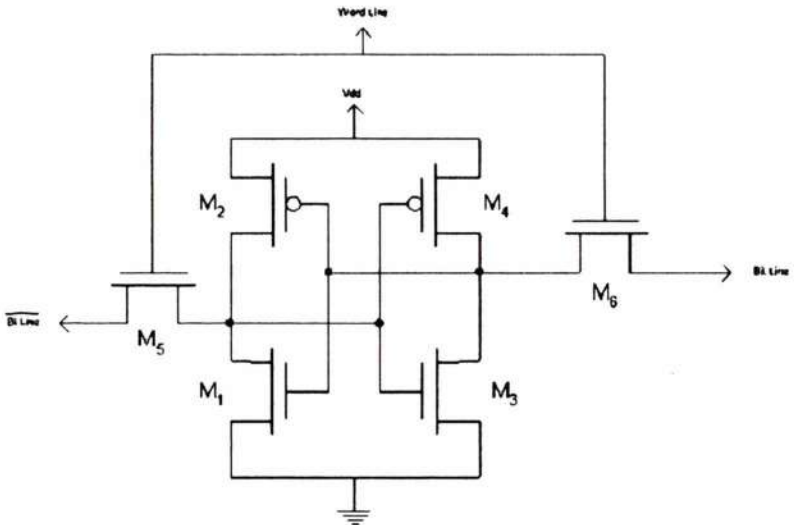
- Đặc điểm:

- + Các bit được lưu dựa trên các Flip – Flop (4 – 8 FF lưu 1 bit).

- + Thông tin lưu ổn định.
- + Cấu trúc phức tạp.
- + Dung lượng nhỏ.
- + Tốc độ nhanh (6 – 8ns).
- + Dùng làm cache.
- + Giá thành cao.

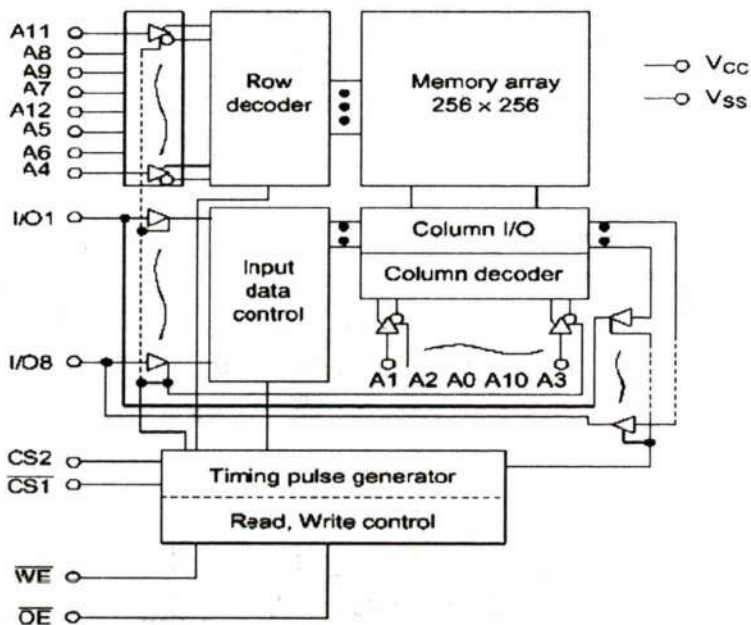


Hình 3.7. Sơ đồ chân Chip SRAM



Hình 3.8. Cấu trúc một vùng nhớ SRAM

- SRAM 6264:
- + Dung lượng  $8K \times 8$ .
- + 8 chân dữ liệu.
- + 13 chân địa chỉ.
- + Hai chân chọn chip.
- + Chân điều khiển đọc.
- + Chân điều khiển ghi.



Hình 3.9. Sơ đồ khối SRAM 6264

#### • DRAM (Dynamic RAM)

- Các bit được ghi dựa trên các tụ điện → Thường xuyên làm tươi.
- Dung lượng lớn.
- Tốc độ chậm (60 – 80ns).
- Dùng làm bộ nhớ chính.
- Giá thành phải chăng.

Các DRAM tiên tiến:

- SDRAM (*Synchronous Dynamic RAM*).
- DDRAM (*Double Data RAM*).
- RAM BUS RDRAM.

**Bảng 3.2.** Bảng các loại RAM

Loại RAM	Bảng thông cực đại lý thuyết
SDRAM 100 MHz	100 MHz x 64 bit = 800 Mbit/s
SDRAM 133 MHz	133 MHz x 64 bit = 1064 Mbit/s
DDRAM 2000 MHz; P PC 1600	2 x 100 MHz x 64 bit = 1600 Mbit/s
DDRAM 266 MHz; PC 2100	2 x 133 MHz x 64 bit = 2128 Mbit/s
DDRAM 266 MHz; PC 2600	2 x 166 MHz x 64 bit = 2656 MB/se
RDRAM 600 MHz	600 MHz x 16 bit = 1200 Mbit/s
RDRAM 700 MHz	700 MHz x 16 bit = 1400 Mbit/s
RDARM 800 MHz	800 MHz x 16 bit = 1600 Mbit/s

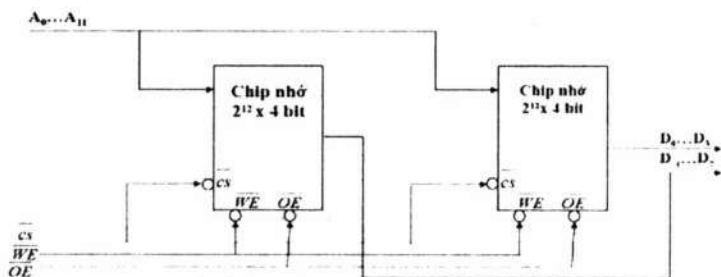
### 3.1.2.3. Thiết kế modul nhớ bán dẫn

*Ví dụ 1:* Cho chip nhớ  $2n \times m$  bit.

- Yêu cầu: Thiết kế module nhớ có dung lượng là bội của chip nhớ trên.
- Giải pháp:
  - Tăng độ dài từ nhớ.
  - Tăng số ô nhớ.

Cho các chip nhớ SDRAM dung lượng  $4K \times 4$  bit. Hãy thiết kế module nhớ có kích thước  $4K \times 8$  bit. Dung lượng chip nhớ  $2^{12} \times 4$  bit.

Thông tin cần cho chip nhớ: số đường địa chỉ  $n = 12$  và số đường dữ liệu  $m = 4$ . Thông tin về module nhớ: số đường địa chỉ là 12 (số ngăn nhớ không thay đổi), số đường dữ liệu là 8 và số chip sử dụng thiết kế 2.



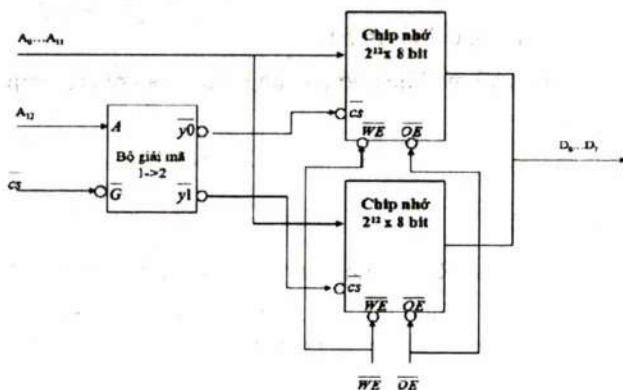
Hình 3.10. Module nhớ kích thước 4K x 8 bit theo ví dụ 1

Ví dụ 2: Cho các chip nhớ có dung lượng  $2n \times m$  bit.

- Yêu cầu: Thiết kế module nhớ có kích thước:  $2k \times 2n \times m$  bit
- Giải quyết: Để thiết kế được ta xác định hai thông số  $n + k$  (số đường địa chỉ) và  $2k$  (số chip nhớ cần để ghép vào module thiết kế).

Cho các chip nhớ SDRAM dung lượng 4K x 8 bit. Hãy thiết kế module nhớ có kích thước 8K x 8 bit.

- Dung lượng chip nhớ giả thiết là  $2^{12} \times 8$  bit.
- Thông tin cần cho chip nhớ: số đường địa chỉ  $n = 12$  và số đường dữ liệu  $m = 8$ .
- Thông tin về module nhớ: số đường địa chỉ là 13 (số ngăn nhớ thay đôi) và số đường dữ liệu là 8 (độ dài từ nhớ không đôi).

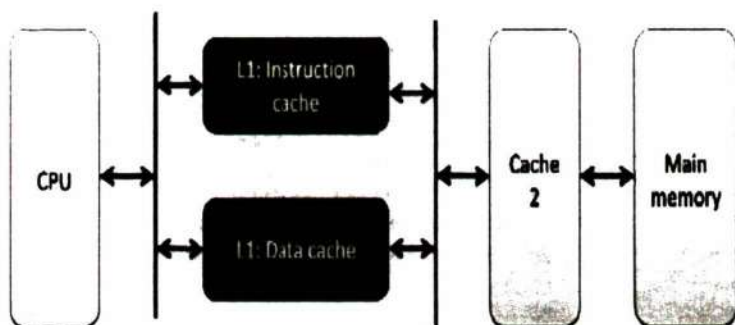


Hình 3.11. Module nhớ 8K x 8 bit theo ví dụ 2



### 3.1.3. Bộ nhớ đệm (Cache)

Mức cache trong bảng các cấp bộ nhớ có cơ cấu vận hành trong suốt đối với bộ xử lý. Với thao tác đọc bộ nhớ, bộ xử lý gửi một địa chỉ và nhận một dữ liệu từ bộ nhớ trong. Với thao tác ghi bộ nhớ, bộ xử lý viết một dữ liệu vào một ô nhớ với một địa chỉ được chỉ ra trong bộ nhớ. Để cho chương trình vận hành bình thường, cache phải chứa một phần nhỏ của bộ nhớ trong để bộ xử lý có thể thâm nhập vào các lệnh hoặc dữ liệu thường dùng từ bộ nhớ cache. Do dung lượng của bộ nhớ cache nhỏ, nên nó chỉ chứa một phần chương trình nằm trong bộ nhớ trong. Để đảm bảo sự đồng nhất giữa nội dung của cache và bộ nhớ trong, cache và bộ nhớ trong phải có cùng cấu trúc. Việc chuyển dữ liệu giữa cache và bộ nhớ trong là việc tải lên hay ghi xuống các khối dữ liệu. Mỗi khối chứa nhiều từ bộ nhớ, tùy thuộc vào cấu trúc bộ nhớ cache. Sự lựa chọn kích thước của khối rất quan trọng cho vận hành của cache có hiệu quả.



Hình 3.12. Bộ nhớ đệm cache

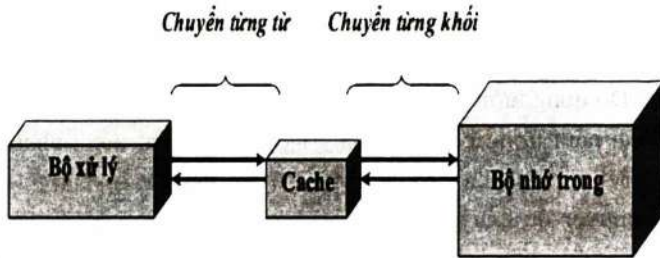
- L1 cache: Nằm trong CPU, tốc độ truy suất rất nhanh.
- L2 cache: Cache nằm giữa CPU và RAM, cấu tạo từ RAM tĩnh.
- L3 cache: Vùng nhớ trung gian giữa đĩa cứng và các thiết bị ngoại vi; cấu tạo từ DRAM.

Trước khi khảo sát vận hành của cache, ta xét đến các khái niệm liên quan:

- Thành công cache (*cache hit*): Bộ xử lý tìm gặp phần tử cần đọc (ghi) trong cache.

– Thất bại cache (*cache miss*): Bộ xử lý không gặp phần tử cần đọc (ghi) trong cache.

– Trừng phạt thất bại cache (*cache penalty*): Thời gian cần thiết để xử lý một thất bại cache. Thời gian bao gồm thời gian thâm nhập bộ nhớ trong cộng với thời gian chuyển khối chứa từ cần đọc từ bộ nhớ trong đến cache.



Hình 3.13. Trao đổi dữ liệu giữa các thành phần CPU – Cache – Bộ nhớ trong

Bảng 3.3. Kích thước Cache của một số hệ thống

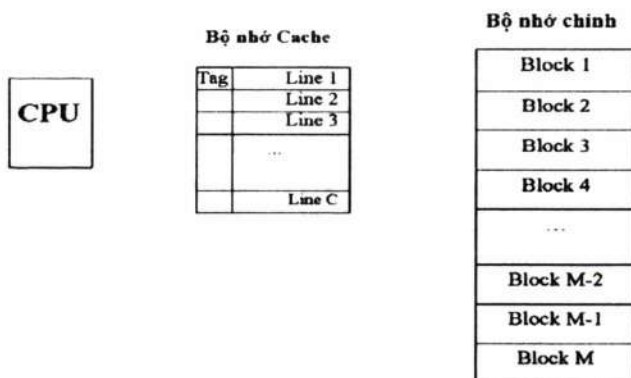
Bộ xử lý	Kiểu	Năm phát hành	L1 Cache*	L2 Cach	L1 Cache
IBM 360/85	Mainframe	1968	16 to 32 KB	-	-
PDP-11/70	Mini Computer	1975	1 KB	-	-
VAX 11/780	Mini Computer	1978	16 KB	-	-
IBM 3033	Mainframe	1978	64 KB	-	-
IBM 3090	Mainframe	1985	128 to 256 KB	-	-
Intel 80486	PC	1989	8 KB	-	-
Pentium	PC	1993	8 KB / 8 KB	256 - 512 KB	-
PowerPC 601	PC	1993	32 KB	-	-
PowerPC 620	PC	1996	32 KB / 32 KB	-	-
PowerPC G4	PC/Server	1999	32 KB / 32 KB	256KB to 1MB	2 MB
IBM S390/G4	Mainframe	1997	32 KB	256 KB	2 MB
IBM S390/G6	Mainframe	1999	256 KB	8 MB	-
Pentium 4	PC/Server	2000	8 KB / 8 KB	256 KB	-
IBM SP	High-End server/ Super Computer	2000	64 KB / 32 KB	8 MB	-
CRAY MTA <sup>b</sup>	Super Computer	2000	8 KB	2 MB	-
Itanium	PC/Server	2001	16 KB / 16 KB	96 KB	2 MB
SGI Origin 2001	High-End server	2001	32 KB / 32 KB	4 MB	-

– Hai giá trị cách nhau bởi dấu "/" chỉ giá trị cache lệnh và cache dữ liệu.

– Cả hai giá trị đều là cache lệnh.

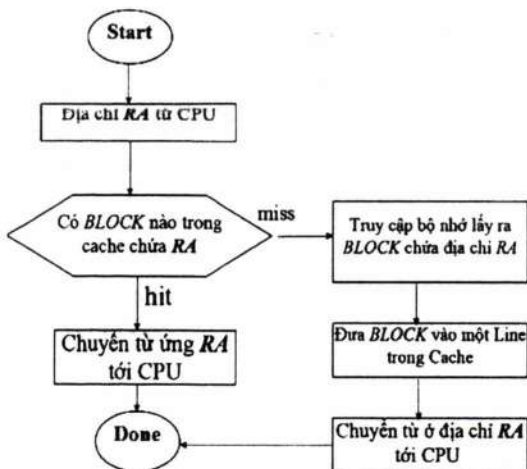
### 3.1.3.1. Cấu tạo cache

- Cache chia thành các hàng có độ rộng (386 : 32 bit; 486 : 128 bit; pentium: 256 bit).
- Tag: Địa chỉ bắt đầu của đoạn bộ nhớ được đưa vào cache.
- Cache được điều khiển bởi bộ điều khiển cache (*Cache Controller*) → Phản ánh chiến lược lưu trữ đệm của cache.



Hình 3.14. Cấu tạo cache

### 3.1.3.2. Hoạt động của cache



Hình 3.15. Hoạt động của cache

Các phương pháp ánh xạ địa chỉ. Một hàng (*line* – dòng) của cache lưu một vài khối (*block*) của bộ nhớ, Tag cho biết block nào của bộ nhớ đang chứa trong line đó. Số dòng < số khối → Cần có "chiến lược" ánh xạ địa chỉ thông tin từ bộ nhớ và cache.

• **Ánh xạ trực tiếp:**

$$B0 \rightarrow L0; B1 \rightarrow L1; B_{m-1} \rightarrow L_{m-1}; B_m \rightarrow L_0...$$

• **Ánh xạ liên kết toàn phần:**

Mỗi một block có thể được nạp vào một line bất kỳ.

• **Ánh xạ liên kết tập hợp:**

Cache được chia thành các tập  $S_0, S_1, \dots$ . Mỗi tập lại có một số line nhất định. Mỗi block được đưa vào 1 line nào đó trong một tập nhất định.

• **Giải thuật thay thế Block trong cache:**

– Cache Miss (Miss: trượt, lỡ) → Nạp block chứa thông tin vào cache để thay thế block cũ.

– Ánh xạ trực tiếp: Chỉ có duy nhất một cách thay thế.

– Ánh xạ liên kết: Sử dụng một trong các giải thuật.

+ Random: Thay thế block ngẫu nhiên.

+ FIFO: Thay thế block tồn tại lâu nhất.

+ LFO: Thay thế block có số lần nạp ít nhất.

+ LRU: Thay thế block có khoảng thời gian dài nhất không được truy nhập → hiệu quả nhất.

*Câu hỏi 1:* Phải để một khối bộ nhớ vào chỗ nào của cache (sắp xếp khối)?

*Trả lời:* Một khối bộ nhớ được đặt vào trong cache theo một trong ba cách sau:

– Kiểu tương ứng trực tiếp: Nếu mỗi khối bộ nhớ chỉ có một vị trí đặt khối duy nhất trong cache được xác định theo công thức:

$$K = i \bmod n$$

Trong đó: K: vị trí khối đặt trong cache;

i: số thứ tự của khối trong bộ nhớ trong;

n: số khối của cache.

Như vậy, trong kiểu xếp đặt khối này, mỗi vị trí đặt khối trong cache có thể chứa một trong các khối trong bộ nhớ cách nhau  $x : n$  khối ( $x: 0, 1, \dots, m; n$ : số khối của cache).

*Ví dụ:* Số thứ tự của khối cache trong bộ nhớ trong:

Số thứ tự khối cache	Số thứ tự của khối trong bộ nhớ trong
0	0, n, 2n ... mn
1	1, n + 1, 2n + 1 ... mn + 1
...	...
n - 1	n - 1, 2n - 1 ... mn - 1

– Kiểu hoàn toàn phối hợp: Trong kiểu đặt khối này, một khối trong bộ nhớ trong có thể được đặt vào vị trí bất kỳ trong cache. Như vậy, mỗi vị trí đặt khối trong cache có thể chứa một trong tất cả các khối trong bộ nhớ.

– Kiểu phối hợp theo tập hợp: Với cách tổ chức này, cache bao gồm các tập hợp của các khối cache. Mỗi tập hợp của các khối cache chứa số khối như nhau. Một khối của bộ nhớ trong có thể được đặt vào một số vị trí, khối giới hạn trong tập hợp được xác định bởi công thức:

$$K = i \text{ mod } s$$

Trong đó: K: vị trí khối đặt trong cache;

i: số thứ tự của khối trong bộ nhớ trong;

s: số lượng tập hợp trong cache.

Trong cách đặt khối kiểu phối hợp theo tập hợp, nếu tập hợp có m khối, sự tương ứng giữa các khối trong bộ nhớ trong và các khối của cache được gọi là phối hợp theo tập hợp m khối.

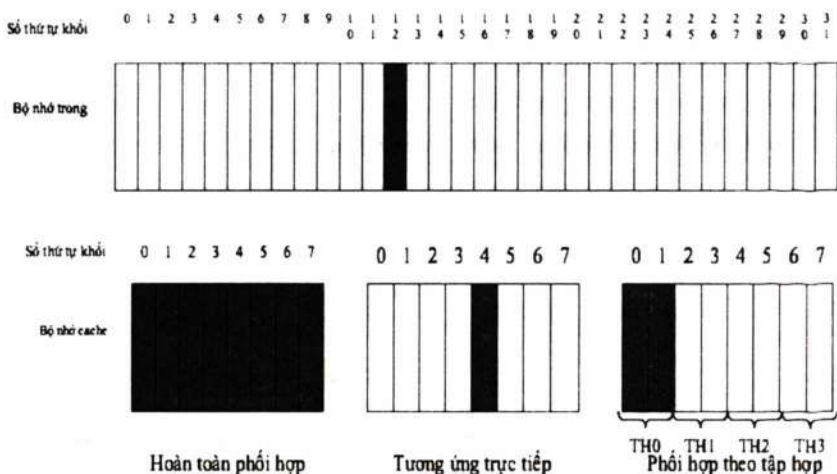
+ Nếu  $m = 1$  (mỗi tập hợp có 1 khối), ta có kiểu tương ứng trực tiếp.

+ Nếu  $m = n$  (n: số khối của cache), ta có kiểu tương ứng hoàn toàn phối hợp.

Hiện nay, phần lớn các cache của các bộ xử lý đều là kiểu tương ứng trực tiếp hay kiểu phối hợp theo tập hợp (mỗi tập hợp gồm 2 hoặc 4 khối).

*Ví dụ:* Bộ nhớ trong có 32 khối, cache có 8 khối, mỗi khối gồm 32 byte, khối thứ 12 của bộ nhớ trong được đưa vào cache.



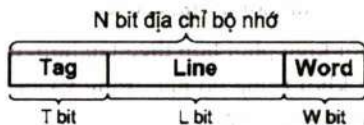


Hình 3.16. Bộ nhớ trong 32 khối và 8 khối cache

### 3.1.3.3. Các thuật toán ánh xạ (mapping) giữa cache và RAM

#### a. Ánh xạ trực tiếp (Direct mapping)

Như đã trình bày, trong phương pháp ánh xạ trực tiếp thì khối nhớ (block) thứ  $j$  đưa vào cache chắc chắn sẽ được đưa vào dòng (Line) thứ  $j \bmod m$  của cache (với  $m$  là tổng số lượng line trong cache). Do vậy, để biết được Block nào được đưa vào Line đang xét, sử dụng trường Tag của Line đó để xác định.



Hình 3.17. Cấu trúc địa chỉ bộ nhớ theo phương pháp ánh xạ trực tiếp

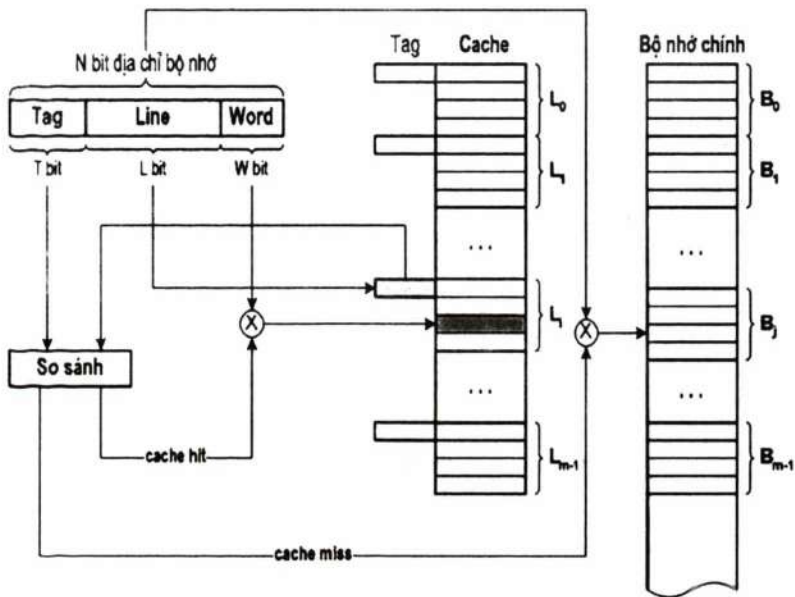
Giả sử CPU cần truy xuất ô nhớ có địa chỉ  $X$ , gồm  $N$  bit.  $N$  bit đó sẽ được chia thành 3 phần: Tag (bit) – Line (bit) – Word (bit) với ý nghĩa:

- Word (có  $W$  bit chính là số bit để đánh địa chỉ các ô nhớ trong 1 Line hay 1 Block): Giá trị của trường Word cho biết ô nhớ CPU đang tìm nằm ở chính xác vị trí nào trong Line/Block (vì nó lưu giá trị địa chỉ ô nhớ trong Line/Block).



- Line (có L bit chính là số bit để đánh địa chỉ các Line trong cache): Giá trị của trường Line cho biết ô nhớ CPU đang tìm nằm ở chính xác Line nào trong cache.

- Tag (có T bit =  $N - L - W$ ): Cho biết Block nào của RAM sẽ được lưu ở Line đang xét, do vậy giá trị của trường Tag (T bit) ở đây dùng để kiểm tra xem Line đang xét có thực sự lưu thông tin của ô nhớ (được chứa bởi Block trong RAM) đang tìm hay không.



Hình 3.18. Thuật toán xác định vị trí của ô nhớ theo phương thức ánh xạ trực tiếp.

Các bước xác định vị trí của ô nhớ nằm trên cache hay RAM:

- Bước 1: Lấy giá trị Line trong N bit của X ra, đây là giá trị cho biết ô nhớ đang tìm nếu nằm trong cache chỉ có thể nằm ở Line này.

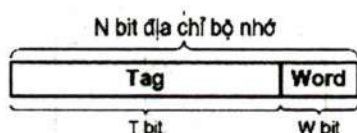
- Bước 2: Lấy giá trị Tag của Line đó ra, so khớp với giá trị Tag của X, có 2 khả năng:

+ Bằng nhau (Cache hit): Chắc chắn ô nhớ cần tìm đang nằm ở Line này, vấn đề còn lại là nằm ở ô nhớ nào trong các ô nhớ của Line đó. Dễ dàng xác định nhờ vào giá trị Word (địa chỉ ô nhớ trong Line/Block).

+ Không bằng nhau (Cache miss): Chắc chắn ô nhớ cần tìm không nằm ở Line đó, nó nằm ở RAM. Để xác định nó nằm ở Block nào trong RAM, cần kết hợp hai giá trị Tag và Line của X, ta có địa chỉ của Block đang chứa ô nhớ đó trong RAM. Vấn đề còn lại là nằm ở ô nhớ nào trong các ô nhớ của Block đó. Dễ dàng xác định nhờ vào giá trị Word (địa chỉ ô nhớ trong Line/Block).

### b. Ánh xạ liên kết toàn phần (Full Associative mapping)

Phương pháp này cho phép 1 khối nhớ (Block) khi chuyển vào cache sẽ có thể nằm ở bất kỳ dòng (Line) nào trong cache, miễn sao Line đó đang còn trống. Nhưng cũng chính vì cách thức tổ chức như vậy, nên khi CPU muốn tìm ô nhớ nào đó phải duyệt lần lượt từng Line một trong cache cho đến khi gặp thì thôi. Đối với phương pháp này, trường địa chỉ ô nhớ X cần tìm gồm N bit sẽ không cần bận tâm đến Line nữa (vì trước đó nó lưu ở Line nào cũng được) mà chỉ còn Tag và Word thôi.



*Hình 3.19. Cấu trúc địa chỉ bộ nhớ theo phương pháp ánh xạ liên kết toàn phần*

Các bước xác định vị trí của ô nhớ nằm trên cache hay RAM:

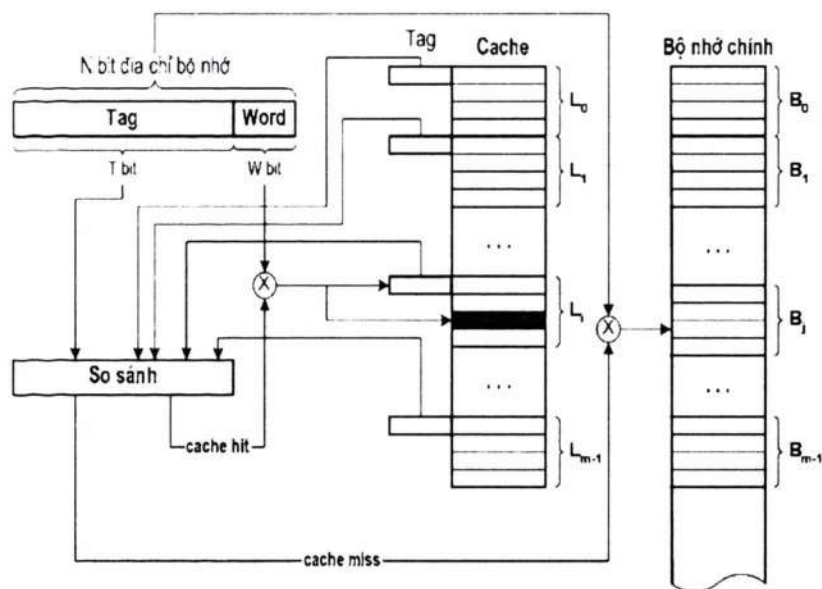
– Bước 1: Duyệt lần lượt từng Line trong cache. Ứng với mỗi Line lấy giá trị Tag của nó ra.

– Bước 2: So khớp giá trị lấy ra ở bước 1 với giá trị Tag của X, có 2 khả năng:

+ Bằng nhau (Cache hit): Chắc chắn ô nhớ cần tìm đang nằm ở Line này, vấn đề còn lại là nằm ở ô nhớ nào trong các ô nhớ của Line đó. Dễ dàng xác định nhờ vào giá trị Word (địa chỉ ô nhớ trong Line/Block).

+ Không bằng nhau (Cache miss): Chắc chắn ô nhớ cần tìm không nằm ở Line đó, nó nằm ở RAM. Để xác định nằm ở Block nào trong RAM, chỉ cần lấy giá trị Tag của X, ta sẽ có địa chỉ của Block đang chứa ô nhớ đó

trong RAM. Vấn đề còn lại là nằm ở ô nhớ nào trong các ô nhớ của Block đó. Để dàng xác định nhờ vào giá trị Word (địa chỉ ô nhớ trong Line/Block)

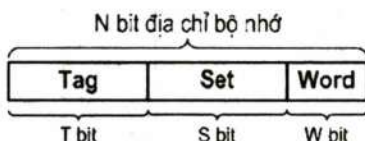


*Hình 3.20. Thuật toán xác định vị trí của ô nhớ theo phương thức ánh xạ liên kết toàn phần*

### c. Ánh xạ liên kết tập hợp (Set associative mapping)

Phương pháp này là sự kết hợp của 2 phương pháp Direct mapping và Associative mapping bằng cách tận dụng các ưu điểm và cố gắng giảm bớt các khuyết điểm. Thay vì khi tìm phải duyệt từ Line đầu tiên cho đến khi gặp Line chứa ô nhớ cần tìm thì dừng (trường hợp xấu nhất là phải duyệt hết tất cả các Line), đây là khuyết điểm của Associative mapping, để khắc phục nó, sử dụng Set với 1 Set gồm nhiều Line (1 Set = 2/4/8/16... Line). Ví dụ như 4 Line/Set – 4-way associative mapping. Thay vì 1 Block khi đưa vào cache sẽ cho phép nằm ở Line nào cũng được trong cache, nó sẽ chỉ cho phép nằm ở Line nào cũng được trong 1 Set mà thôi (do vậy khi tìm chỉ cần duyệt lần lượt từng Line trong Set chứ không phải từng Line trong toàn bộ cache nữa, từ đó rút ngắn phạm vi tìm kiếm). Tuy nhiên, việc chọn Set nào để đưa Block vào thì phải tuân thủ theo nguyên tắc của Direct mapping.

Đối với phương pháp này, trường địa chỉ ô nhớ X cần tìm gồm N bit sẽ không cần quan tâm đến Line nữa (vì trước đó nó lưu ở Line nào cũng được), mà chỉ còn Tag, Set và Word thôi; với trường Set cho biết 1 Block từ RAM trước đó đưa vào sẽ nằm ở chính xác Set nào trong cache.



**Hình 3.21.** Cấu trúc địa chỉ bộ nhớ theo phương pháp ánh xạ liên kết tập hợp

Các bước xác định vị trí của ô nhớ nằm trên cache hay RAM:

– Bước 1: Lấy giá trị Set trong N bit của X ra, đây là giá trị cho biết ô nhớ đang tìm nếu nằm trong cache chỉ có thể nằm ở Set này.

– Bước 2: Duyệt lần lượt từng Line trong Set, ứng với mỗi Line lấy giá trị Tag của nó ra.

– Bước 3: So khớp giá trị lấy ra ở bước 1 với giá trị Tag của X, có 2 khả năng:

+ Bằng nhau (Cache hit): Chắc chắn ô nhớ cần tìm đang nằm ở Line này, vấn đề còn lại là nằm ở ô nhớ nào trong các ô nhớ của Line đó. Dễ dàng xác định nhờ vào giá trị Word (địa chỉ ô nhớ trong Line/Block).

+ Không bằng nhau (Cache miss): Chắc chắn ô nhớ cần tìm không nằm ở Line đó, nó nằm ở RAM. Để xác định nằm ở Block nào trong RAM, chỉ cần lấy giá trị Tag + Set của X, ta sẽ có địa chỉ của Block đang chứa ô nhớ đó trong RAM. Vấn đề còn lại là nằm ở ô nhớ nào trong các ô nhớ của Block đó. Dễ dàng xác định nhờ vào giá trị Word (địa chỉ ô nhớ trong Line/Block).

### 3.1.4. Bộ nhớ ngoài

Trong phần trước đã khảo sát bộ nhớ trong của máy tính và thấy được chúng có những ưu điểm về tốc độ rất lớn và làm việc trực tiếp với CPU. Tuy nhiên, chúng có giới hạn về dung lượng cũng như giá cả khá đắt. Hơn



nữa, bộ nhớ RAM bị mất dữ liệu khi bị mất điện, còn ROM thì chỉ ghi lại được 1 lần. Để có thể lưu giữ dữ liệu và di chuyển chúng một cách độc lập, ta cần bộ nhớ khác có khả năng lưu dữ liệu khi không có điện và di chuyển được dễ dàng hơn. Bộ nhớ đó là bộ nhớ ngoài, bao gồm đĩa từ, đĩa quang, CDROM và một số ổ đĩa khác.

#### 3.1.4.1. Đĩa từ

Đĩa từ là thiết bị lưu trữ thông tin bằng cách từ hoá các hạt từ trên một mặt đĩa theo một quy luật đã được định trước.

Đĩa từ được chia thành 2 loại:

- Đĩa cứng có dung lượng lớn, cố định, truy xuất nhanh.
- Đĩa mềm dung lượng nhỏ, dễ dàng di chuyển, truy xuất chậm.

Đĩa từ được phân chia thành các phần nhỏ để lưu trữ thông tin với các thông số vật lý như sau:

– Trụ (*Cylinder*): Các trụ đồng trục với nhau. Người ta có thể chia đĩa ra thành nhiều trụ. Đường cắt giữa trụ với mặt đĩa được gọi là track. Mặc dù vậy, trụ và vết thường được dùng thay thế cho nhau.

– Đầu (*Head*): Là đầu từ dùng để đọc và ghi. Thường mỗi đầu từ chuyên dùng để đọc một mặt duy nhất, nên khái niệm số đầu cũng chính là số mặt đĩa. Các đầu từ được điều khiển bởi một động cơ bước và có thể dịch chuyển vào/ra theo chiều hướng tâm. Khoảng cách giữa hai bước dịch chuyển đầu từ quy định khoảng cách giữa hai trụ; nói khác đi, số trụ phân chia lệ thuộc vào số bước dịch chuyển của đầu từ.

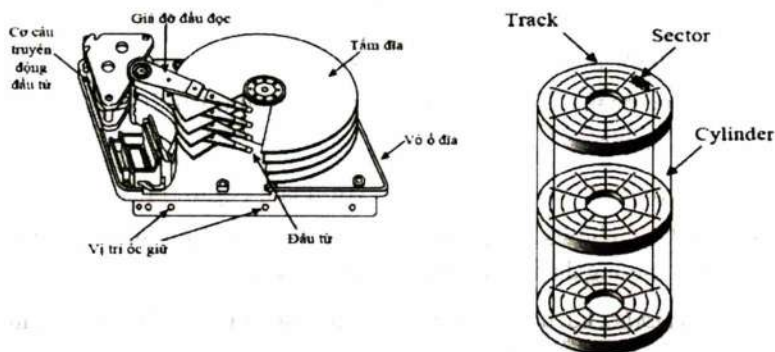
– Mẫu tin (*Record*) hay cung (*Sector*): Trên mỗi vết người ta chia ra thành các cung tròn bằng nhau chính là mẫu tin. Thường mỗi mẫu tin có khả năng giữ được 512 byte dữ liệu.

Dung lượng đĩa từ = Số track  $\times$  Số sector/track  $\times$  Số mặt  $\times$  512 byte.

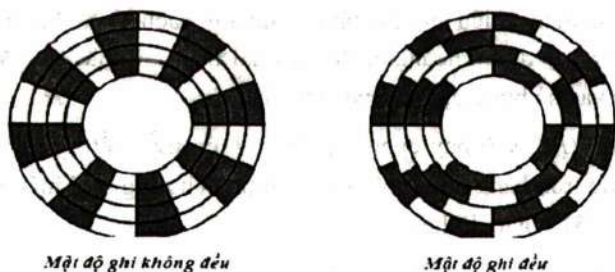
Một đĩa cứng chứa nhiều lớp đĩa (từ 1 đến 4), quay quanh một trục khoảng 3600 – 15.000 vòng mỗi phút. Các lớp đĩa này được làm bằng kim loại với hai mặt được phủ một chất từ tính. Đường kính của đĩa thay đổi từ 1,3 inch đến 8 inch. Mỗi mặt của một lớp đĩa được chia thành nhiều đường



tròn đồng trục gọi là rãnh. Thông thường, mỗi mặt của một lớp đĩa có từ 10.000 đến gần 30.000 rãnh. Mỗi rãnh được chia thành nhiều cung, dùng để chứa thông tin. Một rãnh có thể chứa từ 64 đến 800 cung. Cung là đơn vị nhỏ nhất mà máy tính có thể đọc hoặc viết (thông thường khoảng 512 bytes). Chuỗi thông tin ghi trên mỗi cung gồm có: số thứ tự của cung, một khoảng trống, số liệu của cung đó bao gồm cả các mã sửa lỗi, một khoảng trống, số thứ tự của cung tiếp theo. Với kỹ thuật ghi mật độ không đều, tất cả các rãnh đều có cùng một số cung, điều này làm cho các cung dài hơn ở các rãnh xa trục quay có mật độ ghi thông tin thấp hơn mật độ ghi trên các cung nằm gần trục quay.



Hình 3.22. Đĩa từ



Hình 3.23. Mật độ ghi dữ liệu trên các đĩa cứng

Với công nghệ ghi với mật độ đều, người ta cho ghi nhiều thông tin hơn ở các rãnh xa trục quay. Công nghệ ghi này ngày càng được dùng nhiều với sự ra đời của các chuẩn giao diện thông minh như chuẩn SCSI.

**Bảng 3.4.** Bảng thông số kỹ thuật đĩa cứng

Dung lượng tối đa	Có thể đạt 500GB
Số lượng đầu đọc	1 – 8
Số tấm ghi (đĩa)	1 – 4
Cache (bộ đệm)	2 – 16 MB
Số cung (Sector – 521 bye/sector)	xxx,xxx,xxx
Tốc độ quay đĩa (RPM)	3600 – 15000
Mật độ	Có thể đạt 95 GB/in <sup>2</sup>
Mật độ rãnh (TPI – Max Tracks/Inch)	Có thể đạt 12,000
Mật độ ghi BPI (Max Bits/Inch)	Có thể đạt 702,000
Tốc độ dữ liệu tối đa (Internal)	Có thể đạt 900 Mbit/s
Tốc độ truyền dữ liệu với ngoại vi	Có thể đạt 302 Mbit/s
Thời gian truyền track RW	Có thể đạt 15 ms
Thời gian quay nửa vòng	Có thể đạt 6 ms

Để đọc hoặc ghi thông tin vào một cung, ta dùng một đầu đọc ghi di động áp vào mỗi mặt của mỗi lớp đĩa. Các đầu đọc/ghi này được gắn chặt vào một thanh làm cho chúng cùng di chuyển trên một đường bán kính của mỗi lớp đĩa; và như thế, tất cả các đầu này đều ở trên những rãnh có cùng bán kính của các lớp đĩa. Từ "trụ" (*Cylinder*) được dùng để gọi tất cả các rãnh của các lớp đĩa có cùng bán kính và nằm trên một hình trụ. Người ta luôn muốn đọc nhanh đĩa từ, nên thông thường ổ đĩa đọc nhiều hơn số dữ liệu cần đọc. Công nghiệp chế tạo đĩa từ tập trung vào việc nâng cao dung lượng của đĩa mà đơn vị đo lường là mật độ trên một đơn vị bề mặt.

#### 3.1.4.2. Đĩa quang

Các thiết bị lưu trữ quang rất thích hợp cho việc phát hành các sản phẩm văn hoá, sao lưu dữ liệu trên các hệ thống máy tính hiện nay. Ra đời

vào những năm 1978, đây là sản phẩm của sự hợp tác nghiên cứu giữa hai công ty là Sony và Philips trong công nghiệp giải trí. Từ năm 1980 đến nay, công nghiệp đĩa quang phát triển mạnh trong cả hai lĩnh vực giải trí và lưu trữ dữ liệu máy tính.

Quá trình đọc thông tin dựa trên sự phản chiếu của các tia laser năng lượng thấp từ lớp lưu trữ dữ liệu. Bộ phận tiếp nhận ánh sáng sẽ nhận biết được những điểm mà tại đó tia laser bị phản xạ mạnh hay biến mất do các vết khắc (bit) trên bề mặt đĩa. Các tia phản xạ mạnh chỉ ra rằng, tại điểm đó không có lỗ khắc và điểm này được gọi là điểm nền (*land*). Bộ nhận ánh sáng trong ổ đĩa thu nhận các tia phản xạ và khuếch tán được khúc xạ từ bề mặt đĩa. Khi các nguồn sáng được thu nhận, bộ vi xử lý sẽ dịch các mẫu sáng thành các bit dữ liệu hay âm thanh. Các lỗ trên CD sâu 0,12 micron và rộng 0,6 micron (1 micron bằng một phần ngàn mm). Các lỗ này được khắc theo một track hình xoắn ốc với khoảng cách 1,6 micron giữa các vòng, khoảng 16.000 track/inch. Các lỗ (*pit*) và nền (*land*) kéo dài khoảng 0,9 đến 3,3 micron. Track bắt đầu từ phía trong và kết thúc ở phía ngoài theo một đường khép kín các rìa đĩa 5 mm. Dữ liệu lưu trên CD thành từng khối, mỗi khối chứa 2352 byte. Trong đó, 304 byte chứa các thông tin về bit đồng bộ, bit nhận dạng (ID), mã sửa lỗi (ECC), mã phát hiện lỗi (EDC). Còn lại 2048 byte chứa dữ liệu. Tốc độ đọc chuẩn của CD-ROM là 75 khối/s, hay 153.600 byte/s, hay 150KB/s (1X).

Dưới đây là một số loại đĩa quang thông dụng:

- **CD (Compact Disk):** Đĩa quang không thể xoá được, dùng trong công nghiệp giải trí (các đĩa âm thanh được số hoá). Chuẩn đĩa có đường kính 12 cm, âm thanh phát từ đĩa khoảng 60 phút (không dừng).

- **CD-ROM (Compact Disk Read Only Memory):** Đĩa không xoá, dùng để chứa các dữ liệu máy tính. Chuẩn đĩa có đường kính 12 cm, lưu trữ dữ liệu hơn 650 MB. Khi phát hành, đĩa CD-ROM đã có chứa nội dung. Thông thường, đĩa CD-ROM được dùng để chứa các phần mềm và các chương trình điều khiển thiết bị.

- **CD-R (CD - Recordable):** Giống như đĩa CD, đĩa mới chưa có thông tin, người dùng có thể ghi dữ liệu lên đĩa một lần và đọc được nhiều lần. Dữ liệu trên đĩa CD-R không thể bị xoá.

- **CD-RW** (*CD – Rewritable*): Giống như đĩa CD, đĩa mới chưa có thông tin, người dùng có thể ghi dữ liệu lên đĩa, xoá và ghi lại dữ liệu trên đĩa nhiều lần.

- **DVD** (*Digital Video Disk – Digital Versatile Disk*): Ra đời phục vụ cho công nghiệp giải trí, đĩa chứa các hình ảnh video được số hoá. Ngày nay, DVD được sử dụng rộng rãi trong các ứng dụng công nghệ thông tin. Kích thước đĩa có hai loại: 8 cm và 12 cm. Đĩa DVD có thể chứa dữ liệu trên cả hai mặt đĩa, dung lượng tối đa lên đến 17 GB. Tốc độ đọc chuẩn (1X) của DVD là 1.3 MB/s (1X của DVD tương đương khoảng 9X của CDROM).

- **DVD-R** (*DVD – Recordable*): Giống như đĩa DVD-ROM, người dùng có thể ghi dữ liệu lên đĩa một lần và đọc được nhiều lần. Đĩa này chỉ có thể ghi được trên một mặt đĩa, dung lượng ghi trên mỗi mặt tối đa là 4,7 GB.

- **DVD-RW** (*DVD – Rewritable*): Giống như đĩa DVD-ROM, người dùng có thể ghi, xoá và ghi lại dữ liệu lên đĩa nhiều lần. Đĩa này cũng có thể ghi được trên một mặt đĩa, dung lượng ghi trên mỗi mặt tối đa là 4,7 GB.

So sánh một số thông số của hai loại đĩa CD-ROM và DVD-ROM:

**Bảng 3.5. Bảng so sánh CD-ROM và DVD-ROM**

Đặc trưng	CD-ROM	DVD-ROM
Kích thước pit	0,834 micron	0,4 micron
Khoảng cách rãnh	1,6 micron	0,74 micron
Số lớp dữ liệu trên đĩa	1 lớp	2 lớp
Số mặt đĩa	1 mặt	1 – 2 mặt
Dung lượng	640 – 700 MB	1,36 – 17 GB
Độ phân giải phim	VCD = 320 x 200	720 x 640

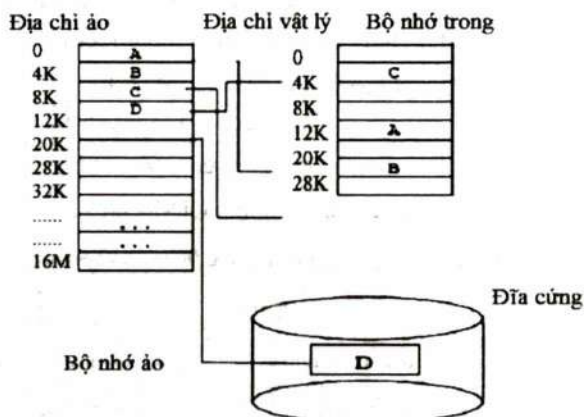
### 3.1.5. Bộ nhớ ảo

Bộ nhớ ảo xác định một cơ chế vận chuyển tự động số liệu giữa bộ nhớ trong và bộ nhớ ngoài (đĩa từ).

Trước đây, khi độ dài của chương trình vượt quá giới hạn dung lượng bộ nhớ, người lập trình phải phân chia chương trình của mình thành từng



phân tự loại bỏ nhau (*Overlays*) và phải tự quản lý việc trao đổi thông tin giữa bộ nhớ và đĩa từ. Bộ nhớ ảo làm nhẹ trách nhiệm của các nhà lập trình bằng cách làm cho việc trao đổi thông tin này được thực hiện một cách tự động. Trong các bộ xử lý hiện đại, bộ nhớ ảo được dùng để cho phép thực hiện cùng lúc nhiều tiến trình (*Process*), mỗi tiến trình có một không gian định vị riêng. Nếu tất cả các không gian định vị này đều thuộc không gian định vị bộ nhớ trong thì rất tốn kém. Bộ nhớ ảo bao gồm bộ nhớ trong và bộ nhớ ngoài, được phân tích thành khối để có thể cung cấp cho mỗi chương trình một số khối cần thiết cho việc thực hiện chương trình đó. Hình bên dưới cho thấy một chương trình chứa trong bộ nhớ ảo gồm 4 khối, ba trong bốn khối nằm ở bộ nhớ trong, khối thứ tư nằm trên đĩa.



**Hình 3.24.** Một chương trình gồm 4 trang A, B, C, D trong đó trang D nằm trong ổ đĩa

Ngoài việc phân chia không gian bộ nhớ, cần bảo vệ và quản lý tự động các cấp bộ nhớ, bộ nhớ ảo đơn giản hoá việc nạp chương trình vào bộ nhớ để thi hành nhờ một cơ chế được gọi là sự tái định địa chỉ (*Address relocation*). Cơ chế này cho phép một chương trình có thể được thi hành khi nó nằm ở bất cứ vị trí nào trong bộ nhớ.

### 3.1.5.1. Các thông số điển hình cho bộ nhớ cache và bộ nhớ ảo

Ngoài sự khác biệt định lượng mà ta đã biết, có những khác biệt khác giữa bộ nhớ cache và bộ nhớ ảo là:



– Khi thất bại cache, sự thay thế một khối trong cache được điều khiển bằng phần cứng, trong khi sự thay thế trong bộ nhớ ảo chủ yếu là do hệ điều hành.

– Không gian định vị mà bộ xử lý quản lý là không gian định vị của bộ nhớ ảo, trong khi đó dung lượng bộ nhớ cache không tùy thuộc vào không gian định vị bộ xử lý.

**Bảng 3.6.** Các thông số điển hình cho bộ nhớ cache và bộ nhớ ảo

Tham số	Cache	Bộ nhớ ảo
Chiều dài mỗi khối (trang)	16 – 128 bytes	4096 – 65.536 byte
Thời gian thâm nhập thành công	1 – 2 xung nhịp	40 – 100 xung nhịp
Trúng phạt khi thất bại	8 – 100 xung nhịp	700.000 – 6 triệu xung
Thời gian thâm nhập	6 – 60 xung	500.000 – 4 triệu xung
Di chuyển số liệu	2 – 40 xung	200.000 – 2 triệu xung
Tỷ số thất bại	0,5% – 10%	0,00001% – 0,001%
Dung lượng	8 KB – 8 MB	16 MB – 8 GB

Bộ nhớ ngoài còn được dùng để lưu trữ tập tin ngoài nhiệm vụ là hậu phương của bộ nhớ trong (trong các cấp bộ nhớ).

Bộ nhớ ảo cũng được thiết kế bằng nhiều kỹ thuật đặc thù cho chính nó. Các hệ thống bộ nhớ ảo có thể được chia thành hai loại: loại với khối có dung lượng cố định gọi là trang; và loại với khối có chiều dài thay đổi gọi là đoạn. Định vị trang xác định một địa chỉ trong trang, giống như định vị trong cache.

Trong định vị đoạn cần 2 từ: một từ chứa số thứ tự đoạn; một từ chứa độ dài trong đoạn. Chương trình dịch gặp khó khăn nhiều hơn trong định vị đoạn.

Do việc thay thế các đoạn, ngày nay ít máy tính dùng định vị đoạn thuần túy. Một vài máy dùng cách hỗn hợp gọi là đoạn trang. Trong đó mỗi đoạn chứa một số nguyên các trang.

### 3.1.5.2. Bảo vệ tiến trình bằng bộ nhớ ảo

Sự xuất hiện của đa chương trình (*Multiprogram*), trong đó máy tính chạy nhiều chương trình song song với nhau, dẫn tới các đòi hỏi mới về việc bảo vệ và phân chia giữa các chương trình.

Đa chương trình đưa đến khái niệm tiến trình (*Process*): Một tiến trình gồm có một chương trình đang thực hiện và tất cả các thông tin cần thiết để tiếp tục thực hiện chương trình này.

Trong đa chương trình, bộ xử lý và bộ nhớ trong được nhiều người sử dụng chia sẻ một cách qua lại (*Interactive*), cùng một thời điểm, để tạo cảm giác rằng, mỗi người dùng đang có một máy tính riêng. Và như thế, tại bất cứ lúc nào, phải có thể chuyển đổi từ một tiến trình này sang một tiến trình khác.

Một tiến trình phải vận hành đúng đắn, dù nó được thi hành liên tục từ đầu tới cuối, hay nó bị ngắt qua lại bởi các tiến trình khác. Trách nhiệm đảm bảo các tiến trình đều chạy đúng, được chia sẻ giữa nhà thiết kế máy tính và nhà thiết kế hệ điều hành. Nhà thiết kế máy tính phải đảm bảo bộ xử lý có thể lưu giữ trạng thái các tiến trình và phục hồi các trạng thái này, còn nhà thiết kế hệ điều hành phải đảm bảo các tiến trình không ảnh hưởng lên nhau. Hệ điều hành giải quyết vấn đề này bằng cách chia bộ nhớ trong cho các tiến trình; và trạng thái của mỗi tiến trình này hiện diện trong phần bộ nhớ được chia cho nó. Điều này có nghĩa rằng, các nhà thiết kế hệ điều hành phải được sự giúp sức của các nhà chế tạo máy tính để bảo vệ một tiến trình không bị ảnh hưởng bởi tiến trình khác.

## 3.2. HỆ THỐNG VÀO/RA

### 3.2.1. Tổng quan hệ thống vào/ra

#### *Chức năng của hệ thống vào/ra:*

Trao đổi thông tin giữa máy tính với thế giới bên ngoài.

- **Các thao tác cơ bản:** Vào (input) và ra (output).
- **Các thành phần chính:**
  - Các thiết bị ngoại vi.
  - Module ghép nối vào/ra.
  - Đặc điểm của hệ thống vào/ra.
  - Tồn tại nhiều thiết bị ngoại vi khác nhau.
  - Nguyên tắc hoạt động.
  - Tốc độ.
  - Khuôn dạng dữ liệu.

Tất cả các thiết bị ngoại vi đều chậm hơn CPU và RAM → Cần có các module vào/ra để ghép nối các thiết bị ngoại vi với CPU và bộ nhớ chính.

• **Phân loại thiết bị ngoại vi**

– Thiết bị ngoại vi giao tiếp người – máy: Chuột, bàn phím, màn hình, máy in...

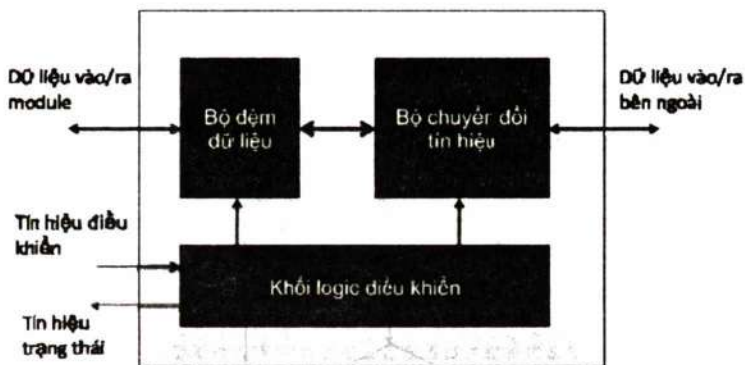
– Thiết bị ngoại vi giao tiếp máy – máy: Các thiết bị theo dõi và kiểm tra.

– Thiết bị truyền thông: Modem, Card mạng...

– Bộ chuyển đổi tín hiệu: Chuyển đổi dữ liệu giữa bên ngoài và bên trong máy.

– Bộ đệm dữ liệu: Đệm dữ liệu khi truyền giữa module vào/ra với thiết bị ngoại vi.

– Khối logic điều khiển: Điều khiển hoạt động của thiết bị ngoại vi, đáp ứng theo yêu cầu từ module vào/ra.



Hình 3.25. Cấu trúc chung của một thiết bị ngoại vi

• **Module vào/ra:**

– Điều khiển và định thời.

– Trao đổi thông tin với CPU.

– Trao đổi thông tin với thiết bị ngoại vi.

– Đệm giữa bên trong máy tính với thiết bị ngoại vi.

– Phát hiện lỗi của các thiết bị ngoại vi.

- **Thanh ghi đệm dữ liệu:** Đệm dữ liệu trong quá trình trao đổi.
- **Các cổng vào/ra (I/O):** Kết nối với thiết bị ngoại vi, mỗi cổng có một địa chỉ xác định.
- **Thanh ghi trạng thái điều khiển:** Lưu giữ thông tin/trạng thái điều khiển cho các cổng vào/ra.
- **Khối logic điều khiển:** Điều khiển module vào/ra.

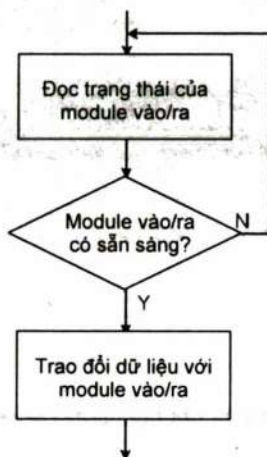
### 3.2.2. Các phương pháp điều khiển vào/ra

- Vào/ra bằng chương trình (*Programmed IO*).
- Vào/ra điều khiển bằng ngắt (*Interrupt Driven IO*).
- Truy nhập bộ nhớ trực tiếp – DMA (*Direct Memory access*).

#### 3.2.2.1. Vào/ra bằng chương trình

##### • Nguyên tắc chung

- CPU điều khiển trực tiếp vào/ra bằng chương trình.
- Kiểm tra trạng thái của thiết bị ngoại vi.
- Phát tín hiệu điều khiển đọc ghi.
- Trao đổi dữ liệu.



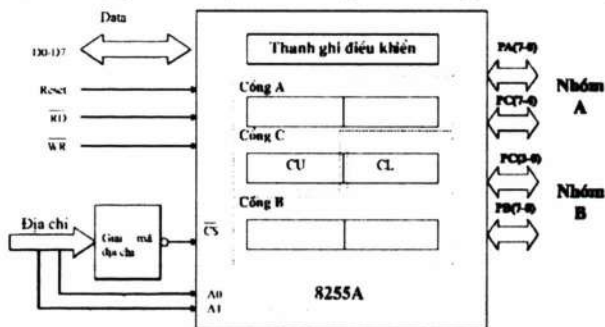
Hình 3.26. Lưu đồ thuật toán vào/ra bằng chương trình

• **Đặc điểm**

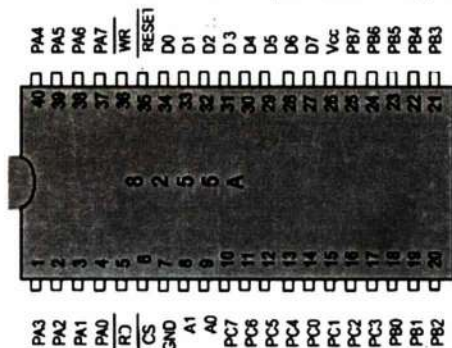
- Vào/ra theo ý muốn người lập trình.
- CPU trực tiếp điều khiển vào/ra.
- CPU đợi module vào/ra → tốn thời gian của CPU.
- Mạch ghép nối vào/ra song song khả trình 8255A.

• **Chế độ làm việc của 8255**

- Chế độ 0: Chế độ vào/ra đơn giản. Ba cổng A, B, C đều có thể lập trình thành cổng vào hoặc ra (không thể điều khiển riêng cho từng bit).
- Chế độ 1: Cổng A hoặc B có thể được dùng làm cổng vào hoặc ra với khả năng bắt tay, các tín hiệu bắt tay được truyền qua các bit của cổng C.
- Chế độ 2: Cổng A có thể được dùng làm cổng vào/ra hai chiều với khả năng bắt tay với các tín hiệu của cổng C. Cổng B có thể ở chế độ 0 hoặc 1.
- Chế độ BSR (lập/xoá bit): Có thể lập trình từng bit cho cổng C.



Hình 3.27. Sơ đồ cấu trúc vi mạch 8255A và ghép nối



Hình 3.28. Sơ đồ chân chip 8255A



**Bảng 3.7.** Chọn cổng và truyền dữ liệu của 82255A

A1	A0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Phương thức truyền
0	0	0	1	0	Cổng A tới bus dữ liệu
0	1	0	1	0	Cổng B tới bus dữ liệu
1	0	0	1	0	Cổng C tới bus dữ liệu
0	0	1	0	0	Bus dữ liệu tới cổng A
0	1	1	0	0	Bus dữ liệu tới cổng B
1	0	1	0	0	Bus dữ liệu tới cổng C
1	1	1	0	0	– Nếu D7 = 1: Bus dữ liệu tới thanh ghi điều khiển; – Nếu D7 = 0: Dữ liệu đầu vào từ bus dữ liệu sẽ được chuyển thành lệnh Set/Reset
x	x	x	x	1	D7 – D0 được đưa về trạng thái trở kháng cao
1	1	0	1	0	Trạng thái cấm
x	x	1	1	0	D7 – D0 được đưa về trạng thái trở kháng cao

### 3.2.2.2. Nguyên tắc vào/ra bằng ngắt

**Ngắt:** CPU tạm dừng công việc hiện tại để trao đổi dữ liệu. Được sử dụng để tận dụng khả năng xử lý của CPU cho nhiều công việc.

#### • Nguyên tắc chung

- CPU không phải đợi trạng thái sẵn sàng của module vào/ra.
- Khi module sẵn sàng thì nó phát tín hiệu ngắt CPU.
- CPU thực hiện chương trình con vào/ra tương ứng để trao đổi dữ liệu.
- CPU trở lại tiếp tục thực hiện chương trình đang bị ngắt.

#### • Đặc điểm

- Có sự kết hợp giữa phần cứng và phần mềm.

- CPU trực tiếp điều khiển vào/ra.
- CPU không phải đợi module vào/ra → hiệu quả tốt hơn.

#### • Phương thức hoạt động

- Module vào/ra:
- + Module vào/ra nhận trực tiếp điều khiển đọc từ CPU.

Module vào/ra nhận dữ liệu từ thiết bị ngoại vi trong khi CPU đang làm việc khác.

Module vào/ra phát tín hiệu ngắt CPU.

CPU yêu cầu dữ liệu, module vào/ra chuyển dữ liệu tới CPU.

- CPU:
- + Phát tín hiệu điều khiển đọc.
- + Làm việc khác.
- + Cuối mỗi chu kỳ lệnh kiểm tra tín hiệu ngắt.
- + Nếu bị ngắt:
- \* Cắt ngữ cảnh (nội dung các thanh ghi).
- \* Thực hiện chương trình con ngắt để vào các dữ liệu.
- \* Khôi phục ngữ cảnh của chương trình đang thực hiện.

#### • Các phương pháp nối ghép ngắt

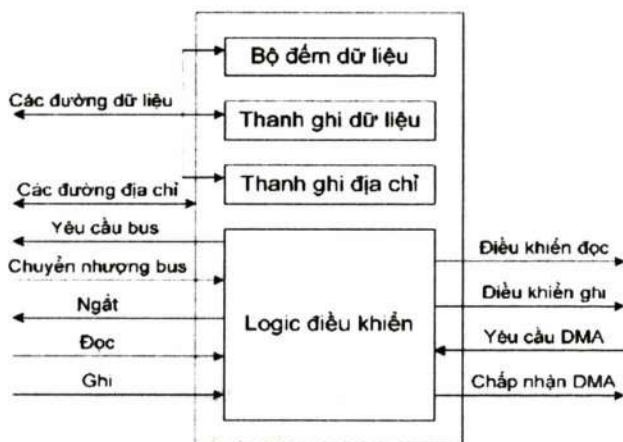
- Sử dụng nhiều đường yêu cầu ngắt.
- Kiểm tra vòng bằng phần mềm (Software poll).
- Kiểm tra vòng bằng phần cứng (Hardware poll).
- Sử dụng bộ điều khiển ngắt PIC.

#### 3.2.2.3. Truy cập bộ nhớ trực tiếp – DMA (Direct Memory access)

Vào/ra bằng chương trình và ngắt: CPU trực tiếp tham gia điều khiển.

- Chiếm thời gian của CPU.
- Tốc độ hạn chế vì phải chuyển qua CPU.
- Khắc phục dùng phương pháp DMA.
- Thêm module phần cứng trên BUS → DMA controller (DMAC)

- DMA điều khiển vào/ra thông qua CPU.
- Thanh ghi dữ liệu: Chứa dữ liệu trao đổi.
- Thanh ghi địa chỉ: Chứa địa chỉ ngăn nhớ dữ liệu.
- Bộ đếm dữ liệu: Chứa số từ dữ liệu cần trao đổi.
- Logic điều khiển: Điều khiển hoạt động của DMAC.
- Nguyên lý hoạt động của DMA.



Hình 3.29. Sơ đồ cấu trúc DMAC

+ Khi cần CPU "nhờ" DMA thực hiện vào/ra với các thông số sau:

- \* Vào hay ra dữ liệu.
- \* Địa chỉ thiết bị vào/ra.
- \* Địa chỉ đầu của mảng nhớ dữ liệu → nạp vào thanh ghi địa chỉ.
- \* Số từ dữ liệu cần truyền → nạp vào bộ đếm dữ liệu.

+ CPU làm việc khác.

+ DMAC điều khiển trao đổi dữ liệu.

+ Sau khi truyền 1 word:

- \* Nội dung thanh ghi địa chỉ tăng.

- \* Nội dung bộ đếm dữ liệu giảm, khi bộ đếm = 0 → DMAC gửi tín hiệu ngắt đến CPU để báo kết thúc DMA.

+ Đặc điểm DMA:

- \* CPU không tham gia quá trình trao đổi dữ liệu.
- \* DMAC điều khiển trao đổi dữ liệu giữa bộ nhớ chính với module vào/ra (hoàn toàn bằng phần cứng) → tốc độ nhanh. Phù hợp với yêu cầu trao đổi mảng dữ liệu có kích thước lớn.

### 3.2.3. Các cổng vào/ra thông dụng

#### 3.2.3.1. Cổng nối tiếp COM

Cổng COM được sử dụng cho những thiết bị phải giao tiếp hai chiều với hệ thống.

##### • Đặc trưng

- Giao tiếp nối tiếp được thiết kế không có sự đồng bộ tín hiệu.
- Mỗi ký tự được truyền nối tiếp và được tổ chức theo kiểu có bit bắt đầu (start) và bit dừng (stop).
- Bảng mạch chính hiện nay đều có chip I/O.
- Độ dài tối đa của các dây cáp nối là 15 m.
- Có hai loại bộ kết nối cổng nối tiếp là 9 chân và 25 chân.

#### 3.2.3.2. Cấu hình cổng nối tiếp

Mỗi lần tiếp nhận một ký tự, cổng nối tiếp thực hiện một yêu cầu ngắt IRQ.

Trong cấu hình chuẩn, COM1 sử dụng IRQ4; COM2 sử dụng IRQ3. Khi một cổng nối tiếp được cài đặt vào hệ thống, nó sẽ được cấu hình để sử dụng địa chỉ I/O, còn gọi là cổng và các ngắt IRQ.

##### • Thông số một cấu hình cho các cổng nối tiếp

- COM1 3F8–3FF IRQ4
- COM2 2F8–2FF IRQ3
- COM3 3F8–3EF IRQ4\*
- COM4 2F8–2EF IRQ3\*

Các cổng COM3, COM4 có thể dùng chung ngắt với COM1 và COM2, nhưng tốt nhất là nên cấu hình COM3 cho IRQ10, COM4 cho IRQ11.

### 3.2.3.3. Cổng song song

Cổng song song ban đầu được thiết kế với mục đích duy nhất là để kết nối máy in, gần đây nó còn được sử dụng để kết nối với các thiết bị khác.

Ban đầu cổng song song chỉ cho phép truyền dữ liệu theo một hướng, hiện nay nó cho phép có thể truyền/nhận dữ liệu.

Các cổng song song có 8 đường truyền, cho phép truyền đồng thời 1 byte dữ liệu. Cổng song song có hạn chế về chiều dài cáp (không quá 3 m) do độ lệch tín hiệu của các đường truyền, cáp càng dài thì độ lệch càng lớn.

#### • Các loại cổng song song

– Cổng song song chuẩn (*Normal*).

– Cổng song song hai chiều còn được gọi là cổng song song PS/2, hay cổng song song mở rộng. Các cổng này cho phép đường dữ liệu vào rộng 8 bit.

– Cổng song song EPP (*Enhanced Parallel Port*) còn gọi là cổng song song Fast Mode. EPP có tốc độ xấp xỉ tốc độ bus ISA, tốc độ truyền lên tới 2 MB/s, nhanh gấp 10 lần cổng song song thường.

– Cổng song song ECP (*Enhanced Capability Port*): Năm 1992, Microsoft hợp tác với Hewlett – Packard đưa ra thiết kế cổng song song tốc độ cao mới ECP, nó dành riêng cho máy in và máy quét, nó khác với các loại cổng song song khác là sử dụng DMA. Đa số PC hiện nay có Super I/O hỗ trợ cả ba chế độ SPP, EPP và ECP. Việc chuyển đổi các chế độ được thực hiện trong BIOS Setup. Chúng ta nên chọn chế độ ECP để có tốc độ truyền dữ liệu lớn nhất.

#### • Cấu hình cổng song song

BIOS đã hỗ trợ tới 3 cổng song song, bảng địa chỉ cổng I/O và IRQ của chúng:

– LPT1 3BC–3BF IRQ7.

– LPT2 378–37A IRQ5.

– LPT3 278–27A IRQ5.



#### 3.2.3.4. Cổng USB (*Universal Serial Bus – đường truyền nối tiếp chung*)

– USB là sản phẩm của sự hợp tác giữa các tập đoàn công nghiệp PC và viễn thông như: Compaq, DEC, IBM, Intel, Microsoft, NEC, Northern Telecom.

– USB mở đường cho việc đưa các thiết bị ngoại vi ra ngoài PC. Một thiết bị khi cắm vào hệ thống sẽ được tự động cấu hình mà không yêu cầu khởi động lại máy hay chạy các chương trình Setup.

– USB cho phép kết nối tối đa 127 thiết bị đồng thời. Nó thay thế cho các công nối tiếp và song song và có tốc độ vượt xa các công này.

– USB đưa lại sự hợp nhất cho các thiết bị ngoại vi, tức là mọi thiết bị ngoại vi đều được kết nối với máy tính thông qua cổng này.

– Muốn sử dụng được các thiết bị USB thì BIOS, bảng mạch chính và hệ điều hành đều phải hỗ trợ USB. Các bảng mạch ATX đều có cổng USB và BIOS hỗ trợ USB, hệ điều hành từ Windows 95 trở đi đều có các trình điều khiển USB

– Các thiết bị USB được cung cấp nguồn thông qua cáp USB, mỗi khi thiết bị được cắm vào, hệ thống sẽ xác định yêu cầu của thiết bị về nguồn điện, nếu yêu cầu vượt quá khả năng của hệ thống thì hệ thống sẽ đưa ra lời cảnh báo.

Một ưu điểm nữa là tất cả các thiết bị được kết nối vào một cổng USB chỉ dùng chung một ngắt.

#### 3.2.3.5. Cổng VGA (*Video Graphic Adapter – cổng nối ghép màn hình*)

VGA là bo mạch đồ họa, bo mạch đồ họa là bo mạch cắm thêm vào máy tính, có nhiệm vụ chuyển các hình ảnh được tạo bên trong máy tính thành các tín hiệu điện tử cần thiết mà màn hình máy tính có thể hiển thị. Nó quyết định số lượng màu, tần số quét và độ phân giải tối đa có thể được hiển thị.

Trên nhiều bo mạch chủ của máy tính cá nhân, các mạch đồ họa được tích hợp vào trong chipset. Khi ấy, không cần phải lắp thêm bất kỳ một bo mạch đồ họa rời nào nếu như không có nhu cầu. Trên các bo mạch đồ họa luôn có không gian nhớ riêng. Không gian nhớ này được sử dụng vào việc

xây dựng các mô hình điểm ảnh trước khi chúng thực sự được hiển thị lên màn hình. Ở các máy tính cá nhân thế hệ đầu tiên, các bo mạch đồ họa (chuẩn CGA, EGA...) gửi các tín hiệu số đến màn hình máy tính. Sau đó, màn hình máy tính lại chuyển chúng sang dạng tín hiệu tương tự để dùng cho ống CRT. Chỉ từ khi bắt đầu với chuẩn VGA, các tín hiệu tương tự mới được xuất thẳng sang màn hình máy tính từ các bo mạch đồ họa. Ngày nay, bo mạch đồ họa có khả năng xuất tín hiệu qua hai cổng: tương tự và số (DVI). Cổng tín hiệu số được sử dụng cho các màn hình tấm phẳng (LCD) mới hiện nay. Đối với máy tính xách tay, mạch đồ họa là loại đầu mút số (*Digital end to end*) và được tích hợp sẵn trên bo mạch chủ. Tuy mạch đồ họa có thể được tích hợp sẵn trong bo mạch chủ, việc sử dụng một bo mạch lắp rời sẽ cần thiết đối với các ứng dụng cần đến khả năng xử lý đồ họa cao như: game hay các trình xử lý đồ họa 3 chiều (3Ds MAX chẳng hạn).

Mọi thứ bắt đầu vào năm 1973, khi hãng Xerox trình làng Alto, máy vi tính đầu tiên trên thế giới sử dụng giao diện người dùng đồ họa (GUI). Và chính sự cách tân này đã thay đổi hoàn toàn cách thức mà chúng ta làm việc với máy tính. Ngày nay, tất cả mọi lĩnh vực sử dụng máy tính, từ việc tạo các ảnh động đến các tác vụ cơ bản như xử lý văn bản, thư điện tử đều sử dụng hình ảnh để tạo ra môi trường làm việc trực quan hơn cho người sử dụng.

### 3.3. GHÉP NỐI THIẾT BỊ NGOẠI VI

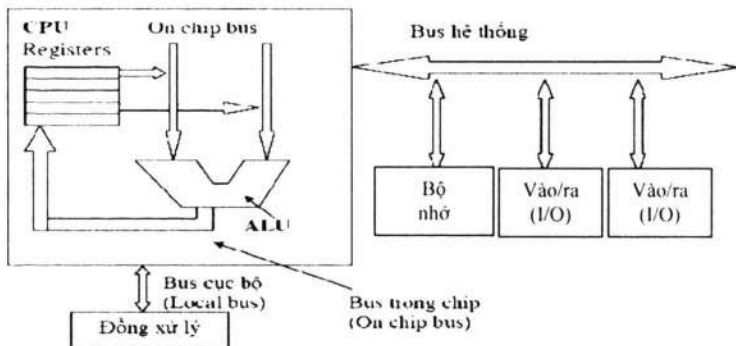
#### 3.3.1. Bus và các vấn đề truyền thông trong máy tính

Bus là đường truyền tín hiệu điện nối các thiết bị khác nhau trong một hệ thống máy tính.

Bus thường có từ 50 đến 100 dây dẫn được gắn trên Mainboard, trên các dây này có các đầu nối đưa ra, các đầu này được sắp xếp và cách nhau những khoảng quy định để có thể cắm vào đó những I/O board hay board bộ nhớ (*bus hệ thống – system bus*).

Trong vi xử lý cũng có một số bus để nối các thành phần bên trong của bộ vi xử lý với nhau. Người thiết kế chip vi xử lý có thể tùy ý lựa chọn loại bus bên trong nó, còn với các bus liên hệ bên ngoài, cần phải xác định rõ các quy tắc làm việc, cũng như các đặc điểm kỹ thuật về điện và cơ khí của

bus để người thiết kế Mainboard có thể ghép nối chip vi xử lý với các thiết bị khác. Nói cách khác, các bus này phải tuân theo một chuẩn nào đó. Tập các quy tắc của chuẩn còn được gọi là giao thức bus (*Bus protocol*).



Hình 3.30. Các bus trong một hệ thống máy tính

Ngoài ra, có rất nhiều loại bus khác nhau được sử dụng, các bus này nói chung là không tương thích với nhau.

Bus thường phân loại theo 3 cách sau:

- Theo tổ chức phần cứng (như trên).
- Theo giao thức truyền thông (bus đồng bộ và không đồng bộ).
- Theo loại tín hiệu truyền trên bus (bus địa chỉ, bus dữ liệu...).

#### • Thiết bị chủ và tớ

Nhiều thiết bị nối ghép và trao đổi thông tin với nhau qua các bus điều khiển, bus địa chỉ và bus dữ liệu. Khi một thiết bị muốn trao đổi thông tin với thiết bị khác, đầu tiên nó cần phải chuyển địa chỉ để phân biệt thiết bị, bởi vì mỗi thiết bị bao giờ cũng có một địa chỉ duy nhất. Đồng thời, nó cũng gửi đi một tín hiệu ghi hoặc đọc để xác định hành động. Thiết bị chủ (*Master*) là thiết bị khởi đầu và điều khiển việc trao đổi thông tin, còn thiết bị đáp lại gọi là thiết bị tớ (*Slave*). Ở các máy tính 80×86, CPU là ví dụ về thiết bị chủ, còn bộ nhớ, ổ đĩa... là các thiết bị tớ.

#### • Phân phối bus

Trong hệ thống máy tính, các thiết bị chủ có thể truy cập được các bus toàn cục, bao gồm bus địa chỉ, bus dữ liệu và bus điều khiển. Nếu có nhiều



thiết bị chủ có yêu cầu sử dụng bus, thì bus phải được phân phối theo một trình tự nhất định, bởi vì không có bus nào phục vụ hai thiết bị chủ cùng một lúc. Để thiết bị chủ truy nhập được bus, đầu tiên nó phải có tín hiệu yêu cầu bộ phân phối bus trung tâm, sau đó phải nhận được tín hiệu trả lời. Phân phối bus trung tâm có thể phân công việc truy nhập của từng thiết bị chủ theo một trình tự ưu tiên, hoặc theo nguyên tắc "đến trước – phục vụ trước".

#### • **Giao thức bus (*Bus protocol*)**

Để phối hợp hoạt động giữa các bộ phận khác nhau của hệ thống, các bus phải tuân theo một loạt các tiêu chuẩn về tín hiệu và định thời. Thuật ngữ giao thức bus muốn đề cập tới các tiêu chuẩn này. Có hai giao thức bus chính là: đồng bộ và không đồng bộ.

Đối với giao thức đồng bộ, các bus được đồng bộ với một tần số trung tâm, đó là tần số của hệ thống. Ở máy IBM, CPU truy nhập bộ nhớ sử dụng giao thức đồng bộ, vì bộ nhớ không thể sai lệch với các tiêu chuẩn định thời của bộ tạo giao động đồng hồ trung tâm.

Giao thức không đồng bộ tuân theo phương pháp định thời riêng, nó có thể không hoạt động theo tần số đồng hồ trung tâm. Lấy máy in làm ví dụ, như đã trình bày, nếu CPU gửi dữ liệu đến máy in, nó phải liên tục giám sát tín hiệu bận của máy in. Chỉ khi máy in rỗi (sẵn sàng) thì CPU mới có thể phát dữ liệu tới bus dữ liệu được. CPU phải phát tín hiệu báo cho máy in biết dữ liệu đã sẵn sàng và chờ tín hiệu báo nhận của máy in. Ở phương pháp không đồng bộ, CPU là thiết bị chủ và máy in là thiết bị tớ. Thiết bị tớ (máy in) hoạt động theo sự định thời riêng so với tín hiệu báo nhận, không phụ thuộc vào tần số hệ thống. Tuy nhiên, khi trao đổi thông tin giữa CPU – bộ nhớ, các tiêu chuẩn định thời của bộ nhớ là phù hợp với tần số hệ thống và CPU không thăm dò bộ nhớ, để xem bộ nhớ đã sẵn sàng để nhận dữ liệu hay không. Giao thức không đồng bộ được sử dụng khi việc định thời bus giữa chủ và tớ khác nhau. Thiết bị tớ thông thường chậm hơn so với thiết bị chủ, thường là bằng một nửa. Điều này khác với ở giao thức đồng bộ, định thời của chủ và tớ là như nhau. Giao thức đồng bộ nói chung có tốc độ truyền dữ liệu cao hơn so với giao thức không đồng bộ.

**• Dài thông của bus (*Bus bandwidth*)**

Tốc độ mà bus có thể truyền dữ liệu từ thiết bị chủ tới thiết bị tớ gọi là dài thông, hay độ rộng dài của bus (*Bandwidth*) hoặc năng suất truyền của bus (*Throughput*).

Đơn vị đo là Mbit/s. Dài thông phụ thuộc vào tốc độ, độ rộng và giao thức của bus.

Tần số đồng hồ càng cao thì dài thông càng lớn. Ở các bộ vi xử lý 386 trở về trước, năng suất truyền của bus thấp, do bộ vi xử lý ở mỗi chu trình nhớ chỉ chuyển dữ liệu dạng từ vào bộ nhớ. Đối với 486 và pentium, việc chuyển dữ liệu ở chế độ khối cho dài thông lớn hơn nhiều nhờ chuyển dữ liệu dạng 4 từ 32 bit chỉ trong 5 nhịp đồng hồ so với bình thường là 8 nhịp. Một giải pháp khác để nâng cao dài thông của bus là tăng độ rộng bus dữ liệu. Với mục đích này, Intel Pentium sử dụng bus dữ liệu ngoài 64 bit. Đường truyền tin nối tiếp có dài thông nhỏ nhất; trái lại, bus dữ liệu 64 bit có dài thông lớn nhất.

**3.3.2. Các kiểu ghép nối vào/ra****3.3.2.1. Giao tiếp nối tiếp****• Cấu trúc cổng nối tiếp**

Cổng nối tiếp được sử dụng để truyền dữ liệu hai chiều giữa máy tính và ngoại vi, có các ưu điểm sau:

- Khoảng cách truyền xa hơn truyền song song.
- Số dây kết nối ít.
- Có thể truyền không dây (dùng hồng ngoại).
- Có thể ghép nối với vi điều khiển hay PLC (*Programmable Logic Device*).
- Cho phép nối mạng.
- Có thể tháo lắp thiết bị trong lúc máy tính đang làm việc.
- Có thể cung cấp nguồn cho các mạch điện đơn giản.



Các thiết bị ghép nối chia thành hai loại: DTE (*Data Terminal Equipment*) và DCE (*Data Communication Equipment*). DCE là các thiết bị trung gian như MODEM, còn DTE là các thiết bị tiếp nhận hay truyền dữ liệu như máy tính, PLC, vi điều khiển... Việc trao đổi tín hiệu thông thường qua hai chân RxD (nhận) và TxD (truyền). Các tín hiệu còn lại có chức năng hỗ trợ để thiết lập và điều khiển quá trình truyền, được gọi là các tín hiệu bắt tay (*Handshake*). Ưu điểm của quá trình truyền dùng tín hiệu bắt tay là có thể kiểm soát đường truyền.

Tín hiệu truyền theo chuẩn RS-232 của EIA (*Electronics Industry Associations*). Chuẩn RS-232 quy định mức logic 1 ứng với điện áp từ  $-3V$  đến  $-25 V$  (*Mark*); mức logic 0 ứng với điện áp từ  $3 V$  đến  $25 V$  (*Space*) và có khả năng cung cấp dòng từ  $10 mA$  đến  $20 mA$ . Ngoài ra, tất cả các đầu ra đều có đặc tính chống chập mạch.

Chuẩn RS-232 cho phép truyền tín hiệu với tốc độ đến  $20000 \text{ bps}$  ( $\text{bit/s}$ ), nhưng nếu cáp truyền đủ ngắn có thể lên đến  $115200 \text{ bps}$ .

#### • Các phương thức nối giữa DTE và DCE

– Đơn công (*Simplex connection*): Dữ liệu chỉ được truyền theo một hướng.

– Bán song công (*Half – duplex*): Dữ liệu truyền theo hai hướng, nhưng mỗi thời điểm chỉ được truyền theo một hướng.

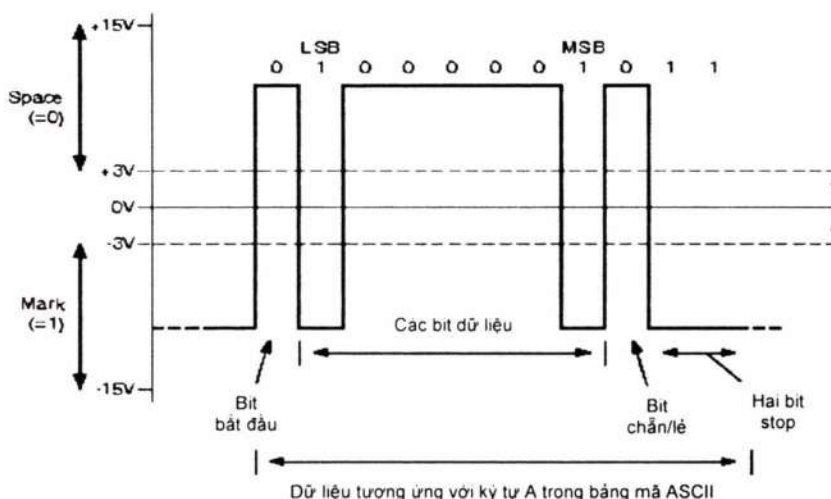
– Song công (*Full – duplex*): Dữ liệu được truyền đồng thời theo hai hướng.

Định dạng của khung truyền dữ liệu theo chuẩn RS-232 như sau:

Start	D0	D1	D2	D3	D4	D5	D6	D7	P	Stop
0										1

**Hình 3.31.** Định dạng của khung truyền dữ liệu theo chuẩn RS-232

Khi không truyền dữ liệu, đường truyền sẽ ở trạng thái mark (điện áp  $-10V$ ). Khi bắt đầu truyền, DTE sẽ đưa ra xung Start (space:  $10V$ ) và sau đó lần lượt truyền từ D0 đến D7 và Parity, cuối cùng là xung Stop (mark:  $-10V$ ) để khôi phục trạng thái đường truyền. Dạng tín hiệu truyền mô tả như sau (truyền ký tự A):



Hình 3.32. Tín hiệu truyền ký tự "A"

Các đặc tính kỹ thuật của chuẩn RS-232 như sau:

Bảng 3.8. Các đặc tính kỹ thuật của chuẩn RS-232

Chiều dài cable cực đại	15 m
Tốc độ dữ liệu cực đại	20 Kbps
Điện áp đầu ra cực đại	$\pm 15$ V
Điện áp đầu ra có tải	$\pm 5$ V đến $\pm 15$ V
Trở kháng tải	3 K đến 7 K
Điện áp đầu vào	$\pm 15$ V
Độ nhạy đầu vào	$\pm 3$ V
Trở kháng đầu vào	3 K đến 7 K

Các tốc độ truyền dữ liệu thông dụng trong công nối tiếp là: 1200 bps, 4800 bps, 9600 bps và 19200 bps.



Hình 3.33. Sơ đồ chân cổng nối tiếp

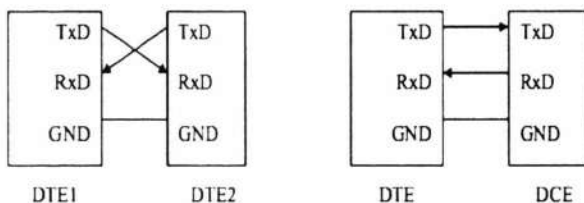
Công COM có hai dạng: Đầu nối DB25 (25 chân) và đầu nối DB9 (9 chân) mô tả như sau:

**Bảng 3.9.** Mô tả chân của COM đầu nối DB25 (25 chân) và đầu nối DB9 (9 chân)

D25	D9	Tín hiệu	Hướng truyền	Mô tả
1	-	-	-	Protected ground: Nối đất bảo vệ
2	3	TxD	DTE → DCE	Transmitted data: Dữ liệu truyền
3	2	RxD	DCE → DTE	Received data: Dữ liệu nhận
4	7	RTS	DTE → DCE	Request to send: DTE yêu cầu truyền dữ liệu
5	8	CTS	DCE → DTE	Clear to send: DCE sẵn sàng nhận dữ liệu
6	6	CTS	DCE → DTE	Data set ready: DCE sẵn sàng làm việc
7	5	GND	-	Ground: Nối đất (0V)
8	1	DCD	DCE → DTE	Data carrier detect: DCE phát hiện sóng mang
20	4	DTR	DTE → DCE	Data terminal ready: DTE sẵn sàng làm việc
22	9	RI	DCE → DTE	Ring indicator: Báo chuông
23	-	DSRD	DCE → DTE	Data signal rate detector: Dò tốc độ truyền
24	-	TSET	DTE → DCE	Transmit Signal Element Timing: Tín hiệu định thời truyền đi từ DTE
15	-	TSET	DCE → DTE	Transmitter Signal Element Timing: Tín hiệu định thời truyền từ DCE để truyền dữ liệu
17	-	RSET	DCE → DTE	Receiver Signal Element Timing: Tín hiệu định thời truyền từ DCE để truyền dữ liệu
18	-	LL		Local Loopback: kiểm tra cổng
21	-	RL	DCE → DTE	Remote Loopback: Tạo ra bởi DCE khi tín hiệu nhận từ DCE lỗi
14	-	STxD	DTE → DCE	Secondary Transmitted Data: Dữ liệu truyền thứ hai
16	-	SRxD	DCE → DTE	Secondary Received Data: Dữ liệu nhận thứ hai
19	-	SRTS	DTE → DCE	Secondary Request To Send: DTE yêu cầu truyền thứ hai
13	-	SCTS	DCE → DTE	Secondary Clear To Send: DCE sẵn sàng nhận dữ liệu thứ hai
12	-	SDSRD	DCE → DTE	Secondary Received Line Signal Detector: Dò tốc độ truyền thứ hai
25	-	TM		Test Mode: Chế độ kiểm tra
9	-			Dành riêng cho chế độ test
10	-			Dành riêng cho chế độ test
11	-			Không dùng

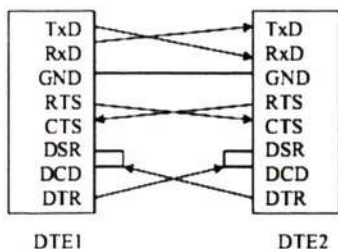
• Truyền thông giữa hai nút

Các sơ đồ khi kết nối dùng công nối tiếp:



Hình 3.34. Kết nối đơn giản trong truyền thông nối tiếp

Khi thực hiện kết nối như trên (hình 3.34), quá trình truyền phải bảo đảm tốc độ ở đầu phát và thu giống nhau. Khi có dữ liệu đến DTE, dữ liệu này sẽ được đưa vào bộ đệm và tạo ngắt. Ngoài ra, khi thực hiện kết nối giữa hai DTE, ta còn dùng sơ đồ sau:



Hình 3.35. Kết nối trong truyền thông nối tiếp

Khi DTE1 cần truyền dữ liệu thì cho DTR tích cực, tác động lên DSR của DTE2, cho biết sẵn sàng nhận dữ liệu; và cho biết đã nhận được sóng mang của MODEM (áo). Sau đó, DTE1 tích cực chân RTS để tác động đến chân CTS của DTE2, cho biết DTE1 có thể nhận dữ liệu.

Khi thực hiện kết nối giữa DTE và DCE, do tốc độ truyền dùng tín hiệu bắt tay, nên phải thực hiện điều khiển lưu lượng. Quá trình điều khiển này có thể thực hiện bằng phần mềm hay phần cứng. Quá trình điều khiển bằng phần mềm thực hiện bằng hai ký tự Xon và Xoff. Ký tự Xon được DCE gửi đi khi rảnh (có thể nhận dữ liệu); nếu DCE bận thì sẽ gửi ký tự Xoff. Quá trình điều khiển bằng phần cứng dùng hai chân RTS và CTS. Nếu DTE muốn truyền dữ liệu thì sẽ gửi RTS để yêu cầu truyền, DCE có khả năng nếu nhận dữ liệu (đang rảnh) thì gửi lại CTS.



### 3.3.2.2. Giao tiếp song song

#### • Cấu trúc cổng song song

Cổng song song gồm có 4 đường điều khiển, 5 đường trạng thái và 8 đường dữ liệu, bao gồm 4 chế độ hoạt động:

- Chế độ tương thích (*Compatibility*).
- Chế độ nửa byte, chế độ byte.
- Chế độ EPP (*Enhanced Parallel Port*).
- Chế độ ECP (*Extended Capabilities Port*).

Ba chế độ đầu tiên sử dụng port song song chuẩn (*SPP – Standard Parallel Port*), trong khi đó chế độ 4, 5 cần thêm phần cứng để cho phép hoạt động ở tốc độ cao hơn.

Bảng chức năng các chân của máy in như sau:

**Bảng 3.10.** Chức năng chân của máy in

Chân	Tín hiệu	Mô tả
1	$\overline{\text{STR}}$ (Out)	Mức tín hiệu thấp
2	D0	Bit dữ liệu 0
3	D1	Bit dữ liệu 1
4	D2	Bit dữ liệu 2
5	D3	Bit dữ liệu 3
6	D4	Bit dữ liệu 4
7	D5	Bit dữ liệu 5
8	D6	Bit dữ liệu 6
9	D7	Bit dữ liệu 7
10	$\overline{\text{ACK}}$ (In)	Mức thấp: Máy in nhận được một ký tự và có khả năng nhận nữa
11	BUSY (In)	Mức cao: Ký tự đã nhận được, bộ đệm máy in đầy, khởi động máy in; máy in ở trạng thái off-line
12	PAPER EMPTY (In)	Mức cao: Hết giấy



Chân	Tín hiệu	Mô tả
13	SELCTET (In)	Mức cao: Máy in ở trạng thái on-line
14	OUTOFEED (Out)	Tự động xuống dòng; mức thấp: máy in xuống dòng tự động
15	ERROR	Mức thấp: Máy in ở trạng thái off-line; lỗi máy in
16	INIT (Out)	Mức thấp: Khởi động máy in
17	SELECTIN (Out)	Mức thấp: Chọn máy in
18 – 25	GROUND	0 V

Cổng song song có ba thanh ghi, có thể truyền dữ liệu và điều khiển máy in. Địa chỉ cơ sở của các thanh ghi cho tất cả cổng LPT (*Line Printer*) từ LPT1 đến LPT4, được lưu trữ trong vùng dữ liệu của BIOS. Thanh ghi dữ liệu được định vị ở offset 00h, thanh ghi trạng thái ở 01h, và thanh ghi điều khiển ở 02h. Thông thường, địa chỉ cơ sở của LPT1 là 378h, LPT2 là 278h, do đó địa chỉ của thanh ghi trạng thái là 379h hoặc 279h và địa chỉ thanh ghi điều khiển là 37Ah hoặc 27Ah. Tuy nhiên, trong một số trường hợp, địa chỉ của cổng song song có thể khác do quá trình khởi động của BIOS. BIOS sẽ lưu trữ các địa chỉ này như sau:

**Bảng 3.11.** Địa chỉ của cổng song song

Địa chỉ	Chức Năng
0000h:0408h	Địa chỉ cơ sở của LPT1
0000h: 040Ah	Địa chỉ cơ sở của LPT2
0000h: 040Ch	Địa chỉ cơ sở của LPT3

#### • Định dạng các thanh ghi

**Bảng 3.12.** Bảng định dạng các thanh ghi

Thanh ghi dữ liệu hai chiều:								
	7	6	5	4	3	2	1	0
Tín hiệu máy in	D7	D6	D5	D4	D3	D2	D1	D0
Chân số	9	8	7	6	5	4	3	2

**Thanh ghi trạng thái máy in (chi đọc):**

	7	6	5	4	3	2	1	0
Tín hiệu máy in	BUSY	$\overline{\text{ACK}}$	PAPER EMPTY	SECLECT	$\overline{\text{ERROR}}$	$\overline{\text{IRQ}}$	x	x
Số chân cắm	11	10	12	13	15	-	-	-

**Thanh ghi điều khiển máy in:**

	7	6	5	4	3	2	1	0
Tín hiệu máy in	X	X	DIR	IRQ Enable	$\overline{\text{SELECTIN}}$	$\overline{\text{INIT}}$	$\overline{\text{OUTOFEED}}$	$\overline{\text{STROBE}}$
Số chân cắm	-	-	-	-	17	16	14	1

Trong đó: x: không sử dụng.

IRQ Enable: yêu cầu ngắt cứng;

1 = cho phép;

0 = không cho phép.

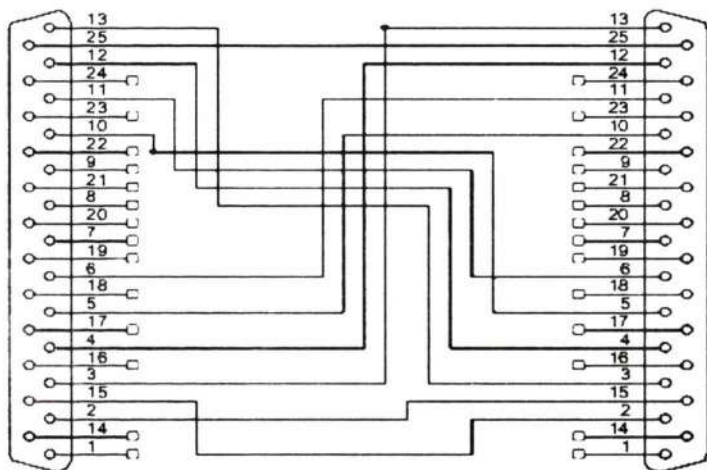
Chú ý rằng, chân BUSY được nối với công đảo trước khi đưa vào thanh ghi trạng thái, các bit SELECTIN, AUTOFEED và STROBE được đưa qua công đảo trước khi đưa ra các chân của cổng máy in.

Thông thường, tốc độ xử lý dữ liệu của các thiết bị ngoại vi như máy in chậm hơn PC nhiều, nên các đường ACK, BUSY và STR được sử dụng cho kỹ thuật bắt tay. Khởi đầu, PC đặt dữ liệu lên bus, sau đó kích hoạt đường STR xuống mức thấp để thông tin cho máy in, biết rằng dữ liệu đã ổn định trên bus. Khi máy in xử lý xong dữ liệu, nó sẽ trả lại tín hiệu ACK xuống mức thấp để ghi nhận. PC đợi cho đến khi đường BUSY từ máy in xuống thấp (máy in không bận) thì sẽ đưa tiếp dữ liệu lên bus.

**• Giao tiếp với thiết bị ngoại vi**

Quá trình giao tiếp với cổng song song dùng hai chế độ: chế độ chuẩn SPP và chế độ mở rộng.

Việc giao tiếp ở chế độ chuẩn mô tả như hình sau:



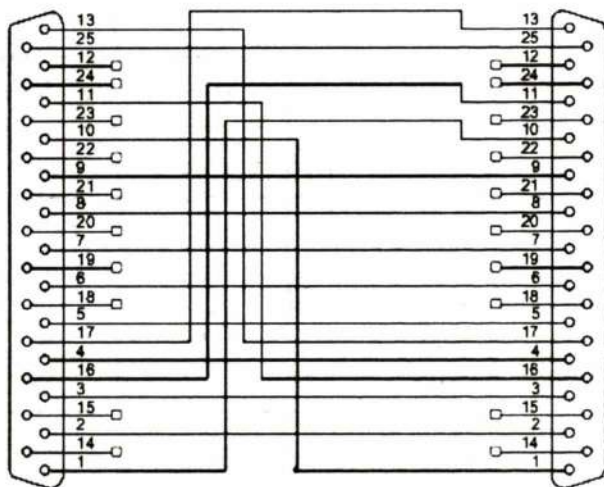
Hình 3.36. Trao đổi dữ liệu qua cổng song song giữa 2PC dùng chế độ chuẩn

Bảng kết nối chân mô tả như sau:

**Bảng 3.13.** Bảng mô tả kết nối chân PC dùng chế độ chuẩn

PC1		PC2	
Chức năng	Chân	Chân	Chức năng
D0	2	15	$\overline{\text{ERROR}}$
D1	3	13	SELECT
D2	4	12	PAPER EMPTY
D3	5	10	$\overline{\text{ACK}}$
D4	11	6	BUSY
BUSY	11	6	D4
$\overline{\text{ACK}}$	10	5	D3
PAPER EMTY	12	4	D2
SEL CET	13	3	D1
$\overline{\text{ERROR}}$	15	2	D0
GND	25	25	GND

Ngoài ra, việc kết nối giữa hai máy tính sử dụng cổng song song, có thể dùng chế độ mở rộng, chế độ này cho phép giao tiếp với tốc độ cao hơn.



Hình 3.37. Trao đổi dữ liệu qua cổng song song giữa 2PC dùng chế độ mở rộng

Sơ đồ chân kết nối mô tả như sau:

**Bảng 3.14.** Bảng mô tả kết nối 2PC dùng chế độ mở rộng

PC1		PC2	
Chức năng	Chân	Chân	Chức năng
D0	2	2	D0
D1	3	3	D1
D2	4	4	D2
D3	5	5	D3
D4	6	6	D4
D5	7	7	D5
D6	8	8	D6
D7	9	9	D7
SELCT	13	17	$\overline{\text{SELECTIN}}$
BUSY	11	16	$\overline{\text{INIT}}$
$\overline{\text{ACK}}$	10	1	$\overline{\text{STROBE}}$
$\overline{\text{SELECTIN}}$	17	13	SELECT
$\overline{\text{INIT}}$	16	11	BUSY
$\overline{\text{STROBE}}$	1	10	$\overline{\text{ACK}}$

## 3.4. MỘT SỐ THIẾT BỊ NGOẠI VI

### 3.4.1. Màn hình (Monitor)

Màn hình là thiết bị đưa thông tin của máy tính ra ngoài để giao tiếp với người sử dụng. Nó là bộ xuất chuẩn cho máy tính hay còn gọi là bộ trực. Hiện nay có nhiều hãng sản xuất màn hình như Acer, IBM, Funal, Samsung, LG, Hitashi... Nếu phân loại theo tính năng, màn hình bao gồm: Mono, EGA, VGA, SVGA... Màn hình giao tiếp với Mainboard qua bộ điều hợp gọi là Card màn hình, được cắm qua khe PCI, ISA, hoặc EISA.



*Hình 3.38. Màn hình*

Ba vấn đề cần quan tâm trên màn hình là con trỏ màn hình, độ phân giải và màu sắc. Con trỏ màn hình chỉ định vị trí dữ liệu sẽ xuất ra trên màn hình, độ phân giải đặc trưng cho độ mịn màn hình.

Con trỏ là nơi máy tính đưa thông tin tiếp theo ra từ đó, nó được đặc trưng bởi cặp tọa độ (x, y chỉ xét cho độ phân giải chứ không xét theo chế độ văn bản hay chế độ đồ họa) trên màn hình.

- **Độ phân giải:** Màn hình được chia thành nhiều điểm ảnh, số điểm ảnh được tính bằng tích số dòng dọc và dòng ngang trên màn hình. Cặp giá trị dọc và ngang gọi là độ phân giải của màn hình như (480 × 640), (600 × 800), (1024 × 768)...

- **Màu sắc:** Màu của các đối tượng trên màn hình do màu các điểm ảnh tạo nên.



• **Card màn hình:** Điều khiển đến từng điểm ảnh trên màn hình. Nội dung của các điểm ảnh (vị trí và màu sắc) trên màn hình được lưu giữ trên một chip nhớ (RAM CARD) và cứ sau một khoảng thời gian nhỏ hơn 1/24 giây nó sẽ quét toàn bộ màn hình một lần làm cho chúng ta thấy hình ảnh hiển thị liên tục trên màn hình. Khi chip này có dung lượng lớn, nó có khả năng lưu giữ số điểm ảnh nhiều hơn và màu cho mỗi điểm ảnh cũng đa dạng hơn, tạo cho màn hình có độ mịn và độ nét cao. Tuy nhiên, nếu màn hình hoạt động chậm, mà độ phân giải cao và chế độ màu lớn sẽ không đảm bảo chu kỳ 1/24 giây dẫn đến màn hình nhấp nháy.

Hệ điều hành DOS và Windows đều hỗ trợ tính năng Plus and Play (cảm sử dụng) cho màn hình. Tuy nhiên, trong các chế độ đồ họa cao cấp, yêu cầu cần phải có trình điều khiển đúng cho màn hình hiển thị mới đạt được hiệu quả cao. Card màn hình có loại được Windows tự động cài trình điều khiển, có loại phải tự cài lấy. Để cài lại trình điều khiển cho card màn hình ta thực hiện như sau:

*Bước 1:* Khởi động lại Windows, vào Control Panel/System/Device manager.

*Bước 2:* Remove điều khiển card màn hình có sẵn.

*Bước 3:* Chọn Add New Hardware rồi chỉ đường dẫn đến trình điều khiển card màn hình.

*Bước 4:* Chọn Apply để áp dụng trình điều khiển mới và khởi động lại máy tính.

### 3.4.2. Bàn phím (Keyboard)

Bàn phím là một thiết bị đưa thông tin vào trực tiếp, giao diện với người sử dụng. Nó được nối kết với Mainboard thông qua cổng bàn phím (Đặc trưng bởi vùng nhớ I/O và ngắt bàn phím).

Bàn phím được tổ chức như một mạng mạch đan xen nhau, mỗi nút mạng là một phím. Khi nhấn một phím sẽ làm chập mạch điện, tạo ra xung điện tương ứng với phím được ấn gọi là mã quét (*Scan code*). Mã này được đưa vào bộ xử lý bàn phím (8048, 8042), diễn dịch ra ký tự theo một chuẩn nào đó, thông thường là chuẩn ASCII (*American Standard code for Information Interchange*). Sau đó bộ xử lý bàn phím yêu cầu ngắt và gửi vào CPU xử lý. Vì thời gian thực hiện rất nhanh nên ta thấy các phím được xử lý tức thời.



*Hình 3.39. Bàn phím và dạng mạch bàn phím*

Hiện nay trên thị trường có nhiều loại bàn phím, do nhiều hãng sản xuất khác nhau như: Acer, IBM, Turbo Plus, Mitsumi... Tuy nhiên, chúng có chung một số các phím cơ bản từ 101 đến 105 phím được chia làm hai nhóm:

- **Nhóm ký tự:** Khi gõ lên có ký tự xuất hiện trên màn hình.
- **Nhóm điều khiển:** Khi gõ không thấy xuất hiện ký tự trên màn hình, thường dùng để thực hiện một tác vụ nào đó.

Tất cả các phím đều được đặc trưng bởi một mã, một số tổ hợp phím cũng có mã riêng của nó. Điều này giúp cho việc điều khiển bàn phím thuận lợi, nhất là trong công việc lập trình.

### 3.4.3. Chuột (Mouse)

Chuột là thiết bị điều khiển trò trực tiếp phổ biến nhất, đặc biệt là trong lĩnh vực đồ họa. Hiện nay có rất nhiều loại chuột, do nhiều hãng sản xuất khác nhau như: IBM, Acer, Mitsumi, Genius, Logitech... đa số được thiết kế theo hai chuẩn cầm tròn và dẹt. Tuy nhiên, chúng có cấu tạo và chức năng như nhau.



*Hình 3.40. Chuột*

Về cấu tạo, có các loại chuột như chuột cơ học, chuột quang học, chuột cơ quang... Song chỉ có loại chuột cơ học là phổ biến, còn các loại chuột khác được dùng trong các lĩnh vực đặc biệt.

Chuột cơ học có hai bộ phận là bi di chuyển và các nút nhấn.

- **Bi di chuyển:** Gồm một viên bi và hai thanh quay ngang, dọc. Khi di chuyển chuột tương ứng theo các chiều, làm các thanh quay tương ứng tạo ra các xung điện di chuyển vị trí chuột tương ứng trên màn hình.

- **Nút nhấn:** Tạo ra xung chỉ thị sự thực hiện các lệnh tại vị trí chuột trên màn hình.

- Nhấp chuột (*Click*) là động tác nhấn phím trái của chuột, song lại thả nhanh ra (thường dùng để chọn một cái gì đó trên màn hình).

- Nhấp chuột kép (*Double click*) có tác dụng cho chạy một chương trình.

- Nhấp phải (*Right click*) thường được sử dụng trong môi trường Windows 9x, 2000, NT... nhằm mở trình đơn phụ của một đối tượng.

Đối với Windows 95 trở lên chuột có thể cắm và chạy, còn đối với DOS ta phải cài đặt trình điều khiển cho chuột (thường là: mouse.com, gmouse.com) thì nó mới có thể hoạt động được

### 3.4.4. Máy in (Print)

Máy in là thiết bị chủ đạo để xuất dữ liệu máy tính lên giấy. Khi muốn in một file dữ liệu ra giấy thì CPU sẽ gửi toàn bộ dữ liệu ra hàng đợi (queue) máy in và máy in sẽ lần lượt in từ đầu cho đến hết file.



Hình 3.41. Máy in

Máy in hiện nay có rất nhiều loại, với nhiều các thức làm việc khác nhau như máy in kim, máy in phun, máy in lazer... Để đánh giá chất lượng của máy in, ta căn cứ vào hai yếu tố của máy in là tốc độ và độ mịn.

Tốc độ của máy in thường được đo bằng trang/giây (chi tương đối). Tốc độ này nhiều khi còn phụ thuộc vào tốc độ của máy tính và mật độ của trang in chứ không chỉ của máy in.

Đối với máy in kim, tốc độ rất hạn chế: song đến máy in lazer thì tốc độ được cải thiện đi rất nhiều.

Độ mịn (*Dots per inch*) phụ thuộc vào nhiều yếu tố, song yếu tố cơ bản phụ thuộc thông số trực tiếp ghi trên máy in.

Máy in giao tiếp với CPU thông qua các cổng song song LPT1, LPT2, LPT3, LPT4 được gắn qua khe cắm Mainboard.

Hầu hết các hệ điều hành đều hỗ trợ máy in. Đối với DOS thì ta cần phải cài đặt Driver của máy in cho hệ điều hành, thì nó mới làm việc được. Song, đối với các hệ điều hành từ Windows 95 trở lên chế độ Plug and Play hỗ trợ hầu hết các loại máy in hiện nay, do đó ta chỉ cần chọn đúng trình điều khiển mà thôi.

Để thiết lập máy in và in được 1 file, ta làm như sau:

*Bước 1:* Cắm máy in vào máy tính và cấp điện cho máy in.

*Bước 2:* Bật máy tính và cài đặt máy in cho hệ điều hành đang sử dụng.

*Bước 3:* Bật điện cho máy in và cho giấy vào để chuẩn bị sẵn sàng.

*Bước 4:* Chọn 1 file cần in và gõ lệnh in. Trong DOS là lệnh PRN <tên file>, trong Windows mở file cần in sau đó chọn File/Print.

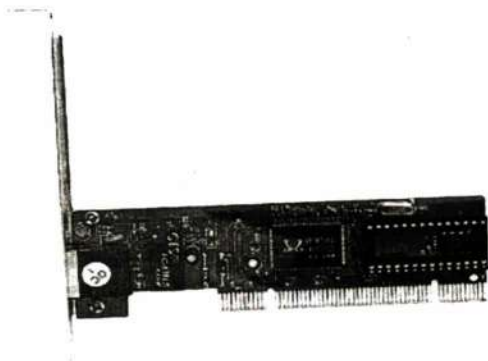
### 3.4.5. Một số thiết bị khác

Ngoài các thiết bị kể trên, còn có rất nhiều thiết bị được cắm vào máy tính để phục vụ cho nhiều mục đích khác nhau như: Card mạng, Modem, Scanner...

#### 3.4.5.1. Card mạng

Card mạng là vi mạch được nối vào máy thông qua BUS PCI hoặc ISA, đầu ra sử dụng các đầu nối để nối dây mạng.





**Hình 3.42.** Card mạng

Card mạng dùng để thiết lập mạng cho mục đích giao tiếp giữa các máy tính với nhau. Để card mạng hoạt động được, ta phải thiết lập đúng trình điều khiển của nó, địa chỉ của các máy tính trên mạng, và cài đúng giao thức (*Protocol*) để giao tiếp.

### 3.4.5.2. Modem

Modem là từ viết tắt của *Modulator - Demodulator*, là thiết bị điều chế – giải điều chế. Modem là thiết bị truyền dữ liệu được dùng để nối các máy tính lại với nhau bằng đường dây viễn thông, với cự ly bất kỳ trên thế giới như mạng Internet.

Tín hiệu xử lý trong máy tính hoặc tín hiệu bắt tay giữa hai máy tính là tín hiệu số (*Digital Signal*), trong khi đường truyền viễn thông chủ yếu phục vụ tín hiệu dạng tương tự (*Analog*). Tín hiệu truyền trên đường dây điện thoại là tín hiệu đã được điều chế biên độ AM (*Amplitude Modulation*), vì vậy Modem có nhiệm vụ chuyển đổi tín hiệu số từ máy tính thành tín hiệu AM và gửi đi.



**Hình 3.43.** Modem



Tại đầu nhận, Modem lại giải điều chế (*Demodulation*) tín hiệu AM, lấy lại tín hiệu số cung cấp cho máy tính. Nhờ có Modem mà hai máy tính ở khoảng cách xa có thể "nói chuyện" được với nhau.

Modem có hai loại: Loại lắp thẳng vào trong máy tính bằng một card riêng, được gọi là Modem trong (*Internal Modem*); loại Modem ngoài (*External Modem*) được nối thông qua công nối tiếp của máy tính như cổng COM1, COM2.

Khi nói đến Modem, người ta thường quan tâm tới tốc độ truyền. Đơn vị là Baud = bit/giây (thường được ký hiệu là bps, Kbps). Tốc độ thường từ 9600 bps đến 33600 bps. Hiện nay đa số Modem có tốc độ 56 Kbps.

### 3.3.5.3. Scanner

Scanner là thiết bị chuyên dùng để quét các hình ảnh và lưu vào máy tính dưới dạng tập tin ảnh.



Hình 3.44. Scanner

## CÂU HỎI VÀ BÀI TẬP

1. Nêu sự khác nhau giữa SRAM và DRAM. Trong máy tính chúng được dùng ở đâu?
2. Trình bày cấu trúc của phần tử nhớ DRAM và SRAM.
3. Nêu mục tiêu của các cấp bộ nhớ.
4. Nêu hai nguyên tắc mà cache dựa vào đó để vận hành.
5. Sự khác biệt giữa cache và bộ nhớ ảo là gì?
6. Trình bày nguyên tắc vận hành của đĩa quang và ưu, khuyết điểm của các loại đĩa quang.
7. Thông thường có bao nhiêu loại bus? Tại sao phải có các chuẩn cho các bus vào/ra?
8. Mô tả các biện pháp an toàn trong việc lưu trữ thông tin trong đĩa cứng.

9. Mô tả vận hành của ổ đĩa cứng. Cách lưu trữ thông tin trong ổ đĩa cứng.
10. Thiết kế mạch để mở rộng bộ nhớ từ  $2K \times 4$  lên  $2K \times 8$ .
11. Cho các chip nhớ SDRAM dung lượng  $4K \times 8$  bit. Hãy thiết kế module nhớ có kích thước  $8K \times 8$  bit.
12. Thiết kế module nhớ SRAM có dung lượng  $64K \times 8$  từ các chip nhớ có dung lượng  $16K \times 4$ .
13. Thiết kế module nhớ EPROM dung lượng  $64K \times 8$  từ các chip nhớ EPROM 2764.
14. Thiết kế module nhớ EPROM dung lượng  $32K \times 8$  từ các chip nhớ EPROM 2764.
15. Làm thế nào để tìm một khối khi nó hiện diện trong cache?
16. Khối nào phải được thay thế trong trường hợp thất bại cache?
17. Việc gì xảy ra khi ghi vào bộ nhớ (chiến thuật ghi)?
18. Mô tả một kỹ thuật đơn giản để thực hiện giải thuật thay thế LRU trong bộ nhớ cây theo phương thức liên kết tập hợp 4 đường (4 – Way set associative).
19. Bộ vi xử lý 80486 có bộ nhớ cache đồng nhất (dùng chung cho cả dữ liệu và lệnh). Cache này dung lượng 8K byte, được ánh xạ theo phương thức kết hợp 4 đường và cache được tổ chức thành 128 tập hợp. Mỗi dòng cache chứa 1 bit thông tin về tính hợp lệ và 3 bit B0, B1, B2 (dùng cho phương pháp thay thế LRU). Khi xảy ra cache miss, bộ vi xử lý đọc một khối 16 byte từ bộ nhớ chính. Vẽ sơ đồ cache và giải thích cách biên dịch địa chỉ của bộ vi xử lý.
20. Cho một bộ nhớ cache với các đặc điểm sau:
  - Phương pháp ánh xạ: 4 đường;
  - Kích thước 1 dòng: 32 bit;
  - Cây có khả năng chứa: 16K byte dữ liệu;
  - Cache dùng cho bộ vi xử lý 16 bit, có 24 bit địa chỉ.
 Hãy thiết kế cấu trúc cache với các thông số trên và giải thích cách biên dịch địa chỉ của bộ vi xử lý.

## Chương 4

# GIỚI THIỆU VỀ MẠNG MÁY TÍNH VÀ MẠNG INTERNET

### MỤC TIÊU

*Sau khi học xong chương này, người học:*

- ✓ Trình bày được lịch sử về mạng máy tính và mạng Internet cũng như các khái niệm về mạng.
- ✓ Nắm bắt được các thành phần của mạng, cấu trúc tổng quát của mạng máy tính, các giao thức và các mô hình dịch vụ.
- ✓ Trình bày được cấu trúc của mạng Internet.
- ✓ Hiểu biết được về lợi ích của mạng.

### NỘI DUNG

- 4.1. Khái niệm về mạng máy tính.
  - 4.2. Các thành phần mạng.
  - 4.3. Giao thức và các mô hình dịch vụ.
  - 4.4. Cấu trúc mạng Internet.
- Câu hỏi và bài tập.

## 4.1. KHÁI NIỆM VỀ MẠNG MÁY TÍNH

### 4.1.1. Lịch sử mạng máy tính và mạng Internet

Vào giữa những năm 50 của thế kỷ XX, khi những thế hệ máy tính đầu tiên được đưa vào hoạt động thực tế với những bóng đèn điện tử, thì chúng có kích thước rất cồng kềnh và tốn nhiều năng lượng. Hồi đó, việc nhập dữ liệu vào các máy tính được thông qua các tấm bia, người viết chương trình đã đục lỗ sẵn. Mỗi tấm bia tương đương với một dòng lệnh, mà mỗi một cột của nó có chứa tất cả các ký tự cần thiết mà người viết chương trình phải

đọc lỗ vào ký tự mình lựa chọn. Các tấm bia được đưa vào một "thiết bị", gọi là thiết bị đọc bia, qua đó các thông tin được đưa vào máy tính hay còn gọi là trung tâm xử lý, sau khi tính toán kết quả sẽ được đưa ra máy in. Như vậy, các thiết bị đọc bia và máy in được thể hiện như các thiết bị vào/ra (I/O) đối với máy tính. Sau một thời gian, các thể hệ máy mới được đưa vào hoạt động, trong đó một máy tính trung tâm có thể được nối với nhiều thiết bị vào/ra (I/O), qua đó nó có thể thực hiện liên tục hết chương trình này đến chương trình khác.

Cùng với sự phát triển của những ứng dụng trên máy tính, các phương pháp nâng cao khả năng giao tiếp với máy tính trung tâm cũng đã được đầu tư, nghiên cứu rất nhiều. Vào giữa những năm 60 của thế kỷ XX, một số nhà chế tạo máy tính đã nghiên cứu thành công những thiết bị truy cập từ xa tới máy tính của họ. Một trong những phương pháp thâm nhập từ xa được thực hiện bằng việc cài đặt một thiết bị đầu - cuối ở một vị trí cách xa trung tâm tính toán, thiết bị đầu - cuối này được liên kết với trung tâm bằng việc sử dụng đường dây điện thoại, và với hai thiết bị xử lý tín hiệu (thường gọi là Modem) gắn ở hai đầu, tín hiệu được truyền thay vì trực tiếp thì thông qua dây điện thoại.

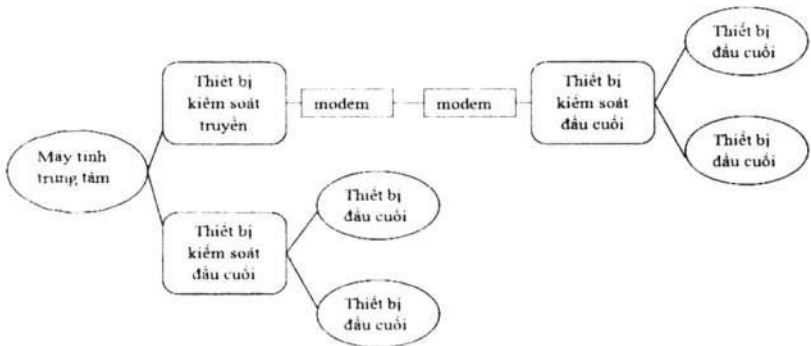
Những dạng đầu tiên của thiết bị đầu - cuối bao gồm máy đọc bia, máy in, thiết bị xử lý tín hiệu, các thiết bị cảm nhận. Việc liên kết từ xa đó có thể thực hiện thông qua những vùng khác nhau, đó là những dạng đầu tiên của hệ thống mạng. Trong lúc đưa ra giới thiệu những thiết bị đầu - cuối từ xa, các nhà khoa học đã triển khai một loạt những thiết bị điều khiển, nhưng thiết bị đầu - cuối đặc biệt cho phép người sử dụng nâng cao được khả năng tương tác với máy tính. Một trong những sản phẩm quan trọng đó là hệ thống thiết bị đầu - cuối 3270 của IBM. Hệ thống đó bao gồm: các màn hình, các hệ thống điều khiển, các thiết bị truyền thông được liên kết với các trung tâm tính toán.

Hệ thống 3270 được giới thiệu vào năm 1971, và được sử dụng dùng để mở rộng khả năng tính toán của trung tâm máy tính tới các vùng xa. Để làm giảm nhiệm vụ truyền thông của máy tính trung tâm và số lượng các liên kết giữa máy tính trung tâm với các thiết bị đầu - cuối, IBM và các công ty máy tính khác đã sản xuất một số các thiết bị sau:



• **Thiết bị kiểm soát truyền thông:** Có nhiệm vụ nhận các bit tín hiệu từ các kênh truyền thông, gom chúng lại thành các byte dữ liệu, sau đó chuyển nhóm các byte đó tới máy tính trung tâm để xử lý; thiết bị này cũng thực hiện công việc ngược lại để chuyển tín hiệu trả lời của máy tính trung tâm tới các trạm ở xa. Thiết bị trên cho phép giảm bớt được thời gian xử lý trên máy tính trung tâm, và xây dựng các thiết bị logic đặc trưng.

• **Thiết bị kiểm soát nhiều đầu - cuối:** Cho phép cùng một lúc kiểm soát nhiều thiết bị đầu - cuối. Máy tính trung tâm chỉ cần liên kết với một thiết bị như vậy là có thể phục vụ cho tất cả các thiết bị đầu - cuối đang được gắn với thiết bị kiểm soát trên. Điều này đặc biệt có ý nghĩa khi thiết bị kiểm soát nằm ở cách xa máy tính, vì chi cần sử dụng một đường điện thoại là có thể phục vụ cho nhiều thiết bị đầu - cuối.



Hình 4.1. Mô hình trao đổi mạng của hệ thống 3270

Vào giữa những năm 1970, các thiết bị đầu - cuối sử dụng những phương pháp liên kết qua đường cáp, nằm trong một khu vực đã được ra đời. Với những ưu điểm từ nâng cao tốc độ truyền dữ liệu, và qua đó kết hợp được khả năng tính toán của các máy tính lại với nhau. Để thực hiện việc nâng cao khả năng tính toán với nhiều máy tính, các nhà sản xuất bắt đầu xây dựng các mạng phức tạp. Vào những năm 1980, các hệ thống đường truyền tốc độ cao đã được thiết lập ở Bắc Mỹ và châu Âu, từ đó cũng xuất hiện các nhà cung cấp các dịch vụ truyền thông với những đường truyền có tốc độ cao hơn nhiều lần so với đường dây điện thoại. Với những



chi phí thuê bao chấp nhận được, người ta có thể sử dụng được các đường truyền này để liên kết máy tính lại với nhau, và bắt đầu hình thành các mạng một cách rộng khắp.

Ở đây các nhà cung cấp dịch vụ đã xây dựng những đường truyền dữ liệu liên kết giữa các thành phố và khu vực với nhau, sau đó cung cấp các dịch vụ truyền dữ liệu cho những người xây dựng mạng. Người xây dựng mạng lúc này không cần xây dựng lại đường truyền của mình, mà chỉ cần sử dụng một phần các năng lực truyền thông của các nhà cung cấp.

Vào năm 1974, Công ty IBM đã giới thiệu một loạt các thiết bị đầu - cuối được chế tạo cho lĩnh vực ngân hàng và thương mại; thông qua các dây cáp mạng, các thiết bị đầu - cuối có thể truy cập cùng một lúc vào một máy tính dùng chung. Với việc liên kết các máy tính nằm ở trong một khu vực nhỏ như một toà nhà, hay là một khu nhà, thì tiền chi phí cho các thiết bị và phần mềm là thấp. Từ đó, việc nghiên cứu khả năng sử dụng chung môi trường truyền thông và các tài nguyên của các máy tính nhanh chóng được đầu tư. Vào năm 1977, Công ty Datapoint Corporation đã bắt đầu bán hệ điều hành mạng của mình là "*Attached Resource Computer Network*" (hay gọi tắt là Arcnet) ra thị trường. Mạng Arcnet cho phép liên kết các máy tính và các trạm đầu - cuối lại bằng dây cáp mạng, qua đó đã trở thành là hệ điều hành mạng cục bộ đầu tiên. Từ đó đến nay đã có rất nhiều công ty đưa ra các sản phẩm của mình, đặc biệt khi các máy tính cá nhân được sử dụng một cách rộng rãi. Khi số lượng máy vi tính trong một văn phòng hay cơ quan được tăng lên nhanh chóng, việc kết nối chúng trở nên vô cùng cần thiết và mang lại nhiều hiệu quả cho người sử dụng.

Ngày nay, với một lượng lớn về thông tin, nhu cầu xử lý thông tin ngày càng cao. Mạng máy tính hiện nay trở nên quá quen thuộc đối với chúng ta, trong mọi lĩnh vực như khoa học, quốc phòng, thương mại, dịch vụ, giáo dục... Hiện nay, ở nhiều nơi mạng đã trở thành một nhu cầu không thể thiếu được. Người ta thấy được, việc kết nối các máy tính thành mạng cho chúng ta những khả năng mới to lớn như:

- **Sử dụng chung tài nguyên:** Những tài nguyên của mạng (như thiết bị, chương trình, dữ liệu) khi được trở thành các tài nguyên chung, thì mọi

thành viên của mạng đều có thể tiếp cận được mà không cần quan tâm tới những tài nguyên đó ở đâu.

- **Tăng độ tin cậy của hệ thống:** Người ta có thể dễ dàng bảo trì máy móc và lưu trữ (*Backup*) các dữ liệu chung, khi có trục trặc trong hệ thống thì chúng có thể được khôi phục nhanh chóng. Trong trường hợp có trục trặc trên một trạm làm việc, người ta cũng có thể sử dụng những trạm khác thay thế.

- **Nâng cao chất lượng và hiệu quả khai thác thông tin:** Khi thông tin có thể được sử dụng chung, nó mang lại cho người sử dụng khả năng tổ chức lại các công việc với những thay đổi về chất như:

- Đáp ứng những nhu cầu của hệ thống ứng dụng kinh doanh hiện đại.
- Cung cấp sự thống nhất giữa các dữ liệu.
- Tăng cường năng lực xử lý nhờ kết hợp các bộ phận phân tán.
- Tăng cường truy nhập tới các dịch vụ mạng khác nhau đang được cung cấp trên thế giới.

Với nhu cầu đòi hỏi ngày càng cao của xã hội, vấn đề kỹ thuật trong mạng là mối quan tâm hàng đầu của các nhà tin học. Ví dụ như, làm thế nào để truy xuất thông tin một cách nhanh chóng và tối ưu nhất, trong khi việc xử lý thông tin trên mạng quá nhiều, đôi khi có thể làm tắc nghẽn mạng và gây ra mất thông tin một cách đáng tiếc.

Hiện nay, việc làm sao có được một hệ thống mạng chạy thật tốt, thật an toàn, với lợi ích kinh tế cao đang rất được quan tâm. Một vấn đề đặt ra, có rất nhiều giải pháp về công nghệ, một giải pháp có rất nhiều yếu tố cấu thành, trong mỗi yếu tố có nhiều cách lựa chọn. Như vậy, để đưa ra một giải pháp hoàn chỉnh, phù hợp, thì phải trải qua một quá trình chọn lọc dựa trên những ưu điểm của từng yếu tố, từng chi tiết rất nhỏ. Để giải quyết một vấn đề phải dựa trên những yêu cầu đặt ra, dựa trên công nghệ để giải quyết. Nhưng, công nghệ cao nhất chưa chắc là công nghệ tốt nhất, mà công nghệ tốt nhất là công nghệ phù hợp nhất.

#### 4.1.2. Các khái niệm về mạng

Với sự phát triển của khoa học và kỹ thuật, hiện nay, các mạng máy tính đã phát triển một cách nhanh chóng, đa dạng cả về quy mô, hệ điều

hành và ứng dụng. Do vậy, việc nghiên cứu chúng ngày càng trở nên phức tạp. Tuy nhiên, các mạng máy tính cũng có cùng các điểm chung, thông qua đó ta có thể đánh giá và phân loại chúng.

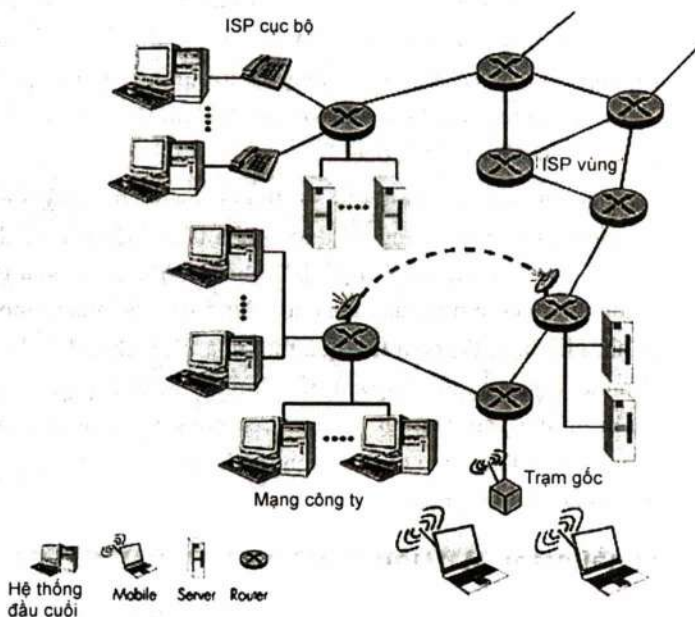
#### 4.1.2.1. Định nghĩa mạng máy tính

Mạng máy tính là một tập hợp các máy tính độc lập, được nối với nhau bởi đường truyền theo một cấu trúc nào đó; và thông qua đó, các máy tính trao đổi thông tin qua lại cho nhau.

Máy tính độc lập được hiểu là các máy tính không có khả năng khởi động hoặc đình chỉ một máy khác.

Các đường truyền vật lý được hiểu là các môi trường truyền tín hiệu vật lý (có thể là hữu tuyến hoặc vô tuyến).

Các quy ước truyền thông chính là cơ sở để các máy tính có thể "nói chuyện" được với nhau, là một yếu tố quan trọng hàng đầu khi nói về công nghệ mạng máy tính.



Hình 4.2. Mô hình mạng máy tính

Với sự trao đổi qua lại giữa máy tính này với máy tính khác đã phân biệt mạng máy tính với các hệ thống thu phát một chiều như truyền hình, phát thông tin từ vệ tinh xuống các trạm thu thụ động... vì tại đây chỉ có thông tin một chiều, từ nơi phát đến nơi thu, mà không quan tâm đến có bao nhiêu nơi thu, có thu tốt hay không.

Đặc trưng cơ bản của đường truyền vật lý là dải thông. Dải thông của một đường truyền chính là độ đo phạm vi tần số mà nó có thể đáp ứng được. 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). Thông lượng còn được đo bằng đơn vị khác là Baud (lấy từ tên nhà bác học – *Emile Baudot*). Baud biểu thị số lượng thay đổi tín hiệu trong một giây. Ở đây Baud và bps không phải bao giờ cũng đồng nhất.

*Vi dụ:* Nếu trên đường dây có 8 mức tín hiệu khác nhau, thì mỗi mức tín hiệu tương ứng với 3 bit hay là 1 Baud, tương ứng với 3 bit. Chỉ khi có 2 mức tín hiệu, trong đó mỗi mức tín hiệu tương ứng với 1 bit thì 1 Baud mới tương ứng với 1 bit.

#### **4.1.2.2. Mục đích của việc kết nối mạng**

Việc nối máy tính thành mạng từ lâu đã trở thành một nhu cầu khách quan, vì:

- Có rất nhiều công việc về bản chất là phân tán hoặc về thông tin, hoặc về xử lý, hoặc cả hai; đòi hỏi có sự kết hợp truyền thông với xử lý, hoặc sử dụng phương tiện từ xa.
- Chia sẻ các tài nguyên trên mạng cho nhiều người sử dụng tại một thời điểm (ổ cứng, máy in, ổ CDROM...).
- Nhu cầu liên lạc, trao đổi thông tin nhờ phương tiện máy tính.
- Các ứng dụng phần mềm đòi hỏi tại một thời điểm cần có nhiều người sử dụng, truy cập vào cùng một cơ sở dữ liệu.

## **4.2. CÁC THÀNH PHẦN MẠNG**

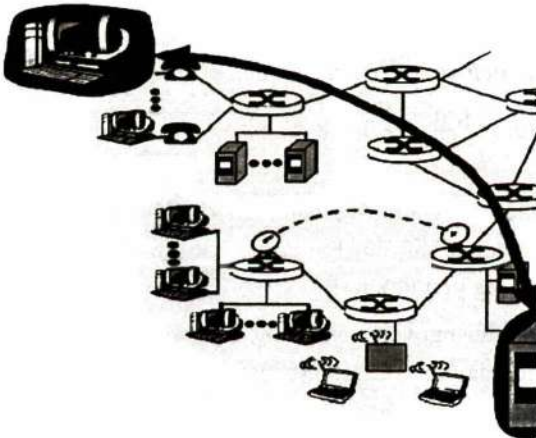
Một hệ thống mạng tổng quát được cấu thành từ ba thành phần:



- **Đường biên mạng** (*Network Edge*): Gồm các máy tính (*Host*) và các chương trình ứng dụng mạng (*Network Application*).
- **Đường trục mạng** (*Network Core*): Gồm các bộ chọn đường (*Router*), đóng vai trò là một mạng trung tâm nối kết các mạng lại với nhau.
- **Mạng truy cập, đường truyền vật lý** (*Access Network, physical media*): Gồm các đường truyền tải thông tin.

#### 4.2.1. Đường biên mạng

Đường biên mạng bao gồm các máy tính trên mạng, nơi thực thi các chương trình ứng dụng mạng (*Network Application*). Đôi khi người ta còn gọi chúng là các hệ thống cuối (*End Systems*), với ý nghĩa đây chính là nơi xuất phát của thông tin di chuyển trên mạng, cũng như là điểm dừng của thông tin. Quá trình trao đổi thông tin giữa hai máy tính trên mạng có thể diễn ra theo hai mô hình: Mô hình khách hàng/người phục vụ (*Client/server model*) hay Mô hình ngang hàng (*Peer-to-peer model*).



Hình 4.3. Đường biên mạng

##### 4.2.1.1. Mô hình khách hàng/người phục vụ (*client/server*)

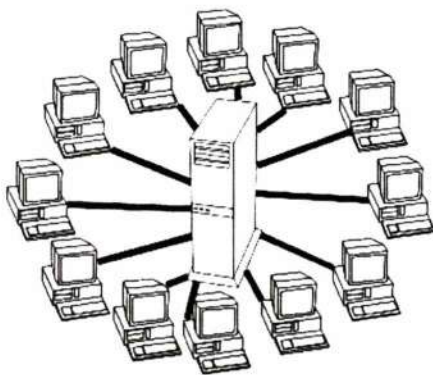
Trong mô hình này, một máy tính sẽ đóng vai trò là client, máy tính kia đóng vai trò là server. Máy tính client sẽ gửi các yêu cầu (*Request*) đến máy tính server để yêu cầu server thực hiện công việc gì đó. Chẳng hạn, khi



người dùng duyệt Web trên mạng Internet, trình duyệt Web sẽ gửi yêu cầu đến Web server đề nghị Web server gửi về trang Web tương ứng.

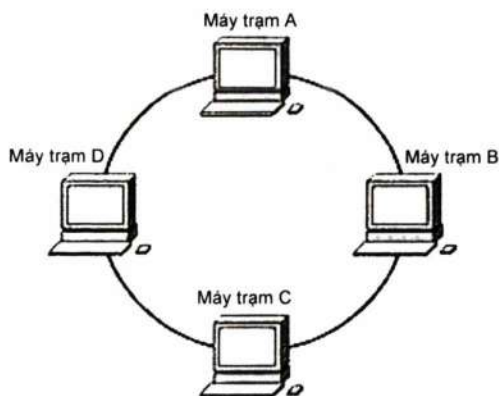
Máy tính server khi nhận được một yêu cầu từ client gửi đến, sẽ phân tích yêu cầu để hiểu được client muốn điều gì để thực hiện đúng yêu cầu của client. Server sẽ gửi kết quả về cho client trong các thông điệp trả lời (*Reply*).

*Ví dụ:* Khi Web server nhận được một yêu cầu gửi đến từ trình duyệt web, nó sẽ phân tích yêu cầu để xác định xem client cần nhận trang Web nào, sau đó mở tập tin html tương ứng trên đĩa cứng cục bộ của nó để gửi về trình duyệt Web trong thông điệp trả lời. Một số ứng dụng được xây dựng theo mô hình client/server như: www, mail, ftp...



**Hình 4.4.** Mô hình *client/server*

#### 4.2.1.2. Mô hình ngang cấp (*peer-to-peer*)



**Hình 4.5.** Mô hình *peer-to-peer*

Mạng ngang hàng (P2P) là mạng mà trong đó hai hay nhiều máy tính chia sẻ tập tin và truy cập các thiết bị như máy in mà không cần đến máy chủ hay phần mềm máy chủ. Ở dạng đơn giản nhất, mạng P2P được tạo ra bởi hai hay nhiều máy tính được kết nối với nhau, và chia sẻ tài nguyên mà không phải thông qua một máy chủ dành riêng.

Mạng P2P có thể là kết nối tại chỗ – hai máy tính nối với nhau qua cổng USB để truyền tập tin. P2P cũng có thể là cơ sở hạ tầng thường trực kết nối 5 – 6 máy tính với nhau trong một văn phòng nhỏ bằng cáp đồng. Hay nó cũng có thể là một dạng quy mô lớn hơn nhiều, dùng các giao thức và ứng dụng đặc biệt để thiết lập những mối quan hệ trực tiếp giữa người dùng trên Internet.

Ứng dụng ban đầu của mạng P2P là sự tiếp nối của việc triển khai các máy tính cá nhân độc lập vào đầu những năm 1980. Khác với máy tính lớn (cỡ mini) ngày ấy, có vai trò máy tính xử lý và lưu trữ trung tâm phục vụ các tác vụ xử lý văn bản và những ứng dụng khác cho các thiết bị đầu cuối (terminal), PC đời mới lúc đó có riêng đĩa cứng và CPU. Nó còn có sẵn các ứng dụng, nghĩa là nó có thể triển khai trên bàn làm việc và thực sự hữu dụng mà không cần phải nối đến máy tính lớn.

Trong mô hình này, một máy tính vừa đóng vai trò là client, vừa đóng vai trò là server. Một số ứng dụng thuộc mô hình này như Gnutella, KaZaA.

Mạng ngang hàng có thể được phân loại theo mục đích sử dụng, ví dụ:

- Chia sẻ file (*File sharing*).
- Điện thoại *VoIP* (*Telephony*).
- Đa phương tiện *media streaming* (*Audio, video*).
- Diễn đàn thảo luận (*Discussion forums*).

Mạng ngang hàng có thể được phân loại theo mức độ tập trung của mạng (đối với P2P overlay networks).

#### 4.2.2. Các đường trục mạng

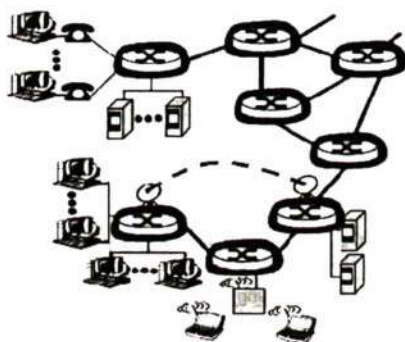
Các đường trục mạng là hệ thống mạng của các bộ chọn đường (*Routers*), làm nhiệm vụ chọn đường và chuyển tiếp thông tin, đảm bảo sự

trao đổi thông tin thông suốt giữa hai máy tính nằm trên hai nhánh mạng cách xa nhau.

Câu hỏi đặt ra là, làm sao thông tin có thể được truyền đi trên mạng? Người ta có thể sử dụng một trong hai chế độ truyền tải thông tin là chuyển mạch kênh (*Circuit switching*) và chuyển gói (*Packet switching*).

#### 4.2.2.1. Chuyển mạch kênh (*Circuit switching*)

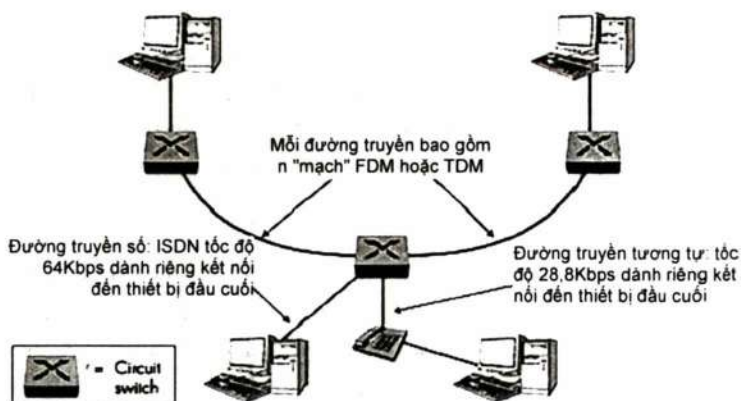
Chế độ này hoạt động theo mô hình của hệ thống điện thoại. Để có thể giao tiếp với máy B, máy A phải thực hiện một cuộc gọi (*Call*). Nếu máy B chấp nhận cuộc gọi, một kênh ao được thiết lập dành riêng cho thông tin trao đổi giữa A và B. Tất cả các tài nguyên được cấp cho cuộc gọi này như băng thông đường truyền, khả năng của các bộ hoán chuyển thông tin đều được dành riêng cho cuộc gọi, không chia sẻ cho các cuộc gọi khác, mặc dù có những khoảng lớn thời gian hai bên giao tiếp "im lặng".



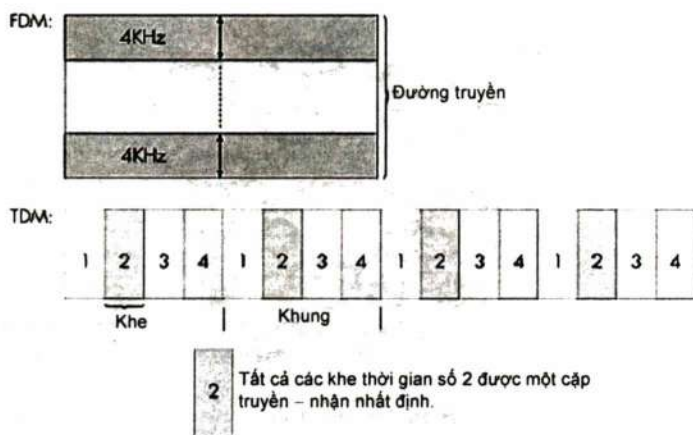
Hình 4.6. Mạng lõi

Tài nguyên (băng thông) sẽ được chia thành những "phần" bằng nhau, và sẽ gán cho các cuộc gọi. Khi cuộc gọi sở hữu một "phần" tài nguyên nào đó, mặc dù không sử dụng đến nó, cũng không chia sẻ tài nguyên này cho các cuộc gọi khác.

Việc phân chia băng thông của kênh truyền thành những "phần" có thể được thực hiện bằng một trong hai kỹ thuật: Phân chia theo tần số (*FDMA – Frequency Division Multi Access*) hay Phân chia theo thời gian (*TDMA – Time Division Multi Access*).



Hình 4.7. Mô hình chuyển mạch kênh bao gồm 3 nút chuyển mạch được nối với nhau bởi 2 đường truyền

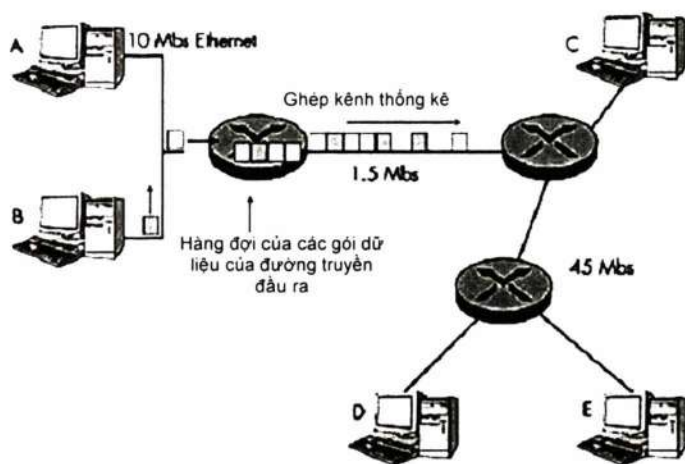


Hình 4.8. Mô hình phân chia theo tần số (FDM) và phân chia theo thời gian (TDM)

#### 4.2.2.2. Chuyển mạch gói (Packet switching)

Trong phương pháp này, thông tin trao đổi giữa hai máy tính (*End systems*) được phân thành những gói tin (*Packet*) có kích thước tối đa, xác định. Gói tin của những người dùng khác nhau (ví dụ của A và B) sẽ chia sẻ nhau băng thông của kênh truyền. Mỗi gói tin sẽ sử dụng toàn bộ băng

thông của kênh truyền khi nó được phép. Điều này sẽ dẫn đến tình trạng lượng thông tin cần truyền đi vượt quá khả năng đáp ứng của kênh truyền. Trong trường hợp này, các router sẽ ứng xử theo giải thuật lưu và chuyển tiếp (*Store and forward*), tức là lưu lại các gói tin chưa gửi đi được vào hàng đợi, cho đến khi kênh truyền rảnh sẽ lần lượt gửi chúng đi.



Hình 4.9. Chuyển mạch gói

Hình trên minh họa cho mạng chuyển mạch gói. Giả sử máy A và máy B đang gửi dữ liệu đến máy E. Máy A và máy B đầu tiên gửi các gói tin của mình trên đường truyền 10 Mbps. Bộ chuyển mạch kênh chuyển hướng các gói tin này vào đường truyền 1.544 Mbps. Nếu xảy ra tắc nghẽn tại nút mạng này, các gói tin sẽ nằm trong các bộ đệm đầu ra của các đường truyền trước khi chúng được truyền đi. Trong trường hợp này, hàng đợi các gói tin từ A và B không theo đúng thứ tự. Thứ tự các gói tin ở đây là ngẫu nhiên, hoặc mang tính chất thống kê, bởi các gói tin được gửi đi ở bất kỳ thời điểm nào. Do những nguyên nhân này, mà chuyển mạch kênh thường sử dụng phương pháp ghép kênh thống kê. Ghép kênh thống kê tương đồng với ghép kênh phân chia theo thời gian, mỗi thiết bị sẽ chiếm một khe thời gian.

Đặc điểm chuyển mạch gói:

- Gói tin của A, B dùng chung tài nguyên mạng.



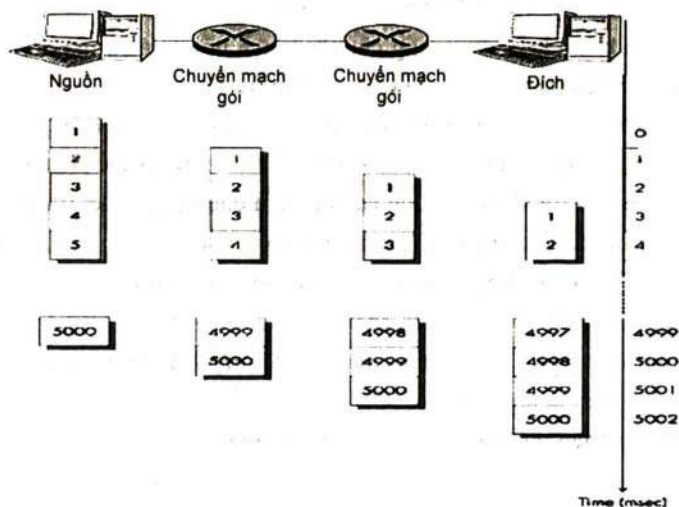
- Mỗi gói tin sử dụng toàn bộ băng thông đường truyền.
- Tài nguyên được sử dụng khi cần thiết.
- Tài nguyên.
- Yêu cầu tài nguyên có thể vượt quá khả năng sẵn có.
- Tắc nghẽn (*Congestion*): Gói tin trong hàng đợi, đợi để được sử dụng kênh truyền.
- Lưu trữ và chuyển tiếp: Các gói tin được chuyển giữa các trạm tại cùng thời điểm.
- Truyền trên đường truyền.
- Đợi trong hàng đợi.

#### 4.2.2.3. So sánh chuyển mạch kênh và chuyển mạch gói

Chuyển mạch gói cho phép có nhiều người sử dụng mạng hơn:

Giả sử:

- Một đường truyền 1 Mbit.
- Mỗi người dùng được cấp 100 Kbps khi truy cập "Active".
- Thời gian active chiếm 10% tổng thời gian.



Hình 4.10. Chuyển mạch gói

Khi đó:

- + Circuit switching: Cho phép tối đa 10 users.
- + Packet switching: Cho phép 35 users (xác suất có hơn 10 active, đồng thời là nhỏ hơn 0,004).
- Chuyển gói:
  - + Thích hợp cho lượng lưu thông dữ liệu lớn nhờ cơ chế chia sẻ tài nguyên và không cần thiết lập cuộc.
  - + Cần có cơ chế điều khiển tắc nghẽn và mất dữ liệu.
  - + Không hỗ trợ được cơ chế chuyển mạch để đảm bảo tăng băng thông cố định cho một số ứng dụng về âm thanh và hình ảnh.

### 4.2.3. Mạng truy cập (Access Network)

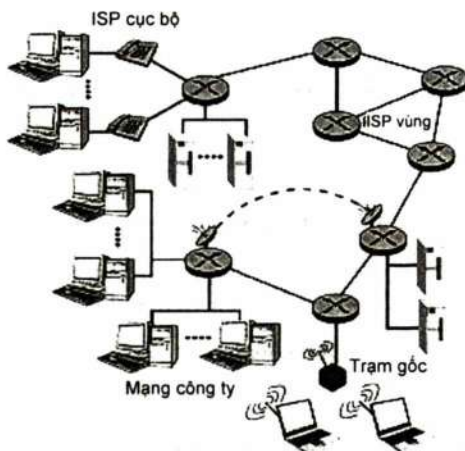
Mạng truy cập có thể phân chia thành ba loại chính sau:

- **Residential access networks:** Mạng truy cập từ hộ gia đình, kết nối hệ thống đầu - cuối vào mạng máy tính.
- **Institutional access networks:** Mạng cục bộ cho các doanh nghiệp kết nối các hệ thống đầu - cuối trong các doanh nghiệp, hoặc các tổ chức giáo dục vào mạng máy tính.
- **Mobile access networks:** Mạng truy cập không dây, kết nối các hệ thống đầu - cuối không dây vào mạng máy tính.

#### 4.2.3.1. Residential access networks

Mạng truy cập từ nhà kết nối một hệ thống đầu - cuối (thường là PC hoặc có thể là Web (Internet – tivi) hoặc hệ thống gia dụng khác) tới rìa mạng.

Hầu hết các kết nối thông thường đều sử dụng Modem thông qua hệ thống điện thoại POTS (*Plain Old Telephone System*) để kết nối đến nhà cung cấp dịch vụ mạng ISP (*Internet Service Provider*). Modem chuyển đổi tín hiệu số đầu ra của PC thành tín hiệu dạng tương tự để truyền trên đường truyền điện thoại. Modem phía nhà cung cấp dịch vụ (ISP) chuyển đổi tín hiệu tương tự trở lại thành tín hiệu số và chuyển tới đầu vào router của ISP. Trong trường hợp này mạng truy cập chỉ là dạng kết nối điểm – điểm (point to point) tới rìa mạng.



**Hình 4.11.** Mô hình Network Access

Ngày nay, tốc độ của Modem cho phép chuyển với tốc độ có thể lên tới 56 Kbps. Tuy nhiên, do chất lượng của cáp truyền khi đi qua nhiều nhà trước khi đến được ISP không được tốt, nên tốc độ mà người dùng có thể đạt được nhỏ hơn 56 Kbps.

Ngược lại, với kỹ thuật quay số, yêu cầu phải chuyển đổi từ tín hiệu số sang tương tự để truyền trên đường truyền thì có một công nghệ mạng băng hẹp là ISDN (*Integrated Services Digital Network*) cho phép truyền tất cả các tín hiệu số trên đường điện thoại. ISDN cho phép tốc độ truyền dữ liệu cao hơn so với phương pháp quay số thông thường (ví dụ 128 Kbps). Phương pháp Modem quay số và ISDN băng hẹp là các công nghệ đã được triển khai rộng khắp.

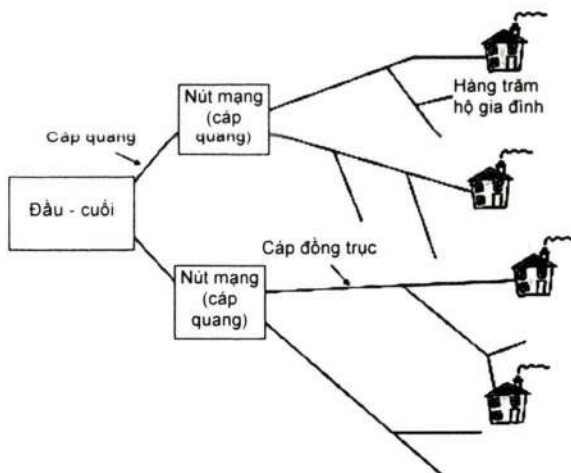
Hai công nghệ mới là ADSL (*Asymmetric Digital Subscribe Line*) và HFC (*Hybrid Fiber Coaxial cable*) cũng đã được triển khai. ADSL cũng tương tự như phương pháp quay số: cùng sử dụng đường dây điện thoại, nhưng cho phép tốc độ truyền cao hơn (có thể lên đến 8 Mbps đường downlink và 1 Mbps đường xuống).

ADSL sử dụng phương pháp phân chia theo tần số. Trong thực tế, ADSL chia băng tần thành hai dải tần số:

- High-speed downstream channel: Tần số 50 KHz – 1MHz.
  - Medium-speed upstream channel: Tần số 4KHz – 50KHz.
- Tần số tín hiệu thoại từ 0 đến 4 KHz.

Một lợi ích của ADSL là cho phép người sử dụng có thể gọi điện thoại trong khi đang sử dụng mạng. Điều này không thể thực hiện được đối với phương pháp quay số thông thường. Lượng băng thông dành cho người dùng đối với cả hai đường lên và xuống phụ thuộc vào khoảng cách giữa Modem ở nhà và Modem của ISP, phụ thuộc và chất lượng của đường dây và mức độ nhiễu điện từ. Đối với các đường dây chất lượng cao, có vỏ tiếp đất để chống nhiễu, thì tốc độ truyền của đường xuống có thể đạt được 8 Mbps nếu khoảng cách là dưới 3 km và tốc độ này giảm xuống 2 Mbps nếu khoảng cách tăng lên 6 km. Đồng thời với đó thì tốc độ đường lên cũng nằm trong khoảng từ 16 Kbps đến 1 Mbps.

Các công nghệ ADSL, ISDN và Modem quay số đều sử dụng đường dây điện thoại, nhưng phương pháp truy cập HFC lại sử dụng cáp truyền hình. Trong hệ thống cáp truyền hình, các trạm cáp đầu - cuối truyền thông tin qua bộ phân phối cáp đồng trục và khuếch đại đến các hộ gia đình. Ví dụ, trong hệ thống cáp quang có thể hỗ trợ kết nối từ 500 đến 5000 hộ gia đình.



Hình 4.12. Mạng truy cập lai cáp quang – cáp đồng trục



Giống như ADSL, HFC cũng yêu cầu sử dụng một loại Modem đặc biệt. Công ty cung cấp dịch vụ truy cập Internet qua hệ thống cáp, yêu cầu khách hàng phải mua hoặc thuê Modem. Ví dụ như Công ty Cyber Cable sử dụng Motorola's Cyber Surfer Cable Modem. Modem này cho phép người dùng truy cập Internet với tốc độ cao. Thông thường, cable Modem là một thiết bị mở rộng và được kết nối với hệ thống PC thông qua cổng Ethernet 10 BaseT. Cable Modem chia mạng HFC thành hai kênh là downstream và upstream (10 Mbps và 768 Kbps).

Một đặc điểm quan trọng của HFC là chia sẻ môi trường quảng bá. Thực tế là, tất cả các gói tin truyền từ trạm đầu - cuối đều được truyền tới tất cả các hộ gia đình và tất cả các gói tin từ một hộ gia đình đều được truyền tới trạm đầu - cuối. Chính vì vậy, tốc độ mỗi người dùng nhận được nhỏ hơn tốc độ của đường downstream. Ngược lại, người dùng có thể lướt Web với tốc độ đầy đủ nếu có số lượng không nhiều các user và các user này ít khi yêu cầu các trang Web trong cùng một thời điểm. Do đường lên là đường dùng chung, nên gói tin được gửi bởi hai người dùng tại cùng thời điểm sẽ gây xung đột, do đó làm giảm băng thông thực tế của mạng.

Những người ủng hộ ADSL cho rằng, ADSL là kết nối điểm - điểm giữa hộ gia đình và ISP, cho nên băng thông của ADSL là dành riêng chứ không phải là dùng chung. Những người ủng hộ HFC thì cho rằng, HFC băng thông mà HFC cung cấp lớn hơn băng thông của ADSL, và cuộc chiến giữa ADSL và HFC có lẽ đã bắt đầu.

#### **4.2.3.2. Mạng truy nhập của các tổ chức**

Trong mạng truy cập cơ quan, mạng LAN được sử dụng để kết nối các hệ thống đầu - cuối tới router của rìa mạng. Các công nghệ mạng LAN sẽ được đề cập đến trong chương sau. Tuy nhiên, công nghệ mạng Ethernet là công nghệ mạng đang được sử dụng chủ yếu. Ethernet hoạt động ở tốc độ 10 Mbps, hoặc 100Mbps, hoặc có thể lên đến 1 Gbps. Giống như HFC, Ethernet cũng sử dụng chung đường truyền, vì vậy người dùng sử dụng chung tốc độ truyền dẫn. Ngoài ra, mạng Ethernet còn sử dụng thiết bị chuyển mạch, cho phép người dùng có thể sử dụng toàn bộ băng thông của mạng.



#### 4.2.3.3. Các mạng truy cập di động

Mạng truy cập không dây sử dụng sóng vô tuyến để kết nối hệ thống đầu - cuối di động (ví dụ như laptop, PDA...) tới trạm gốc (*Base station*). Trong khi đó, trạm gốc lại được kết nối tới rìa mạng. Chuẩn cho mạng không dây là CDPD (*Cellular Digital Packet Data*). CDPD chia nhỏ mạng thành các mạng tế bào, do đó nó sử dụng cùng một dải tần số giống như hệ thống điện thoại tế bào và hoạt động với tốc độ hàng chục Kbps. Các hệ thống đầu - cuối sử dụng chung môi trường truyền với các hệ thống đầu - cuối khác bên trong một cell. Giao thức MAC (*Media Access Control*) được sử dụng để phân xử việc chia sẻ kênh truyền giữa các hệ thống đầu - cuối. Hệ thống CDPD hỗ trợ cho giao thức IP, do đó cho phép các hệ thống đầu - cuối có thể trao đổi các gói tin IP với trạm gốc.

#### 4.2.4. Môi trường truyền

Trong phần trước, đã đưa ra các khái niệm tổng quát về các công nghệ mạng truy cập. Như đã mô tả, các môi trường vật lý cũng được chỉ ra, ví dụ như Modem ADLS hay ISDN sử dụng cáp xoắn, hay như mạng truy cập không dây sử dụng đường truyền là phổ tín hiệu vô tuyến, đó đều là các mô tả về các môi trường truyền dẫn trên mạng Internet. Thêm vào đó, việc định nghĩa môi trường vật lý giúp chúng ta hiểu rõ hơn sự tồn tại của bit.

Xét quá trình di chuyển của một bit từ một hệ thống đầu - cuối này tới hệ thống đầu - cuối khác thông qua một loạt các tuyến đường và các liên kết. Các bit được truyền rất nhiều lần. Đầu tiên hệ thống đầu - cuối phát tín hiệu, ngay sau đó tín hiệu này được truyền đến router thứ nhất, truyền bit đi đến router thứ 2; và tiếp tục truyền như vậy cho đến khi bit thông tin được đưa tới thiết bị đầu - cuối phía thu. Như vậy, các bit của di chuyển từ nguồn tới đích thông qua một loạt các cặp thu phát. Trong mỗi cặp thu phát, thông tin được truyền nhờ sự chuyển động của sóng điện từ, hoặc các xung quang học trên môi trường truyền vật lý. Môi trường vật lý có thể mang nhiều hình dáng và cách thức khác nhau, trong các cặp thiết bị truyền nhận không cần thiết phải sử dụng chung một loại môi trường vật lý.

Ví dụ về các môi trường vật lý bao gồm dây cáp xoắn bằng đồng, cáp đồng trục, phổ tín hiệu sóng vô tuyến mặt đất, phổ tín hiệu sóng vô tuyến vệ

ting. Môi trường vật lý có thể chia thành hai loại: Môi trường truyền hữu tuyến và môi trường truyền vô tuyến. Với môi trường truyền hữu tuyến, các sóng điện từ được định hướng dọc theo môi trường truyền như các loại cáp quang, cáp đồng trục, cáp xoắn. Với môi trường truyền vô tuyến, các sóng điện từ được truyền trong không khí và trong không gian.

Giả thiết rằng, bạn muốn đi dây cho toà nhà và cho phép các máy tính có thể truy cập Internet hoặc Intranet. Bạn sẽ sử dụng cáp xoắn, cáp đồng trục hay là cáp quang? Loại cáp nào cho tốc độ truyền dẫn cao nhất với khoảng cách xa nhất? Chúng ta sẽ đi làm rõ các câu hỏi trong phần tiếp theo. Tuy nhiên, trước khi tìm hiểu các đặc tính của đường truyền hữu tuyến, ta hãy nói một chút về giá thành của chúng. Giá thực tế của các đường truyền (cáp đồng, cáp quang...) tương đối nhỏ hơn so với các thiết bị mạng khác.

Thực tế, việc lắp đặt các đường truyền vật lý có thể có chi phí cao hơn so với giá thành của các đường dây. Lý do đó là, trong các toà nhà thường bố trí các đường cáp xoắn, cáp quang, cáp đồng trục tới tất cả các phòng. Thậm chí, nếu ban đầu chỉ sử dụng một loại đường truyền, thì có thể trong tương lai gần cần phải sử dụng thêm nhiều dạng đường truyền khác nữa. Vì vậy, không thể tiết kiệm được tiền vốn nếu muốn lắp đặt thêm các đường dây.

#### • **Cáp xoắn** (*Twisted – Pair Copper Wire*)

Cáp xoắn là loại dây dẫn ít tốn kém nhất, được sử dụng rộng rãi nhất. Cách đây hàng trăm năm, cáp xoắn đã được sử dụng cho mạng điện thoại. Thực tế là, có đến 99% các đường dây nối từ máy thu phát cầm tay tới các tổng đài sử dụng cáp xoắn. Hầu hết chúng ta đã nhìn thấy cáp xoắn trong gia đình, công sở. Cáp xoắn gồm có hai dây bằng đồng, được bọc cách ly, mỗi dây có đường kính là 1 mm, được sắp xếp dưới dạng xoắn ốc. Các đường dây được xoắn lại với nhau để giảm sự nhiễu giao thoa từ các đường dây bên cạnh. Thông thường, các cặp đường dây được gói lại trong một lớp bảo vệ. Mỗi cặp đường dây tạo thành một đường truyền riêng rẽ.



*Hình 4.13. Cáp xoắn*

Cáp xoắn không có vỏ bảo vệ UTP (*Unshielded twisted pair*), thường được sử dụng trong các toà nhà, nghĩa là thường được dùng trong các mạng nội bộ (LAN). Tốc độ truyền dẫn trong mạng LAN sử dụng cáp xoắn thay đổi từ 10 Mbps tới 1000 Mbps (1Gbps). Tốc độ truyền dẫn phụ thuộc vào đường kính dây dẫn và khoảng cách thu phát. Có hai loại cáp UTP được sử dụng trong mạng LAN: CAT3 và CAT5. CAT3 là loại dùng cho truyền âm thanh. Ngoài ra, CAT3 được dùng trong mạng Ethernet 10 Mbps. CAT5 nhiều nút xoắn trên 1 cm, và có thêm lớp bảo vệ, dẫn tới tốc độ bit cao hơn so với CAT3. Mạng Ethernet 100 Mbps sử dụng cáp UTP CAT5 ngày càng phổ biến.

Cáp quang được đưa ra vào những năm 1980, nhiều người đã chê rằng, cáp xoắn có tốc độ truyền thấp. Nhiều người còn nghĩ rằng, cáp quang sẽ thay thế hoàn toàn cáp xoắn. Nhưng cáp xoắn không hoàn toàn bị từ bỏ một cách dễ dàng. Công nghệ cáp xoắn mới như CAT5 có thể đạt được tốc độ truyền dẫn tới 100 Mbps trong khoảng cách hàng trăm mét. Thậm chí, tốc độ có thể cao hơn trong khoảng cách ngắn hơn. Sau cùng, cáp xoắn nổi lên nhờ công nghệ cho mạng LAN với tốc độ cao.

Cáp xoắn cũng thường được sử dụng trong mạng truy cập của gia đình. Chúng ta có thể thấy các Modem quay số có thể đạt đến tốc độ truy cập là 56 Kbps trên đường truyền sử dụng cáp xoắn. Tương tự như vậy, các công nghệ truy cập khác như ADSL, ISDN cũng sử dụng đường truyền cáp xoắn.

- **Cáp đồng trục** (*Coaxial Cable*)



**Hình 4.14.** Cáp đồng trục

Giống như cáp xoắn, cáp đồng trục cũng sử dụng lõi đồng, tuy nhiên, không giống như cáp xoắn, cáp đồng trục có các lõi đồng đồng tâm với nhau. Với cấu trúc như vậy, và sử dụng thêm lớp cách ly cùng với lớp vỏ, dẫn tới cáp đồng trục có tốc độ bit cao hơn so với cáp xoắn. Cáp đồng trục có hai loại chính: Cáp đồng trục băng cơ bản và cáp đồng trục băng rộng.



Cáp đồng trục băng cơ bản, hay còn gọi là cáp 50 ohm, độ dày 1 cm, nhẹ và dễ dàng bẻ cong. Nó thường được dùng trong mạng LAN, sử dụng cáp nối hình T. Khái niệm băng cơ bản xuất phát từ việc dòng bit được đưa trực tiếp vào cáp, mà không cần phải chuyển tín hiệu tới một tần số khác

Cáp đồng trục băng rộng, hay còn gọi là cáp 75 ohm, có độ dày lớn hơn, nặng hơn và cứng hơn so với cáp băng cơ bản. Nó vẫn được dùng trong một số các thiết bị cũ. Cáp đồng trục băng rộng được sử dụng chủ yếu trong hệ thống cáp truyền hình. Trong cáp băng rộng này, thiết bị phát dịch chuyển tín hiệu số sang một dải tần nhất định, kết quả là có tín hiệu tương tự được truyền từ phía phát đến một hoặc một số thiết bị thu. Cả băng cơ bản và băng rộng đều là các môi trường truyền hữu hướng. Đặc biệt là một số các thiết bị có thể nối cáp trực tiếp và các thiết bị thu có thể thu được bất kỳ tín hiệu nào mà phía phát phát đi.

#### • Cáp quang (*Fiber Optics*)

Cáp quang mỏng, mềm, sử dụng để định hướng cho xung ánh sáng. Một đường cáp quang có thể hỗ trợ cho tốc độ bit rất cao, có thể lên tới 10 Gbps, thậm chí là 100 Gbps. Cáp quang không bị ảnh hưởng của điện từ trường, suy hao tín hiệu rất thấp trên quãng đường 100 km. Chính những đặc điểm này rất thích hợp sử dụng cáp quang cho việc truyền dẫn rất xa trong môi trường hữu tuyến, đặc biệt là các đường truyền dưới biển. Cáp quang được sử dụng trong nhiều hệ thống mạng điện thoại đường dài, đặc biệt là trong các đường trục xương sống của mạng Internet. Tuy nhiên, giá thành của các thiết bị quang như bộ phát, thu, chuyển mạch gây ra những trở ngại cho việc lắp đặt cáp quang cho các mạng truyền dẫn trong phạm vi gần như là các LAN.

#### • Sóng vô tuyến:

Phổ điện từ trường của sóng vô tuyến mang tín hiệu. Đây là phương pháp truyền dẫn rất đặc biệt, bởi nó không cần sử dụng bất kỳ đường dây nào, có thể đi xuyên qua tường, cung cấp kết nối đến người dùng di động và có khả năng truyền tín hiệu với khoảng cách rất xa. Đặc điểm kênh truyền sóng vô tuyến phụ thuộc chủ yếu vào môi trường truyền và khoảng cách tín hiệu truyền. Môi trường truyền quyết định độ suy hao đường truyền và hiệu

ứng fading bóng (giảm độ lớn tín hiệu khi truyền qua quãng đường và đi vòng quanh hoặc xuyên qua vật chắn), fading đa đường (do sự phản xạ của sóng tín hiệu) và sự giao thoa (do ảnh hưởng của các kênh vô tuyến khác hoặc các tín hiệu điện từ trường).

Truyền thông không dây trên trạm mặt đất có thể chia thành hai loại: Một loại hoạt động trong phạm vi hẹp, tương tự như mạng LAN (phạm vi từ 10 tới vài trăm mét); một loại kênh sóng vô tuyến trên diện rộng, sử dụng cho các dịch vụ di động.

Truyền thông vệ tinh liên kết hai hoặc nhiều trạm thu phát trên mặt đất sử dụng sóng cực ngắn (*Viba*). Vệ tinh nhận tín hiệu truyền tới trên một dải tần, tái tạo lại tín hiệu sử dụng bộ lặp và truyền tín hiệu đã được tái tạo đó trên một dải tần khác. Vệ tinh có thể cung cấp băng tần hàng Gbps trong mỗi dải. Có hai loại vệ tinh được sử dụng trong truyền thông là: vệ tinh địa tĩnh và vệ tinh có độ cao thấp.

Vệ tinh địa tĩnh quay xung quanh Trái Đất ở độ cao 36.000 km. Do độ cao lớn như vậy, nên có độ trễ trên đường truyền (250 ms). Tuy nhiên, đường truyền vệ tinh có thể hoạt động ở tốc độ hàng trăm Mbps, thường được sử dụng trong các mạng điện thoại và các mạng đường trục của Internet (*Backbone*).

Vệ tinh quỹ đạo thấp ở gần Trái Đất hơn, không duy trì vị trí cố định so với trạm mặt đất. Chúng quay quanh Trái Đất giống như là Mặt Trăng vậy. Để việc truyền tin được diễn ra liên tục trong một khu vực, cần phải sử dụng nhiều vệ tinh trên quỹ đạo. Có rất nhiều hệ thống truyền thông quỹ đạo thấp được sử dụng. Ví dụ như, hệ thống truyền thông vệ tinh dành cho cá nhân bao gồm 66 vệ tinh trên quỹ đạo.

### 4.3. GIAO THỨC VÀ CÁC MÔ HÌNH DỊCH VỤ

Như đã đề cập, Internet là hệ thống hết sức phức tạp. Chúng ta cũng biết được rằng, Internet cũng bao gồm rất nhiều các thành phần, gồm nhiều các ứng dụng: giao thức, các loại hệ thống đầu - cuối và các kết nối giữa các hệ thống đầu - cuối, các thiết bị định tuyến, các môi trường truyền dẫn. Một câu hỏi đặt ra là, liệu có thể tổ chức thiết lập mạng theo cấu trúc không? Câu trả lời là có.



### 4.3.1. Mô hình phân lớp

Để một mạng máy tính trở thành một môi trường truyền dữ liệu, nó cần phải có những yếu tố sau:

- Mỗi máy tính cần phải có một địa chỉ phân biệt trên mạng.
- Việc chuyển dữ liệu từ máy tính này đến máy tính khác do mạng thực hiện thông qua những quy định thống nhất, gọi là giao thức của mạng.

– Khi các máy tính trao đổi dữ liệu với nhau, thì một quá trình truyền giao dữ liệu đã được thực hiện hoàn chỉnh. Ví dụ như, để thực hiện việc truyền một file giữa một máy tính với một máy tính khác cùng được gắn trên một mạng, các công việc sau đây phải được thực hiện:

- + Máy tính cần truyền cần biết địa chỉ của máy nhận.
- + Máy tính cần truyền phải xác định được máy tính nhận đã sẵn sàng nhận thông tin hay chưa.
- + Chương trình gửi file trên máy truyền cần xác định được rằng, chương trình nhận file trên máy nhận đã sẵn sàng tiếp nhận file.

\* Nếu cấu trúc file trên hai máy không giống nhau, thì một máy phải làm nhiệm vụ chuyển đổi file từ dạng này sang dạng kia.

\* Khi truyền file, máy tính truyền cần thông báo cho mạng biết địa chỉ của máy nhận để các thông tin được mạng đưa tới đích.

Điều trên cho thấy, giữa hai máy tính đã có một sự phối hợp hoạt động ở mức độ cao. Bây giờ, thay vì chúng ta xét cả quá trình trên như là một quá trình chung, ta sẽ chia quá trình trên thành một số công đoạn, mỗi công đoạn con hoạt động một cách độc lập với nhau.

Ở đây chương trình truyền nhận file của mỗi máy tính được chia thành ba module là: Module truyền và nhận file, Module truyền thông và Module tiếp cận mạng. Hai module tương ứng sẽ thực hiện việc trao đổi với nhau trong đó:

- Module truyền và nhận file cần được thực hiện tất cả các nhiệm vụ trong các ứng dụng truyền nhận file. Ví dụ, truyền nhận thông số về file, truyền nhận các mẫu tin của file, thực hiện chuyển đổi file sang các dạng

khác nhau nếu cần. Module truyền và nhận file không cần thiết phải trực tiếp quan tâm tới việc truyền dữ liệu trên mạng như thế nào, mà nhiệm vụ đó được giao cho module truyền thông.

– Module truyền thông quan tâm tới việc các máy tính đang hoạt động và sẵn sàng trao đổi thông tin với nhau. Nó còn kiểm soát các dữ liệu sao cho những dữ liệu này có thể trao đổi một cách chính xác và an toàn giữa hai máy tính. Điều đó có nghĩa là, phải truyền file trên nguyên tắc đảm bảo an toàn cho dữ liệu. Tuy nhiên, ở đây có thể có một vài mức độ an toàn khác nhau được dành cho từng ứng dụng. Ở đây, việc trao đổi dữ liệu giữa hai máy tính không phụ thuộc vào bản chất của mạng đang liên kết chúng. Những yêu cầu liên quan đến mạng đã được thực hiện ở module thứ ba, là module tiếp cận mạng, và nếu mạng thay đổi thì chỉ có module tiếp cận mạng bị ảnh hưởng.

Module tiếp cận mạng được xây dựng liên quan đến các quy cách giao tiếp với mạng, và phụ thuộc vào bản chất của mạng. Nó đảm bảo việc truyền dữ liệu từ máy tính này đến máy tính khác trong mạng. Như vậy, thay vì xét cả quá trình truyền file với nhiều yêu cầu khác nhau như một tiến trình phức tạp, chúng ta có thể xét quá trình đó với nhiều tiến trình con phân biệt dựa trên việc trao đổi giữa các module tương ứng trong chương trình truyền file. Cách này cho phép phân tích kỹ quá trình chuyển file và dễ dàng trong việc viết chương trình.

Việc xét các module một cách độc lập với nhau như vậy cho phép giảm độ phức tạp cho việc thiết kế và cài đặt. Phương pháp này được sử dụng rộng rãi trong việc xây dựng mạng, các chương trình truyền thông và được gọi là phương pháp phân tầng (*Layer*).

Nguyên tắc của phương pháp phân tầng là:

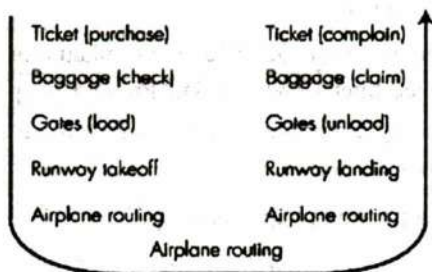
– Mỗi hệ thống thành phần trong mạng được xây dựng như một cấu trúc nhiều tầng và đều có cấu trúc giống nhau như: số lượng tầng, chức năng của mỗi tầng...

– Các tầng nằm chồng lên nhau, dữ liệu chỉ được trao đổi trực tiếp giữa hai tầng kề nhau, từ tầng trên xuống tầng dưới, và ngược lại.

– Cùng với việc xác định chức năng của mỗi tầng, phải xác định mối quan hệ giữa hai tầng kề nhau. Dữ liệu được truyền đi từ tầng cao nhất của hệ thống, truyền lần lượt đến tầng thấp nhất, sau đó truyền qua đường nối vật lý dưới dạng các bit tới tầng thấp nhất của hệ thống nhận, sau đó dữ liệu được truyền ngược lên lần lượt đến tầng cao nhất của hệ thống nhận.

– Chỉ có hai tầng thấp nhất có liên kết vật lý với nhau, còn các tầng trên cùng thứ tư chỉ có các liên kết logic với nhau. Liên kết logic của một tầng được thực hiện thông qua các tầng dưới, và phải tuân theo những quy định chặt chẽ, các quy định đó được gọi giao thức của tầng.

*Ví dụ về mô hình phân lớp:* Xét mô hình vận tải hàng không: Yêu cầu mô tả hệ thống vận chuyển hàng không. Làm thế nào để bạn có thể mô tả được cấu trúc phức tạp của hệ thống này như các hệ thống đại lý bán vé, kiểm soát hành lý, phi công, máy bay, điều khiển không lưu và hệ thống toàn cầu để xác định lộ trình cho các chuyến bay? Có một cách để mô tả hệ thống này thông qua việc phải thực hiện một loạt các thao tác để có thể có mặt trên chuyến bay: mua vé, kiểm tra hành lý, đến cửa và cuối cùng là vào trong máy bay. Máy bay sẽ cất cánh và bay tới đích. Sau khi máy bay hạ cánh, bạn đi ra phía cửa và lấy lại hành lý. Nếu chuyến bay không tốt, bạn có thể đi tới đại lý bán vé để phàn nàn về chuyến bay. Toàn bộ kịch bản này được mô tả như hình dưới đây.

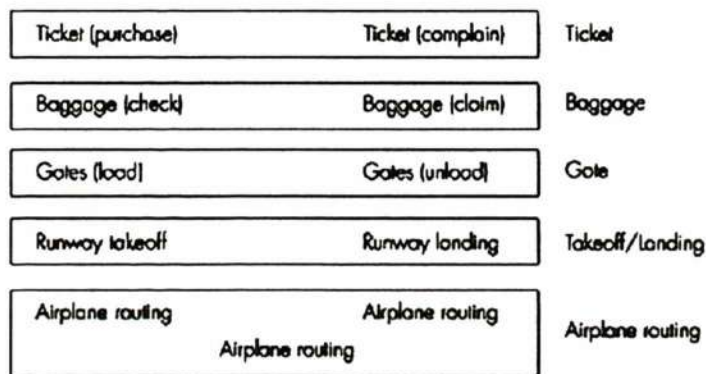


**Hình 4.15.** Các hoạt động của vận tải hàng không

Ta có thể phân chia chức năng vận tải hành khách thành các lớp cung cấp các dịch vụ nền tảng nhất định. Mỗi lớp thực hiện cung cấp một dịch vụ nhất định bằng các hành động bên trong lớp đó, và sử dụng các dịch vụ của



lớp ngay phía dưới. Cấu trúc phân chia theo lớp cho phép mô tả rõ ràng, chi rõ các thành phần của một hệ thống lớn và phức tạp. Ưu điểm của phương pháp này chính là sự đơn giản hoá hoạt động. Khi một hệ thống có cấu trúc phân lớp, nó dễ dàng thay đổi, bổ sung các dịch vụ của các lớp, miễn là các dịch vụ lớp đó cung cấp cho lớp trên nó không đổi và sử dụng cùng các dịch vụ mà lớp dưới cung cấp, đồng thời duy trì các thành phần không đổi của hệ thống khi có sự thay đổi hoạt động của các lớp. Đối với các mạng cực kỳ lớn, thường xuyên cập nhật thông tin, thì khả năng thay đổi các thiết lập của các dịch vụ mà không làm ảnh hưởng đến các thành phần khác của hệ thống là một ưu điểm của phương pháp phân lớp.



*Hình 4.16. Mô hình phân chia theo lớp chức năng*

Để đơn giản hoá, người thiết kế mạng tổ chức các giao thức và các thiết bị phân cứng và phần mềm chạy các giao thức trong các lớp. Với cấu trúc giao thức lớp, mỗi lớp đều có các giao thức riêng của nó. Chú ý là, một giao thức trong lớp thứ  $n$  được phân bố trong tất cả các thực thể tồn tại trong các thành phần mạng (bao gồm hệ thống đầu - cuối, thiết bị chuyển mạch) thực hiện giao thức đó.

Thực tế, trong các lớp trao đổi thông tin theo từng mẫu thông tin một. Các mẫu thông tin này được trao đổi thông qua các thông điệp (*Message*). Các thông điệp này được gọi là đơn vị dữ liệu giao thức lớp  $n$  ( $n$ -PDU). Nội dung của đơn vị dữ liệu giao thức (PDU) của một lớp cũng như cách PDU trao đổi thông tin giữa các thành phần của mạng được định nghĩa bởi các

giao thức. Khi cùng hoạt động, các giao thức của các lớp khác nhau hình thành chồng giao thức (*Protocol stack*).

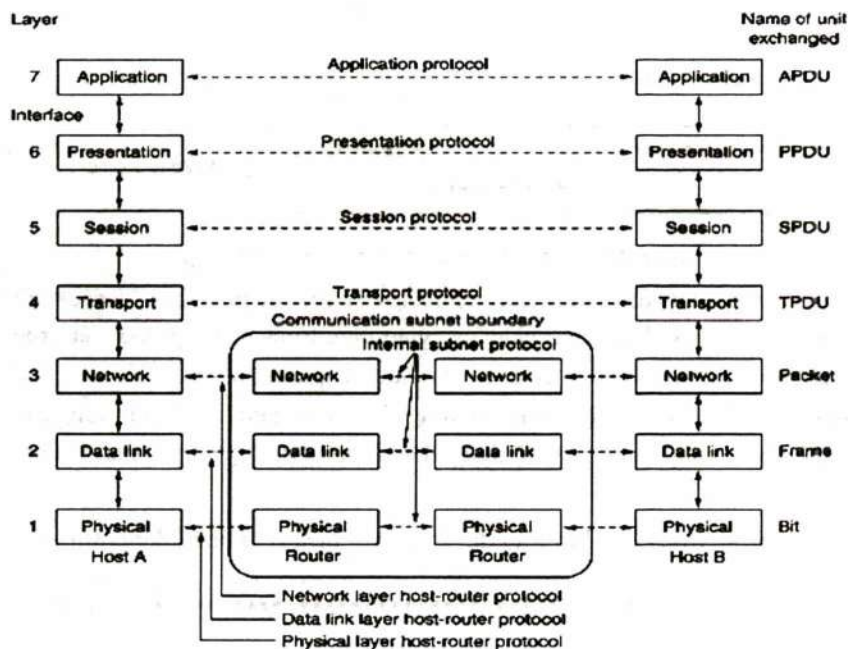
Khi host A gửi một PDU lớp  $n$  tới lớp  $n$  của một host B, khi đó lớp  $n$  của host A chuyển  $n$ -PDU xuống lớp  $n - 1$  và cho phép lớp  $n - 1$  chuyển  $n$ -PDU tới lớp  $n$  của host B. Như vậy, lớp  $n$  sử dụng các dịch vụ lớp  $n - 1$  để chuyển đơn vị dữ liệu giao thức của mình đi tới đích. Một khái niệm quan trọng ở đây đó là mô hình dịch vụ của một lớp (*Service model*).

*Ví dụ:* Lớp  $n - 1$  đảm bảo rằng, PDU của lớp  $n$  đến được đến đích mà không có lỗi hoặc đảm bảo một tỷ lệ lỗi nhất định.

### 4.3.2. Mô hình tham chiếu: OSI, TCP

#### 4.3.2.1. Mô hình tham chiếu OSI (*Open Systems Interconnection*)

Mô hình tham chiếu OSI được mô tả trong hình dưới đây.



Hình 4.17. Mô hình tham chiếu OSI



Mô hình ISO được đưa ra bởi đề xuất của tổ chức tiêu chuẩn quốc tế ISO. Mô hình này được gọi là mô hình tham chiếu OSI – thực hiện kết nối các hệ thống mở với nhau.

Mô hình OSI có 7 lớp (tầng), nguyên lý cơ bản áp dụng cho 7 lớp có thể được mô tả một cách khái quát như sau:

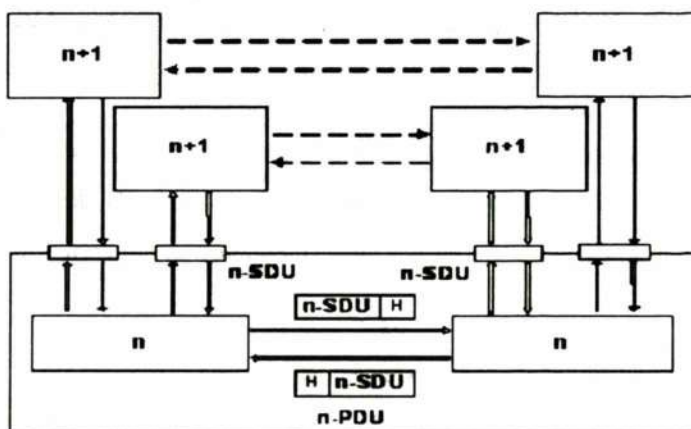
- Mỗi lớp có các khái niệm trừu tượng khác nhau.
- Mỗi lớp thực hiện đúng chức năng xác định.
- Chức năng của mỗi lớp phải phù hợp với các chuẩn giao thức quốc tế.
- Phần giao tiếp giữa các lớp được lựa chọn sao cho thông tin đi qua giao diện là nhỏ nhất.
- Số lượng các lớp đủ lớn để các chức năng khác nhau không cùng nằm trong một lớp và đủ nhỏ để kiến trúc không quá cồng kềnh.

Thành phần chính của các lớp là các giao thức và dịch vụ tương ứng:

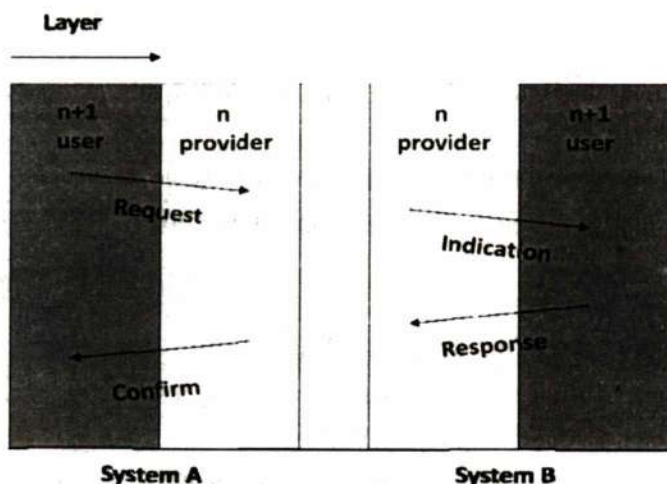
- Giao thức (*Protocol*)
  - + Layer n trong một trạm tương tác với layer n trong một trạm khác để cung cấp dịch vụ cho tầng n + 1.
  - + Các tầng tương ứng giữa hai trạm được gọi là quá trình ngang hàng (*Peer process*).
    - + Giao thức tầng n là tập hợp các quy ước và quy tắc của tầng đó.
    - + Các quá trình ngang hàng của tầng n trao đổi các PDU (*Protocol Data Unit*).
- OSI (*Services*)
  - + Trao đổi giữa các quá trình ngang hàng.
  - + Tầng n + 1 truyền tin dựa vào dịch vụ của tầng n.
  - + Các dịch vụ được cung cấp tại SAP (*Service Access Points*).
  - + Dữ liệu được truyền xuống tầng dưới gọi là SDU (*Service Data Unit*), SDU được đóng gói trong PDU.

Chú ý rằng, mô hình OSI tự bản thân nó không phải là một kiến trúc mạng, bởi nó không xác định một cách cụ thể các dịch vụ và các giao thức sử dụng trong mỗi lớp. Nó chỉ ra rằng, mỗi lớp nên làm gì. Tuy nhiên, mô

hình OSI cũng đưa ra các chuẩn cho tất cả các lớp, mặc dù nó không phải là thành phần trong mô hình tham chiếu. Mỗi chuẩn đưa ra sẽ được công bố như một chuẩn quốc tế riêng rẽ.



Hình 4.18. Mô hình kết nối giữa các lớp trong OSI



Hình 4.19. Tương tác giữa các lớp

#### • Lớp vật lý (Physical)

Lớp vật lý liên quan đến các bit tín hiệu được truyền trên đường truyền. Vấn đề thiết kế đặt ra rằng, phải đảm bảo khi phía phát gửi bit 1 đi, thì phía

nhận cũng nhận được bit 1, chứ không phải là bit 0. Vấn đề cần quan tâm là mức điện áp cho các mức logic 0 và 1 là bao nhiêu, độ rộng của mỗi bit là bao nhiêu ns, khi nào thì đường truyền thực hiện truyền đồng thời trên cả hai hướng; cách thiết lập, khởi tạo kết nối; cách dừng kết nối, có bao nhiêu điểm kết nối trong mạng, chức năng của các điểm này. Yêu cầu thiết kế này liên quan nhiều đến vấn đề cơ học, điện, đồng bộ và môi trường truyền vật lý nằm dưới lớp vật lý này.

Khác với các tầng khác, tầng vật lý không có gói tin riêng, do vậy không có phần đầu (*Header*) chứa thông tin điều khiển, dữ liệu được truyền đi theo dòng bit. Một giao thức tầng vật lý tồn tại giữa các tầng vật lý để quy định về phương thức truyền (đồng bộ, phi đồng bộ), tốc độ truyền.

Các giao thức được xây dựng cho tầng vật lý được phân chia thành hai loại giao thức sử dụng phương thức truyền thông dị bộ (*Asynchronous*) và phương thức truyền thông đồng bộ (*Synchronous*).

– Phương thức truyền dị bộ: Không có một tín hiệu quy định cho sự đồng bộ giữa các bit của giữa máy gửi và máy nhận, trong quá trình gửi tín hiệu máy gửi sử dụng các bit đặc biệt START và STOP, được dùng để tách các xâu bit biểu diễn các ký tự trong dòng dữ liệu cần truyền đi. Nó cho phép một ký tự được truyền đi bất kỳ lúc nào mà không cần quan tâm đến các tín hiệu đồng bộ trước đó.

– Phương thức truyền đồng bộ: Sử dụng phương thức truyền cần có đồng bộ giữa máy gửi và máy nhận, nó chèn các ký tự đặc biệt như SYN (*Synchronization*), EOT (*End Of Transmission*) hay đơn giản hơn, một cái "cờ" (*Flag*) giữa các dữ liệu của máy gửi để báo hiệu cho máy nhận biết được dữ liệu đang đến hoặc đã đến.

#### • Lớp liên kết dữ liệu (*Data link*)

Nhiệm vụ chính của lớp liên kết dữ liệu là vận chuyển các bit dữ liệu vào các đường truyền không có lỗi. Nhiệm vụ này được thực hiện bằng cách chia nhỏ dữ liệu đầu vào thành các khung dữ liệu – Frame (thường là từ vài trăm cho đến vài nghìn byte), và truyền liên tiếp các frame đi. Dịch vụ có thể tin cậy được nếu phía nhận có thông báo xác nhận truyền đúng đối với từng khung dữ liệu (gửi lại khung ack để báo hiệu).

Một vấn đề khác đặt ra đối lớp liên kết dữ liệu đó là, làm thế nào để truyền dữ liệu từ bộ phát có tốc độ truyền cao tới bộ thu có tốc độ chậm hơn. Một vài cơ cấu điều chỉnh được sử dụng để báo cho phía phát biết được kích thước trống của bộ đệm phía thu ở thời điểm hiện tại là bao nhiêu. Thông thường, phần điều khiển luồng này thường đi kèm với việc kiểm soát lỗi. Trong các mạng quảng bá (*Broadcast*) còn có thêm một yêu cầu nữa đối với lớp data link: Làm thế nào để có thể kiểm soát cách truy cập vào kênh truyền dùng chung? Một lớp con của data link là lớp MAC (*Medium Access Control*) được đưa ra để thực hiện chức năng này. Các giao thức lớp liên kết dữ liệu chia làm hai loại chính là các giao thức hướng ký tự và các giao thức hướng bit. Các giao thức hướng ký tự được xây dựng dựa trên các ký tự đặc biệt của một bộ mã chuẩn nào đó (như ASCII hay EBCDIC), trong khi đó các giao thức hướng bit lại dùng các cấu trúc nhị phân (xâu bit) để xây dựng các phần tử của giao thức (đơn vị dữ liệu, các thủ tục); và khi nhận, dữ liệu sẽ được tiếp nhận lần lượt từng bit một.

#### • Lớp mạng (*Network layer*)

Lớp mạng nhắm đến việc kết nối các mạng với nhau, bằng cách tìm đường (*Routing*) cho các gói tin từ mạng này đến mạng khác. Nó xác định việc chuyển hướng, vạch đường các gói tin trong mạng, các gói này có thể phải đi qua nhiều chặng trước khi đến được đích cuối cùng. Nó luôn tìm các tuyến truyền thông không tắc nghẽn để đưa các gói tin đến đích.

Lớp mạng cung cấp các phương tiện để truyền các gói tin qua mạng, thậm chí qua một mạng của mạng (*Network of network*). Bởi vậy, nó cần phải đáp ứng với nhiều kiểu mạng và nhiều kiểu dịch vụ cung cấp bởi các mạng khác nhau. Hai chức năng chủ yếu của lớp mạng là chọn đường (*Routing*) và chuyển tiếp (*Relaying*). Lớp mạng là quan trọng nhất khi liên kết hai loại mạng khác nhau như mạng Ethernet với mạng Token Ring, khi đó phải dùng một bộ tìm đường (quy định bởi lớp mạng) để chuyển các gói tin từ mạng này sang mạng khác và ngược lại. Đối với một mạng chuyển mạch gói (*Packet – switched network*) – gồm tập hợp các nút chuyển mạch gói nối với nhau bởi các liên kết dữ liệu. Các gói dữ liệu được truyền từ một hệ thống mở tới một hệ thống mở khác trên mạng, phải được chuyển qua



một chuỗi các nút. Mỗi nút nhận gói dữ liệu từ một đường vào (*Incoming link*) rồi chuyển tiếp nó tới một đường ra (*Outgoing link*) hướng đến đích của dữ liệu. Như vậy, ở mỗi nút trung gian nó phải thực hiện các chức năng chọn đường và chuyển tiếp.

Việc chọn đường là sự lựa chọn một con đường để truyền một đơn vị dữ liệu (một gói tin chẳng hạn) từ trạm nguồn tới trạm đích của nó. Một kỹ thuật chọn đường phải thực hiện hai chức năng chính sau đây:

– Quyết định chọn đường tối ưu dựa trên các thông tin đã có về mạng tại thời điểm đó thông qua những tiêu chuẩn tối ưu nhất định.

– Cập nhật các thông tin về mạng, tức là thông tin dùng cho việc chọn đường, trên mạng luôn có sự thay đổi thường xuyên nên việc cập nhật là việc cần thiết.

Có hai phương thức đáp ứng cho việc chọn đường là phương thức xử lý tập trung và xử lý tại chỗ:

– Phương thức chọn đường xử lý tập trung được đặc trưng bởi sự tồn tại của một (hoặc vài) trung tâm điều khiển mạng, chúng thực hiện việc lập ra các bảng đường đi tại từng thời điểm cho các nút, sau đó gửi các bảng chọn đường tới từng nút dọc theo con đường đã được chọn đó. Thông tin tổng thể của mạng cần dùng cho việc chọn đường chỉ cần cập nhật và được cất giữ tại trung tâm điều khiển mạng.

– Phương thức chọn đường xử lý tại chỗ được đặc trưng bởi việc chọn đường được thực hiện tại mỗi nút của mạng. Trong từng thời điểm, mỗi nút phải duy trì các thông tin của mạng và tự xây dựng bảng chọn đường cho mình. Như vậy, các thông tin tổng thể của mạng cần dùng cho việc chọn đường cần cập nhật và được cất giữ tại mỗi nút.

Thông thường, các thông tin được đo lường và sử dụng cho việc chọn đường bao gồm:

- Trạng thái của đường truyền.
- Thời gian trễ khi truyền trên mỗi đường dẫn.
- Mức độ lưu thông trên mỗi đường.
- Các tài nguyên khả dụng của mạng.



Khi có sự thay đổi trên mạng (ví dụ thay đổi về cấu trúc của mạng do sự cố tại một vài nút, phục hồi của một nút mạng, nối thêm một nút mới... hoặc thay đổi về mức độ lưu thông), các thông tin trên cần được cập nhật vào các cơ sở dữ liệu về trạng thái của mạng. Hiện nay, khi nhu cầu truyền thông đa phương tiện (tích hợp dữ liệu văn bản, đồ họa, hình ảnh, âm thanh) ngày càng phát triển, đòi hỏi các công nghệ truyền dẫn tốc độ cao, nên việc phát triển các hệ thống chọn đường tốc độ cao đang rất được quan tâm.

#### • Lớp vận chuyển (*Transport layer*)

Lớp vận chuyển cung cấp các chức năng cần thiết giữa lớp mạng và các lớp trên nó, là lớp cao nhất có liên quan đến các giao thức trao đổi dữ liệu giữa các hệ thống mở. Nó cùng các lớp dưới cung cấp cho người sử dụng các phục vụ vận chuyển.

Lớp vận chuyển là lớp cơ sở, ở đó một máy tính của mạng chia sẻ thông tin với một máy khác. Lớp vận chuyển đồng nhất mỗi trạm bằng một địa chỉ duy nhất và quản lý sự kết nối giữa các trạm. Lớp vận chuyển cũng chia các gói tin lớn thành các gói tin nhỏ hơn trước khi gửi đi. Thông thường, lớp vận chuyển đánh số các gói tin và đảm bảo chúng chuyển theo đúng thứ tự. Lớp vận chuyển là lớp cuối cùng chịu trách nhiệm về mức độ an toàn trong truyền dữ liệu, nên giao thức lớp vận chuyển phụ thuộc rất nhiều vào bản chất của lớp mạng.

Người ta chia giao thức lớp mạng thành các loại sau:

– Mạng loại A: Có tỷ suất lỗi và sự cố có báo hiệu chấp nhận được (tức là chất lượng chấp nhận được). Các gói tin được giả thiết là không bị mất. Lớp vận chuyển không cần cung cấp các dịch vụ phục hồi hoặc sắp xếp thứ tự lại.

– Mạng loại B: Có tỷ suất lỗi chấp nhận được, nhưng tỷ suất sự cố có báo hiệu lại không chấp nhận được. Lớp giao vận phải có khả năng phục hồi lại khi xảy ra sự cố.

– Mạng loại C: Có tỷ suất lỗi không chấp nhận được (không tin cậy), hay là giao thức không liên kết. Lớp giao vận phải có khả năng phục hồi lại khi xảy ra lỗi và sắp xếp lại thứ tự các gói tin.

Trên cơ sở loại giao thức lớp mạng, ta có 5 lớp giao thức lớp vận chuyển, đó là:

– Giao thức lớp 0 (*Simple Class – Lớp đơn giản*) cung cấp các khả năng rất đơn giản để thiết lập liên kết, truyền dữ liệu và huỷ bỏ liên kết trên mạng "có liên kết" loại A. Nó có khả năng phát hiện và báo hiệu các lỗi, nhưng không có khả năng phục hồi.

– Giao thức lớp 1 (*Basic Error Recovery Class – Lớp phục hồi lỗi cơ bản*) dùng với các loại mạng B, ở đây các gói tin (TPDU) được đánh số. Ngoài ra, giao thức lớp này còn có khả năng báo nhận cho nơi gửi và truyền dữ liệu khẩn. So với giao thức lớp 0, giao thức lớp 1 có thêm khả năng phục hồi lỗi.

– Giao thức lớp 2 (*Multiplexing Class – Lớp dồn kênh*) là một cải tiến của lớp 0, cho phép dồn một số liên kết chuyển vận vào một liên kết mạng duy nhất, đồng thời có thể kiểm soát luồng dữ liệu để tránh tắc nghẽn. Giao thức lớp 2 không có khả năng phát hiện và phục hồi lỗi. Do vậy, nó cần đặt trên một lớp mạng loại A.

– Giao thức lớp 3 (*Error Recovery and Multiplexing Class – Lớp phục hồi lỗi cơ bản và dồn kênh*) là sự mở rộng giao thức lớp 2, với khả năng phát hiện và phục hồi lỗi, nó cần đặt trên một lớp mạng loại B.

– Giao thức lớp 4 (*Error Detection and Recovery Class – Lớp phát hiện và phục hồi lỗi*) là lớp có hầu hết các chức năng của các lớp trước, ngoài ra còn bổ sung thêm một số khả năng khác để kiểm soát việc truyền dữ liệu.

#### • Lớp phiên (*Session layer*)

Lớp phiên thiết lập "các giao dịch" giữa các trạm trên mạng, nó đặt tên nhất quán cho mọi thành phần muốn đối thoại với nhau, và lập ánh xạ giữa các tên với địa chỉ của chúng. Một phiên làm việc phải được thiết lập trước khi dữ liệu được truyền trên mạng, đảm bảo cho các giao dịch được thiết lập và duy trì theo đúng quy định.

Lớp phiên còn cung cấp cho người sử dụng các chức năng cần thiết để quản trị các giao dịch ứng dụng của họ, cụ thể là:

- Điều phối việc trao đổi dữ liệu giữa các ứng dụng bằng cách thiết lập và giải phóng (một cách logic) các phiên (hay còn gọi là các hội thoại).
- Cung cấp các điểm đồng bộ để kiểm soát việc trao đổi dữ liệu.
- Áp đặt các quy tắc cho các tương tác giữa các ứng dụng của người sử dụng.
- Cung cấp cơ chế "lấy lượt" (nắm quyền) trong quá trình trao đổi dữ liệu.

Trong trường hợp mạng hai chiều luân phiên, thì nảy sinh vấn đề: hai người sử dụng luân phiên phải "lấy lượt" để truyền dữ liệu. Lớp phiên duy trì tương tác luân phiên bằng cách, báo cho mỗi người sử dụng khi đến lượt họ được truyền dữ liệu. Vấn đề đồng bộ hoá trong tầng giao dịch cũng được thực hiện như cơ chế kiểm tra/phục hồi, dịch vụ này cho phép người sử dụng xác định các điểm đồng bộ hoá trong dòng dữ liệu đang chuyển vận, khi cần thiết có thể khôi phục việc hội thoại bắt đầu từ một trong các điểm đó.

Ở một thời điểm chỉ có một người sử dụng có quyền đặc biệt, được gọi các dịch vụ nhất định của tầng giao dịch, việc phân bổ các quyền này thông qua trao đổi thẻ bài (*Token*). Ví dụ: Ai có được token sẽ có quyền truyền dữ liệu, và khi người giữ token trao token cho người khác thì cũng có nghĩa là trao quyền truyền dữ liệu cho người đó.

Các hàm cơ bản của lớp phiên:

- Give Token cho phép người sử dụng chuyển một token cho một người sử dụng khác của một liên kết giao dịch.
- Please Token cho phép một người sử dụng chưa có token có thể yêu cầu token đó.
- Give Control dùng để chuyển tất cả các token từ một người sử dụng sang một người sử dụng khác.

#### • Lớp trình bày (*Presentation*)

Trong giao tiếp giữa các ứng dụng thông qua mạng với cùng một dữ liệu có thể có nhiều cách biểu diễn khác nhau. Thông thường, dạng biểu diễn dùng bởi ứng dụng nguồn và dạng biểu diễn dùng bởi ứng dụng đích có thể khác nhau do các ứng dụng được chạy trên các hệ thống hoàn toàn khác nhau (như hệ máy Intel và hệ máy Motorola). Lớp trình bày phải chịu trách nhiệm chuyển đổi dữ liệu gửi đi trên mạng từ loại biểu diễn này sang loại



khác. Để đạt được điều đó, nó cung cấp một dạng biểu diễn chung, dùng để truyền thông và cho phép chuyển đổi từ dạng biểu diễn cục bộ sang biểu diễn chung và ngược lại. Lớp trình bày cũng có thể dùng kỹ thuật mã hoá để xáo trộn các dữ liệu trước khi được truyền đi, và giải mã ở đầu đến để bảo mật. Ngoài ra, tầng biểu diễn cũng có thể dùng các kỹ thuật nén, sao cho chỉ cần một ít byte dữ liệu để thể hiện thông tin khi nó được truyền ở trên mạng; ở đầu nhận, lớp trình bày bung trở lại để được dữ liệu ban đầu.

#### • Lớp ứng dụng (*Application*)

Lớp ứng dụng là lớp cao nhất của mô hình OSI, nó xác định giao diện giữa người sử dụng và môi trường OSI, và giải quyết các kỹ thuật mà các chương trình ứng dụng dùng để giao tiếp với mạng. Để cung cấp phương tiện truy nhập môi trường OSI cho các tiến trình ứng dụng, người ta thiết lập các thực thể ứng dụng (AE), các thực thể ứng dụng sẽ gọi đến các phần tử dịch vụ ứng dụng (*ASE – Application Service Element*) của chúng. Mỗi thực thể ứng dụng có thể gồm một hoặc nhiều các phần tử dịch vụ ứng dụng. Các phần tử dịch vụ ứng dụng được phối hợp trong môi trường của thực thể ứng dụng thông qua các liên kết (*Association*) gọi là đối tượng liên kết đơn (*SAO – Single Association Object*). SAO điều khiển việc truyền thông trong suốt vòng đời của liên kết đó, cho phép tuần tự hoá các sự kiện đến từ các ASE thành tổ của nó.

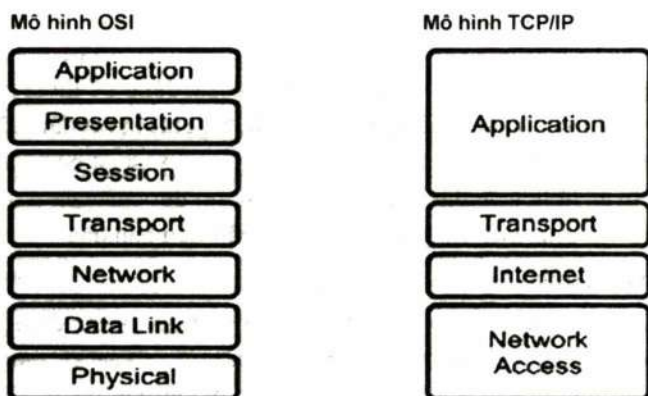
Chức năng:

- Cung cấp các dịch vụ của người sử dụng trực tiếp.
- Cung cấp giao diện tương tác vào các hệ thống đầu - cuối.
- Giao diện lập trình ứng dụng API (*Application Programming Interface*)

*Ví dụ:* WWW, e-mail, FTP, telnet...

#### 4.3.2.2. 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.



Hình 4.20. Tương quan mô hình OSI và mô hình TCP/IP

Vai trò và chức năng các lớp (tầng) trong mô hình TCP/IP:

- **Tầng ứng dụng** (*Process/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 tầng vận chuyển. 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** (*Host to host*)

Ứng với lớp vận chuyển trong mô hình OSI, tầng vận chuyển này thực hiện những kết nối giữa hai máy chủ trên mạng bằng hai 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, tinh 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, độc lập với nhau trên các kết nối khác nhau. TCP cung cấp kết nối song công, dữ liệu có thể được trao đổi trên một kết nối đơn theo hai chiều. Giao thức UDP được sử dụng cho những ứng dụng không đòi hỏi độ tin cậy cao.

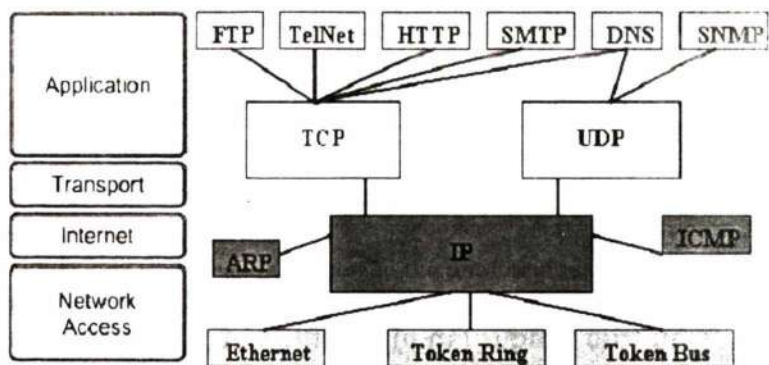


- **Tầng mạng (Internet layer)**

Ứng với lớp mạng (*Network Layer*) trong mô hình OSI, tầng mạng cung cấp một địa chỉ luận lý 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, tầng 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 tầng Network Access Layer cung cấp với địa chỉ luận lý 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 chẩn 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 lớp 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, Tolen Ring, Token Bus... Cung cấp các dịch vụ cho tầng Internet phân đoạn dữ liệu thành các khung.



Hình 4.21. Mô hình kiến trúc TCP/IP

Trong hình trên: – FTP: File Transfer Protocol

- Telnet: TELEcommunication NETwork
- HTTP: HyperText Transfer Protocol
- SMTP: Simple Mail Transfer Protocol
- DNS: Domain Name System
- SNMP: Simple Network Management Protocol
- TCP: Transmission Control
- UDP: Unit Data Protocol
- IP: Internet Protocol
- ARP: Address Resolution Protocol
- RARP: Reverse ARP
- ICMP: Internet Control Message.

## 4.4. CẤU TRÚC MẠNG INTERNET

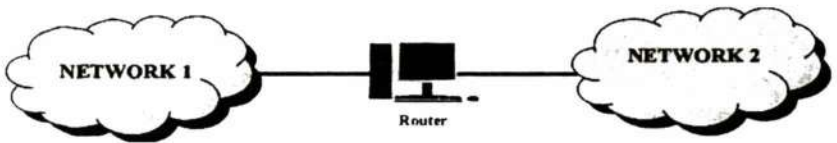
Internet (*Inter-network*) là một mạng máy tính rất rộng lớn, kết nối các mạng máy tính khác nhau nằm rải rộng khắp toàn cầu. Một mạng (*Network*) là một nhóm máy tính kết nối nhau, các mạng này lại liên kết với nhau bằng nhiều loại phương tiện, tốc độ truyền tin khác nhau. Do vậy, có thể nói, Internet là mạng của các mạng máy tính. Các mạng liên kết với nhau dựa trên bộ giao thức (như là ngôn ngữ giao tiếp) TCP/IP (*Transmission Control Protocol – Internet Protocol*): Giao thức điều khiển truyền dẫn – giao thức Internet. Bộ giao thức này cho phép mọi máy tính liên kết, giao tiếp với nhau theo một ngôn ngữ máy tính thống nhất, giống như một ngôn ngữ quốc tế (ví dụ như Tiếng Anh) mà mọi người sử dụng để giao tiếp.

Mạng Internet không chỉ cho phép chuyển tải thông tin nhanh chóng, nó còn giúp cung cấp thông tin. Nó cũng là diễn đàn trao đổi và là thư viện toàn cầu đầu tiên.

### 4.4.1. Cấu trúc tổng quan mạng Internet

Internet là một liên mạng, tức là mạng của các mạng con, để kết nối các mạng con với nhau, có hai vấn đề cần giải quyết. Về mặt vật lý, các mạng

con chỉ có thể kết nối với nhau khi các máy tính có thể kết nối với các mạng này. Việc kết nối đơn thuần về vật lý chưa thể làm cho các mạng con có thể trao đổi thông tin với nhau. Vấn đề thứ hai là, máy tính kết nối được về mặt vật lý với các mạng con phải hiểu được cả hai giao thức truyền tin được sử dụng trên các mạng con này, và các gói thông tin của các mạng con sẽ được gửi qua nhau thông qua đó. Máy tính này được gọi là Internet gateway hay Router.



*Hình 4.22. Hai mạng Network 1 và Network 2 kết nối với nhau thông qua router*

Khi kết nối trở nên phức tạp hơn, các máy Gateway cần phải biết về sơ đồ kiến trúc của các mạng kết nối. Ví dụ, hình sau đây cho thấy nhiều mạng được kết nối bằng 2 router.



*Hình 4.23. Các mạng Network 1, Network 2 và Network 3 kết nối với nhau thông qua router*

Như vậy, Router 1 phải chuyển tất cả các gói thông tin đến một máy nằm ở mạng Network 2 hoặc Network 3. Với kích thước lớn như mạng Internet, việc các router làm sao có thể quyết định về việc chuyển các gói thông tin cho các máy nằm trong các mạng sẽ trở nên phức tạp hơn.

Để các router có thể thực hiện được công việc chuyển một số lớn các gói thông tin thuộc các mạng khác nhau, người ta đề ra quy tắc là: Các router chuyển các gói thông tin dựa trên địa chỉ mạng của nơi đến, chứ không phải dựa trên địa chỉ mạng của máy nhận. Như vậy, dựa trên địa chỉ

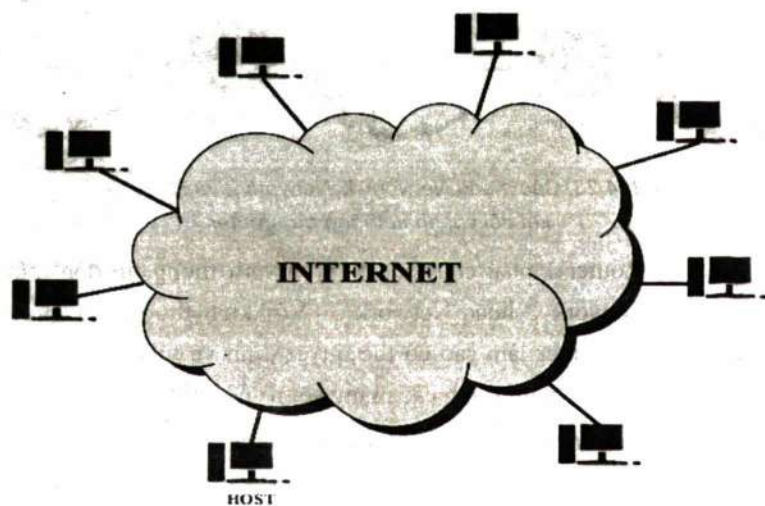
mạng nên tổng số thông tin mà router phải lưu trữ về sơ đồ kiến trúc mạng sẽ tuân theo số mạng trên Internet, chứ không phải là số máy trên Internet.

Trên Internet, tất cả các mạng đều có quyền bình đẳng, cho dù chúng có tổ chức hay số lượng máy rất chênh lệch nhau. Giao thức TCP/IP của Internet hoạt động tuân theo quan điểm: Tất cả các mạng con trong Internet như là Ethernet, một mạng diện rộng như NSFNET back bone, hay một liên kết điểm – điểm giữa hai máy duy nhất đều được coi như là một mạng.

Điều này xuất phát từ quan điểm đầu tiên khi thiết kế giao thức TCP/IP là để có thể liên kết giữa các mạng có kiến trúc hoàn toàn khác nhau, khái niệm "mạng" đối với TCP/IP bị ẩn đi phần kiến trúc vật lý của mạng. Đây chính là điểm giúp cho giao thức TCP/IP tỏ ra rất mạnh.

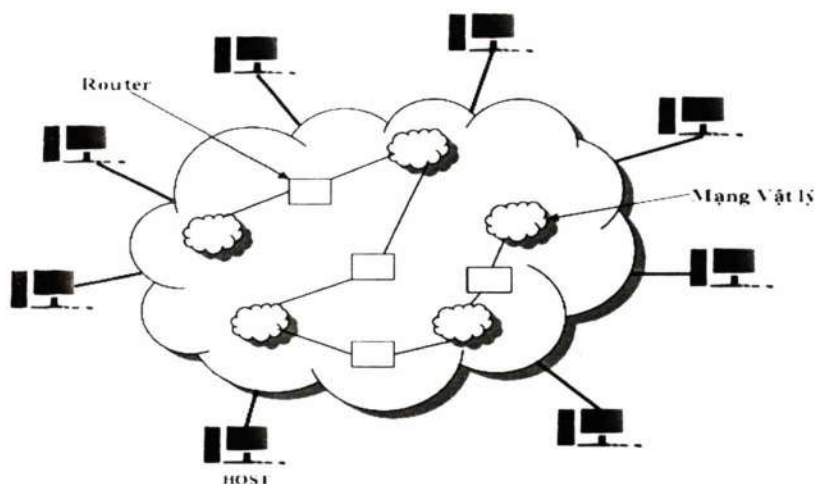
Như vậy, người dùng trong Internet hình dung Internet là một mạng thống nhất, và bất kỳ hai máy nào trên Internet đều được nối với nhau thông qua một mạng duy nhất.

Hình sau mô tả kiến trúc tổng thể của Internet.

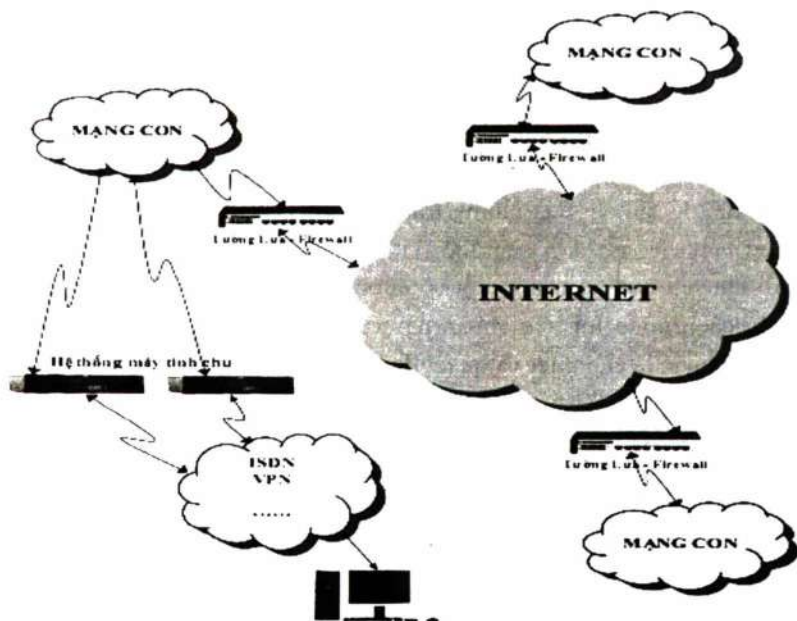


*Hình 4.24. Mạng Internet dưới con mắt người sử dụng  
(Các máy được nối với nhau thông qua một mạng duy nhất)*





Hình 4.25. Kiến trúc tổng quát của mạng Internet  
(Các router cung cấp các kết nối giữa các mạng)



Hình 4.26. Mô hình hệ thống mạng chi tiết



#### 4.4.2. Vấn đề quản lý mạng Internet

Thực chất, Internet không thuộc quyền quản lý của bất kỳ ai. Nó không có giám đốc, không có ban quản trị. Mỗi thành viên đều có thể tham gia hoặc không tham gia vào Internet, đó là quyền của mỗi thành viên. Mỗi mạng thành phần sẽ có một giám đốc hay chủ tịch, một cơ quan chính phủ hoặc một hãng điều hành, nhưng không có một tổ chức nào chịu trách nhiệm về toàn bộ Internet.

Tổ chức Internet "Internet society" – gọi tắt là ISOC, là một tổ chức nguyện có trách nhiệm hoàn toàn về Internet và ở đây là trụ sở chính của Internet, tổ chức này có mục đích phát triển khả năng trao đổi thông tin dựa vào công nghệ Internet. Ý tưởng cơ bản của tổ chức này là khuyến khích sự trao đổi thông tin toàn cầu thông qua Internet. Tổ chức Internet là một uỷ ban với những thành viên tự nguyện – chính những thành viên này là người quyết định hướng tiến lên phía trước của Internet và cũng chính họ là người quản lý kỹ thuật và quy định các chức năng thích hợp của Internet.

Tổ chức ISOC bầu ra Internet Architecture Board (IAB) (Ủy ban Kiến trúc mạng). Ủy ban này có trách nhiệm đưa ra các hướng dẫn về kỹ thuật cũng như phương hướng để phát triển Internet. Nhiệm vụ của IAB là đề ra các nguyên tắc, quy định để tiêu chuẩn hoá và phân chia các nguồn dữ liệu như là: những địa chỉ của những trang Web hoặc vị trí của nguồn dữ liệu. Internet làm việc rất đơn giản, bởi vì những máy vi tính ở trên thế giới có thể trò chuyện qua lại trong một số đường lối chuẩn, và IAB có nhiệm vụ quản lý các đường lối tiêu chuẩn này cũng như IAB sẽ ra quyết định khi thấy tiêu chuẩn là cấp thiết và quyết định ban tiêu chuẩn nên làm gì...

#### CÂU HỎI VÀ BÀI TẬP

1. Mạng máy tính là gì? Nêu các thành phần của mạng.
2. Trình bày các môi trường truyền dẫn trong mạng máy tính.
3. Nêu các lớp trong mô hình phân lớp OSI.
4. Trình bày lớp ứng dụng trong mô hình OSI.
5. Trình bày lớp truyền tải trong mô hình OSI.

6. Trình bày lớp liên kết dữ liệu trong mô hình OSI.
7. Trình bày lớp mạng trong mô hình OSI.
8. Nêu các tầng tầng trong mô hình TCP/IP.
9. So sánh hai mô hình phân lớp OSI và TCP/IP.
10. Nêu cấu trúc của mạng Internet.
11. Cho biết khoảng thời gian cần thiết để truyền file có dung lượng 640 Kbit từ A đến B trên mạng chuyển mạch gói. Giả sử đường truyền trong mạng đều sử dụng phương pháp ghép kênh TDM với 24 khe thời gian và ta có tốc độ bits là 1.536Mbps. Cũng giả thiết là thời gian để thiết lập đường truyền trước khi A có thể truyền tới B phải mất 500 ms. Hỏi thời gian truyền file trên là bao nhiêu?

# Chương 5

## CÁC ỨNG DỤNG MẠNG

### MỤC TIÊU

Sau khi học xong chương này, người học:

- ✓ Trình bày được giao thức hoạt động của lớp ứng dụng.
- ✓ Sử dụng được các chương trình ứng dụng:
  - Dịch vụ DNS: dịch vụ tên;
  - Dịch vụ e-mail;
  - World wide web: HTTP;
  - File transfer: FTP.
- ✓ Phân biệt được các dịch vụ trong chương trình ứng dụng.
- ✓ Phát huy khả năng tư duy sáng tạo.

### NỘI DUNG

5.1. Nguyên lý hoạt động của các giao thức lớp ứng dụng.

5.2. Các chương trình ứng dụng: World Wide Web: HTTP, File transfer: FTP, DNS, E.Mail.

Câu hỏi và bài tập.

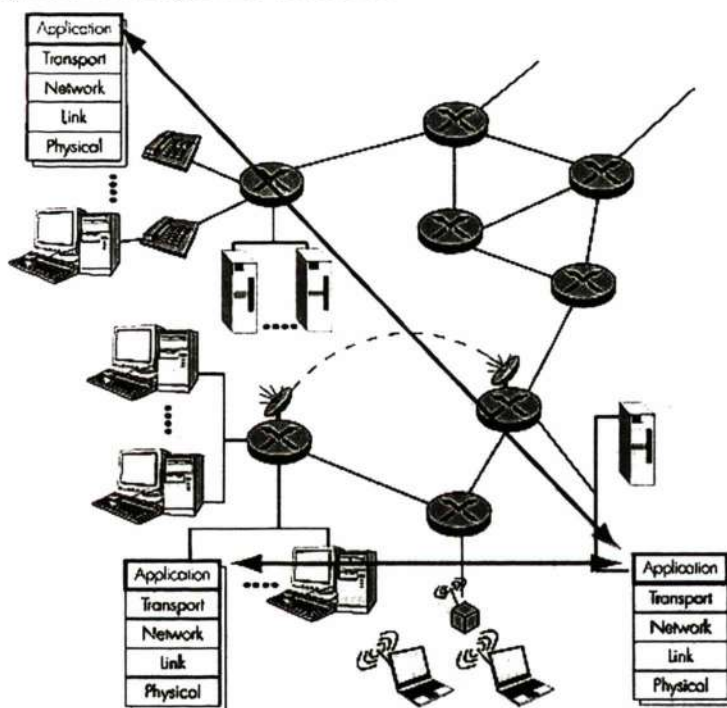
## 5.1. NGUYÊN LÝ HOẠT ĐỘNG CỦA CÁC GIAO THỨC LỚP ỨNG DỤNG

### 5.1.1. Các khái niệm

Nếu không có một chương trình hữu ích nào, thì chúng ta không cần phải thiết kế các giao thức để hỗ trợ cho các chương trình. Tuy nhiên, trong vòng 30 năm trở lại đây, nhiều người đã nghĩ ra các chương trình ứng dụng mạng tuyệt vời. Các chương trình ứng dụng bao gồm những ứng dụng dưới dạng text, và trở lên nổi tiếng vào những năm 1980 như: truy nhập máy tính

từ xa, e-mail, truyền file, chat. Gần đây các ứng dụng mạng còn chứa cả các ứng dụng đa phương tiện như WWW, VoIP, video conferencing, audio và video theo yêu cầu. Mặc dù các ứng dụng mạng gồm nhiều loại và có nhiều thành phần tương tác, các phần mềm luôn đóng vai trò hạt nhân trung tâm. Các phần mềm ứng dụng thường là phân tán giữa hai hoặc nhiều các hệ thống đầu - cuối.

*Ví dụ:* Trong ứng dụng Web, có hai loại phần mềm giao tiếp với nhau: phần mềm trình duyệt chạy bởi người sử dụng đầu - cuối (PC, Mac...) và chương trình trình duyệt Web trên server.



**Hình 5.1.** Giao tiếp ứng dụng

Trong thuật ngữ của hệ điều hành, thực tế không sử dụng khái niệm các mẫu phần mềm mà sử dụng thuật ngữ "Processes" để truyền thông được với nhau. Một quá trình có thể hoạt động như một chương trình chạy bên trong hệ thống đầu - cuối. Khi một quá trình truyền thông xảy ra trên một hệ

thống đầu - cuối, chúng sẽ giao tiếp với các thành phần khác có tham gia vào quá trình truyền thông. Quá trình trên hai hệ thống đầu - cuối khác nhau được thực hiện bằng cách trao đổi thông điệp trên mạng máy tính. Quá trình gửi tạo ra và gửi đi thông điệp vào trong mạng. Quá trình nhận nhận thông điệp và có thể phản ứng lại bằng một thông điệp phản hồi. Các ứng dụng mạng có nhiều giao thức lớp ứng dụng, định nghĩa khuôn dạng và thứ tự các thông điệp được trao đổi giữa các quá trình – giống như việc định nghĩa các hành động truyền và nhận thông tin (hình 5.1).

Lớp ứng dụng là nơi thích hợp nhất để ta bắt đầu nghiên cứu các giao thức, ở đây có thể làm quen với nhiều ứng dụng chạy trên các giao thức nghiên cứu. Nó cho cảm giác tốt về các giao thức mà chúng ta sẽ đề cập đến trong các lớp: truyền dẫn, mạng, liên kết dữ liệu.

### 5.1.2. Giao thức lớp ứng dụng

Điểm quan trọng là phải phân biệt được sự khác biệt của các ứng dụng mạng và các giao thức lớp ứng dụng. Một giao thức lớp ứng dụng chỉ là một phần của chương trình ứng dụng mạng.

*Ví dụ:* Ứng dụng Web bao gồm nhiều thành phần như: các tài liệu chuẩn (html), trình duyệt Web (IE, Netscape, Firefox), Web server và các giao thức lớp ứng dụng. Giao thức sử dụng ở đây là HTTP (*Hyper Text Transfer Protocol*), định nghĩa các thông tin được truyền giữa trình duyệt Web và Web server.

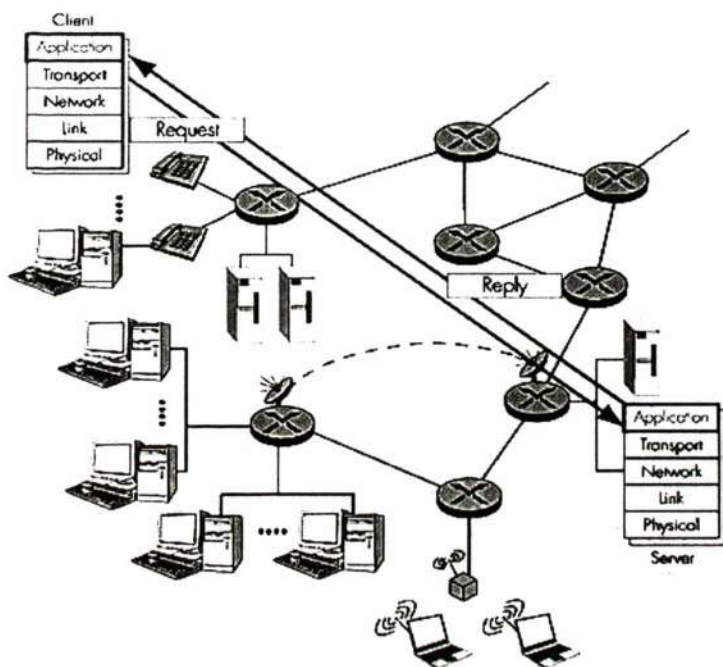
Do đó, HTTP chỉ là một phần của ứng dụng Web. Một ví dụ khác đó là chương trình e-mail – thư điện tử. Thư điện tử bao gồm nhiều thành phần như: mail server dùng để chứa hộp thư, chương trình đọc mail cho phép người dùng đọc và tạo thư mới, một chuẩn định nghĩa cấu trúc một bức thư và các giao thức lớp ứng dụng định nghĩa cách thức truyền thư giữa các server, cách thức truyền thư giữa mail server và người dùng. Giao thức cơ bản lớp ứng dụng của thư điện tử là SMTP (*Simple Mail Transfer Protocol [RFC 821]*). Do đó SMTP chỉ là một phần của ứng dụng e-mail.

Giao thức lớp ứng dụng được mô tả trong các tiêu chuẩn này mô tả cấu trúc cú pháp của các trường thông tin cũng như ý nghĩa, cách thức thực hiện của các trường thông tin tương ứng.



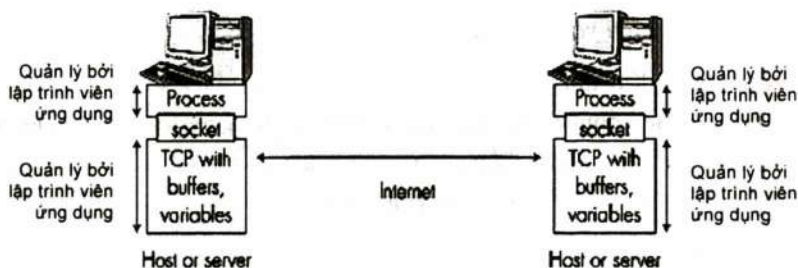
### 5.1.2.1. Khách và chủ (Clients and servers)

Giao thức lớp ứng dụng thường bao gồm hai phần: client (khách) và server (chủ). Bên phía client trên một hệ thống đầu - cuối thực hiện giao tiếp với bên server trên một hệ thống đầu - cuối khác. Trong nhiều ứng dụng, các máy chủ đóng vai trò vừa là client, vừa là server.



Hình 5.2. Quá trình truyền thông trên mạng

Như đã biết, một ứng dụng bao gồm hai quá trình trên hai thiết bị muốn trao đổi thông tin với nhau. Hai quá trình truyền thông đó bao gồm việc truyền và nhận thông tin thông qua các socket (đế) của chúng. Một tiến trình trên socket được thực hiện trên các cổng: thông tin được gửi và nhận qua cổng đó. Khi một tiến trình muốn gửi thông tin tới một tiến trình khác, hoặc máy chủ khác, thì nó sẽ đẩy thông tin ra cổng đó. Giả thiết cần chuyển một lưu lượng từ socket này tới một socket khác trên thiết bị đích. Hình 5.3 mô phỏng truyền thông trên socket giữa hai quá trình trên mạng Internet.



**Hình 5.3.** Các tiến trình ứng dụng, socket và giao thức truyền dẫn của lớp bên dưới

Trong hình trên, socket đóng vai trò như một giao diện giữa lớp ứng dụng và lớp truyền dẫn trong thiết bị chủ. Giống như giao diện cho người lập trình ứng dụng (API) giữa ứng dụng và mạng, các socket là các giao diện lập trình mà người lập trình mạng có thể tạo ra các ứng dụng sử dụng trên mạng Internet.

Các nhà phát triển ứng dụng có thể kiểm soát mọi thứ bên phía lớp ứng dụng của socket, nhưng chỉ kiểm soát một phần bên phía lớp truyền dẫn của socket. Người lập trình chỉ có thể kiểm soát được cách lựa chọn giao thức truyền dẫn và có thể chỉ là độ rộng tối đa của bộ đệm hoặc kích thước tối đa của các phân đoạn. Khi một nhà phát triển ứng dụng lựa chọn một giao thức truyền dẫn thì ứng dụng đó sẽ sử dụng dịch vụ cung cấp của giao thức lớp truyền dẫn đó.

#### 5.1.2.2. Quá trình đánh địa chỉ (Addressing Processes)

Quá trình một thiết bị chủ gửi thông tin đến một thiết bị chủ khác là phải xác định được quá trình nhận. Để xác định quá trình nhận phải chỉ rõ hai loại thông tin: tên hoặc địa chỉ của máy; định danh được quá trình nhận trên thiết bị đích.

Để hiểu rõ, ta xét địa chỉ các thiết bị chủ trên mạng. Trong các ứng dụng, thiết bị đích được xác định bởi các địa chỉ IP. Ta biết rằng, địa chỉ IP gồm 32 bit duy nhất một hệ thống đầu - cuối. Việc đánh địa chỉ cần thực hiện cẩn thận, tránh lặp lại. Thêm vào đó, việc biết được địa chỉ của hệ thống đầu - cuối mà thông tin được chuyển tới, xác định thông tin cho phép

thiết bị nhận chuyên hướng thông tin đến quá trình tương ứng trên hệ thống đầu - cuối đó. Số hiệu cổng bên nhận đảm nhiệm vai trò đó trong mạng Internet. Các giao thức lớp ứng dụng thường gắn với một số hiệu cổng nhất định.

*Ví dụ:* Một tiến trình Web (sử dụng giao thức HTTP) sử dụng cổng 80. Chương trình mail server (SMTP) sử dụng cổng (Port) 25. Danh sách các cổng thông dụng cho tất cả các giao thức chuẩn trên Internet có thể được tìm thấy trong RFC 1700. Khi một nhà phát triển mạng tạo ra một chương trình ứng dụng mạng thì chương trình đó phải gắn với một số hiệu cổng nhất định.

### 5.1.2.3. Tác nhân người dùng (User Agents)

Trước khi nghiên cứu chi tiết về các giao thức lớp ứng dụng, ta tìm hiểu về khái niệm tác nhân người dùng. Tác nhân người dùng là một giao diện giữa người dùng và chương trình ứng dụng mạng. Ví dụ đối với Web, tác nhân người dùng là trình duyệt Web như IE, Netscape. Các trình duyệt cho phép người dùng xem nội dung trang Web, điều hướng trong Web, đưa ra các mẫu nhập liệu... Trình duyệt có thể thực hiện ở phía client và sử dụng giao thức HTTP. Do đó, khi quá trình được thực hiện cùng với một giao diện người dùng, thông tin được truyền và nhận thông qua các socket. Một chương trình khác đó là chương trình ứng dụng mail, tác nhân người dùng là chương trình đọc mail, cho phép người dùng soạn thư và đọc thư. Rất nhiều các công ty đưa ra các chương trình đọc mail (ví dụ: Outlook) với giao diện đồ họa chạy trên các hệ máy PC, Mac và các máy trạm.

### 5.1.3. Dịch vụ lớp ứng dụng

Các socket là các giao diện giữa ứng dụng và giao thức truyền dẫn. Một ứng dụng phía phát gửi thông tin đi qua cổng. Ở cổng phía thu, giao thức truyền dẫn chịu trách nhiệm chuyên thông tin trên mạng đến cổng của quá trình nhận. Mạng Internet có nhiều giao thức truyền dẫn, khi bạn phát triển một ứng dụng, bạn phải chọn một giao thức truyền dẫn nhất định. Làm cách nào có thể lựa chọn? Hầu hết bạn phải tìm hiểu các dịch vụ được giao thức truyền dẫn đó cung cấp, và bạn chọn một giao thức có các dịch vụ cung cấp



phù hợp với yêu cầu của ứng dụng mà bạn cần. Các dịch vụ chương trình ứng dụng mạng cần từ phía giao thức truyền dẫn có thể phân loại một cách khái quát các theo yêu cầu dịch vụ mà ứng dụng cần thành ba loại: suy hao (mất) dữ liệu, băng thông, thời gian.

### **5.1.3.1. Suy hao dữ liệu (Data Loss)**

Một vài ứng dụng như: e-mail, truyền file, truy cập máy chủ từ xa, truyền dữ liệu trên Web và các ứng dụng về tài chính yêu cầu dữ liệu truyền hoàn toàn tin cậy, không mất dữ liệu. Thực tế, việc mất dữ liệu sẽ làm hỏng kết quả. Các ứng dụng khác có thể chấp nhận sự mất dữ liệu, đặc biệt là các ứng dụng đa phương tiện như các dịch vụ về audio/video thời gian thực có thể chấp nhận một lượng dữ liệu suy hao nhất định. Ảnh hưởng của việc suy hao dữ liệu đến chất lượng của ứng dụng và số lượng gói tin thực tế chấp nhận được phụ thuộc vào ứng dụng và cách mã hoá được sử dụng.

### **5.1.3.2. Băng thông (Bandwidth)**

Một vài ứng dụng phải được truyền ở một tốc độ bit nhất định để đảm bảo đạt được các hiệu ứng nhất định. Ví dụ, nếu một ứng dụng thoại mã hoá tiếng nói ở tốc độ 32 Kbps, khi đó dữ liệu cần phải được truyền vào trong mạng và đưa đến phía thu với tốc độ như vậy. Nếu số lượng băng thông không đạt được, thì chương trình phải mã hoá ở tốc độ khác, hoặc không làm việc được. Nhiều ứng dụng đa phương tiện yêu cầu băng thông cần thiết, tuy nhiên cũng có nhiều chương trình ứng dụng kỹ thuật điều biến mã hoá để mã hoá ở tốc độ phù hợp với băng thông hiện có. E-mail, file transfer đều là các chương trình sử dụng băng thông một cách mềm dẻo. Tất nhiên, nếu có nhiều băng thông thì càng tốt.

### **5.1.3.3. Thời gian truyền (Timing)**

Dịch vụ yêu cầu cuối cùng đó là sự tính toán thời gian. Các ứng dụng tương tác thời gian thực như điện thoại Internet, ứng dụng môi trường ảo, hội nghị từ xa, chơi game trực tuyến, nhiều người dùng yêu cầu rất cao về khoảng thời gian phân phát dữ liệu để đạt được hiệu ứng cần thiết. Các ứng dụng trên đòi hỏi thời gian trễ đầu - cuối nhỏ hơn vài trăm ms. Thời gian trễ

quá dài làm cho xuất hiện hiện tượng tín hiệu bị ngừng, hoặc có thể làm cho chương trình chạy không được giống thực tế. Trong các hệ thống thời gian thực, thời gian trễ nhỏ hơn thích hợp hơn đối với trường hợp độ trễ cao, tuy nhiên không có ràng buộc nào về độ trễ trên hệ thống đầu - cuối.

**Bảng 5.1.** Các yêu cầu của các chương trình ứng dụng mạng

Ứng dụng	Suy hao dữ liệu	Băng thông	Yêu cầu về thời gian
File transfer	Không chấp nhận suy hao	Mềm dẻo	Không yêu cầu thời gian trễ
E-mail	Không chấp nhận suy hao	Mềm dẻo	Không yêu cầu thời gian trễ
Web Documents	Không chấp nhận suy hao	Mềm dẻo (vài Kbps)	Không yêu cầu thời gian trễ
Real-time Audio/video	Chấp nhận suy hao	Audio: Vài Kbps Video: 10 Kb – 5 Mb	Có yêu cầu thời gian trễ: 100 $\mu$ s
Stored audio/video	Chấp nhận suy hao	Audio: Vài Kbps – 1 Mb	Có yêu cầu thời gian trễ: vài giây
Games tương tác	Chấp nhận suy hao	Vài Kbps – 10 Mb	Có yêu cầu thời gian trễ: 100 $\mu$ s
Ứng dụng tải chính	Không chấp nhận suy hao	Mềm dẻo	Có và không yêu cầu thời gian trễ

Internet cung cấp hai giao thức truyền dẫn cho các ứng dụng là UDP (*User Datagram Protocol*) và TCP (*Transmission Control Protocol*). Khi một nhà phát triển tạo ra một ứng dụng mới trên mạng Internet, đầu tiên nhà phát triển đó phải lựa chọn sử dụng giao thức UDP hay TCP. Mỗi một giao thức cung cấp một mô hình dịch vụ khác nhau để làm việc với các ứng dụng.

#### 5.1.3.4. TCP Services (Dịch vụ hướng kết nối)

Mô hình dịch vụ của TCP bao gồm dịch vụ hướng kết nối và đảm bảo truyền dữ liệu.

Khi một chương trình sử dụng giao thức truyền dẫn TCP, chương trình sẽ nhận được cả hai dịch vụ này từ giao thức TCP. TCP yêu cầu client và server thực hiện trao đổi thông tin điều khiển với nhau trước khi thông tin ở



mức ứng dụng bắt đầu được truyền đi. Quá trình này gọi là quá trình bắt tay giữa client và server, cho phép chúng chuẩn bị truyền các gói tin. Sau giai đoạn bắt tay, kết nối TCP được thiết lập giữa hai socket của hai quá trình. Kết nối ở đây là kết nối song công, hai quá trình có thể truyền thông tin cho nhau trên cùng đường truyền tại cùng một thời điểm. Khi chương trình ngừng truyền thông tin phải ngắt kết nối. Dịch vụ này được gọi là dịch vụ hướng kết nối, bởi hai quá trình được kết nối theo cách thức rất lỏng lẻo (không bền vững).

Quá trình truyền thông có thể dựa vào giao thức TCP để chuyển tất cả dữ liệu mà không có lỗi và theo đúng thứ tự. Khi chương trình chuyển dòng dữ liệu vào socket, nó có thể dựa vào TCP để chuyển dòng dữ liệu tương tự tới socket nhận mà không có sự mất mát, hoặc trùng lặp của các byte. TCP còn có cơ chế điều khiển tắc nghẽn, điều chỉnh quá trình khi trong mạng xuất hiện tắc nghẽn.

Thực tế, TCP sẽ cố gắng giới hạn từng kết nối TCP một để đáp ứng đủ băng thông dùng chung của mạng. Việc thắt nút "cổ chai" tốc độ truyền dẫn có hạn chế rất lớn đến các ứng dụng thời gian thực, đòi hỏi phải có một băng thông nhất định như audio/video. Tuy nhiên, các ứng dụng thời gian thực có khả năng chấp nhận suy hao và không cần đảm bảo truyền dữ liệu hoàn toàn tin cậy. Vì vậy, các nhà phát triển các ứng dụng thời gian thực thường chạy các ứng dụng của họ trên UDP hơn là TCP. Những dịch vụ mà TCP không cung cấp.

Đầu tiên, TCP không đảm bảo tốc độ truyền dẫn nhỏ nhất. Thực tế, quá trình truyền không cho phép truyền ở tốc độ bất kỳ mà chương trình mong muốn; thay vào đó, tốc độ truyền được quy định bởi điều khiển tắc nghẽn, dẫn tới thông tin được truyền đi với tốc độ trung bình thấp.

Thứ hai, TCP không cung cấp việc đảm bảo độ trễ. Thực tế, khi quá trình truyền chuyển dữ liệu qua socket của TCP, dữ liệu sẽ đến được socket nhận, nhưng TCP không đảm bảo giới hạn tuyệt đối khoảng thời gian mà dữ liệu đến được đích. Chẳng hạn như nhiều người phải đợi từ 10 giây cho tới hàng phút để chuyển thông tin đi. Như vậy, TCP đảm bảo truyền thông tin đi, nhưng không đảm bảo tốc độ truyền và không quan tâm tới thời gian đợi.

### 5.1.3.5. UDP Services (Dịch vụ không kết nối)

UDP là một giao thức gọn nhẹ, là mô hình có ít các dịch vụ nhất. UDP là dịch vụ không kết nối, vì vậy không cần thực hiện quá trình bắt tay trước khi hai quá trình bắt đầu truyền thông với nhau. UDP cung cấp dịch vụ truyền file không đảm bảo, có nghĩa là khi quá trình truyền thông tin vào socket của UDP, UDP không đảm bảo rằng thông tin đó có thể đến được tới đích; hơn nữa, thông tin đến được phía socket nhận thì có thể chúng không đến theo đúng thứ tự.

Ngược lại, UDP không cung cấp cơ chế điều khiển tắc nghẽn, vì vậy quá trình truyền có thể đẩy dữ liệu vào socket UDP với bất kỳ tốc độ mong muốn nào. Mặc dù tất cả dữ liệu có thể không đến được phía thu, nhưng phần lớn dữ liệu đều đến được đích. Các nhà phát triển các ứng dụng thời gian thực thường lựa chọn cách chạy các ứng dụng của họ trên UDP. Tương tự như TCP, UDP cũng không đảm bảo về thời gian trễ của các gói tin. Hình 5.4 chỉ ra các giao thức được sử dụng trong các ứng dụng phổ biến trong mạng Internet. Ta có thể nhận thấy e-mail truy cập thiết bị đầu - cuối từ xa và truyền file sử dụng TCP. Các dịch vụ này sử dụng TCP vì TCP cung cấp dịch vụ đảm bảo truyền file tin cậy, đảm bảo dữ liệu đều đến được đích. Ta cũng nhận thấy rằng, các chương trình gọi điện trên Internet thường sử dụng UDP. Trong ứng dụng gọi điện, mỗi người dùng cần truyền dữ liệu thông qua mạng ở tốc độ thường là nhỏ nhất. Điều này dường như có vẻ được thực hiện dễ đối với UDP hơn là TCP.

**Bảng 5.2.** Các chương trình ứng dụng thông thường trên Internet, các giao thức lớp ứng dụng của chúng và giao thức truyền dẫn phía dưới

Application	Giao thức lớp ứng dụng	Giao thức truyền dẫn phía dưới
Electronic mail	SMTP	TCP
Remote Terminal Access	Telnet	TCP
Web	HTTP	TCP
File Transfer	FTP	TCP
Remote File Server	NFS	UDP hoặc TCP
Streaming Multimedia	Giao thức riêng	UDP hoặc TCP
Internet Telephony	Giao thức riêng	Thường dùng UDP

## 5.2. CÁC CHƯƠNG TRÌNH ỨNG DỤNG

### 5.2.1. World Wide Web: HTTP

#### 5.2.1.1. Giao thức http

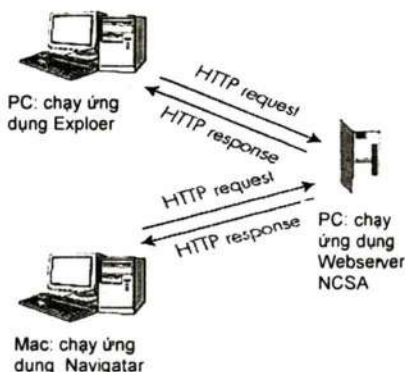
Cho đến những năm 1990, Internet vẫn được sử dụng chủ yếu bởi các nhà nghiên cứu, các trường đại học và các sinh viên đăng nhập từ xa vào máy chủ và truyền file từ máy chủ cục bộ tới máy chủ ở xa; và ngược lại, nhận và gửi các thông báo mới, gửi và nhận thư. Mặc dù các ứng dụng đó cực kỳ hữu ích (và vẫn được sử dụng) thì Internet vẫn không được phổ cập ra bên ngoài các trường học và các tổ chức nghiên cứu, do đó đến đầu những năm 1990 bắt đầu xuất hiện World Wide Web (WWW).

Web là ứng dụng trên nền Internet, nó cho phép nắm bắt được cái nhìn chung nhất của cộng đồng. Nó ngay lập tức làm thay đổi cách thức tương tác giữa môi trường làm việc bên trong và bên ngoài của mọi người. Nhiều công ty được thành lập và xây dựng Internet từ một mạng của các mạng dữ liệu trở thành một mạng dữ liệu duy nhất.

Một trang Web (hay còn gọi là một tài liệu) bao gồm các đối tượng. Mỗi đối tượng có thể là một file HTML, một ảnh JPEG, GIF và được định địa chỉ bởi một URL. Mỗi URL có hai thành phần: tên của máy chủ chứa các đối tượng và tên của đường dẫn đối tượng.

*Ví dụ:* [www.utehy.edu.vn/Categories.aspx?CategoryID=74](http://www.utehy.edu.vn/Categories.aspx?CategoryID=74). Có:  
[www.utehy.edu.vn](http://www.utehy.edu.vn) xác định tên máy chủ;  
 /Categories.aspx?CategoryID=74 là tên của đường dẫn.

Một trình duyệt là một tác nhân người dùng cho Web. Nó hiển thị các yêu cầu của trang Web và cung cấp sự điều hướng cũng như cấu hình các đặc tính. Trình duyệt Web còn được chạy bên phía máy khách.



Hình 5.4. Yêu cầu – phúc đáp của HTTP



Những trình duyệt Web thông dụng là Netscape, IE, Firefox. Web server chứa các đối tượng của trang Web, được đánh địa chỉ bởi URL. Web server thông dụng là Apache, Microsoft Internet Information Server và Netscape Enterprise Server. Khi người dùng yêu cầu trang Web, trình duyệt sẽ gửi thông báo "HTTP request" cho các đối tượng trong trang Web đến server. Server nhận yêu cầu và đáp lại bằng thông báo "HTTP response" về các đối tượng. Đến năm 1997, hầu hết các trình duyệt và các Web server đều sử dụng version HTTP/1.0, được định nghĩa trong chuẩn RFC 1945. Đến đầu năm 1998, một vài Web server và các trình duyệt bắt đầu sử dụng phiên bản HTTP/1.1, được định nghĩa trong RFC 2616. HTTP/1.1 có khả năng tương thích ngược với HTTP/1.0. Một Web server chạy phiên bản 1.1 có thể nói chuyện với trình duyệt chạy phiên bản 1.0 và ngược lại.

Cả HTTP/1.0 và HTTP/1.1 đều sử dụng giao thức lớp truyền dẫn phía dưới là TCP. HTTP client trước tiên khởi tạo kết nối TCP với server, khi kết nối được thiết lập, trình duyệt và server thực hiện truy xuất TCP thông qua các giao diện socket của chúng. Bên phía máy khách, socket ở đây là cổng nối giữa tiến trình bên máy khách và kết nối TCP. Bên phía server là cổng nối giữa tiến trình trên server và kết nối TCP. Máy khách gửi thông tin yêu cầu HTTP từ giao diện socket của nó và nhận các thông tin HTTP phản hồi cũng từ giao diện socket của nó.

Tương tự, phía server cũng làm việc thông qua các socket. Khi thông tin đi ra khỏi socket của máy khách nó sẽ thuộc quyền kiểm soát của TCP. TCP đảm bảo truyền thông tin cho HTTP. Đây chính là ưu điểm của HTTP khi mà nó không cần phải quan tâm đến hiện tượng mất gói hay thứ tự truyền không đúng, mà chúng được các giao thức lớp dưới thực hiện. Do TCP có cơ chế điều khiển tắc nghẽn, do đó nó bắt buộc các kết nối TCP mới ban đầu phải truyền với tốc độ khá thấp, nhưng nó có thể đẩy tốc độ truyền dẫn lên cao hơn khi trong mạng không có tắc nghẽn xảy ra. Điểm quan trọng đó là, server gửi file yêu cầu tới client mà không hề lưu bất kỳ một thông tin về trạng thái của client. Nếu một client yêu cầu một đối tượng hai lần trong khoảng thời gian nhỏ (vài giây), khi đó server không trả lời rằng vừa mới gửi đối tượng cho client, mà thực hiện truyền lại và quên hết những gì nó đã làm trước đó.

### 5.2.1.2. Các thông điệp http

#### a. Thông điệp yêu cầu

Khi người dùng chọn xem một trang Web, trình duyệt Web sẽ nạp trang Web đó từ Web server về và sử dụng giao thức HTTP chạy trên TCP. Giống như SMTP, HTTP là giao thức hướng ký tự. Về cốt lõi, một thông điệp HTTP có khuôn dạng tổng quát sau:

```
START_LINE <CRLF>
MESSAGE_HEADER <CRLF>
<CRLF>
MESSAGE_BODY <CRLF>
```

Dòng thứ nhất chỉ ra là thông điệp yêu cầu hay trả lời. Nó chỉ ra "*thực tế cần được thực hiện từ xa*" (trong tình huống là thông điệp yêu cầu), hoặc là "*trạng thái trả về*" (trong tình huống là thông điệp trả lời). Các hàng kế tiếp chỉ ra các tùy chọn, hoặc tham số nhằm xác định cụ thể tính chất của yêu cầu hoặc trả lời.

Dòng thứ hai: MESSAGE\_HEADER có thể không có hoặc có một vài hàng tham số và được kết thúc bằng một hàng trống. HTTP định nghĩa nhiều kiểu header, trong đó một số liên quan đến các thông điệp yêu cầu, một số liên quan đến các thông điệp trả lời, và một số lại liên quan đến phần dữ liệu trong thông điệp. Ở đây chỉ giới thiệu một số kiểu thường dùng. Dòng cuối cùng (sau hàng trống) là phần nội dung của thông điệp trả lời (MESSAGE\_BODY), phần này thường là rỗng trong thông điệp yêu cầu. Dòng thứ nhất của thông điệp yêu cầu HTTP chỉ ra ba yêu cầu: thao tác cần được thực thi, trang Web mà thao tác đó sẽ áp lên và phiên bản HTTP được sử dụng. Bảng 5.3 giới thiệu một số thao tác phổ biến.

Hai thao tác thường được sử dụng nhiều nhất là GET (lấy một trang Web về) và HEAD (lấy về thông tin của một trang Web). GET thường được sử dụng khi trình duyệt muốn tải một trang Web về và hiển thị nó cho người dùng. HEAD thường được sử dụng để kiểm tra tính hợp lệ của một liên kết siêu văn bản, hoặc để xem một trang nào đó có bị thay đổi gì không kể từ lần tải về trước đó.



**Bảng 5.3.** Các thao tác của HTTP

Hành động	Mô tả
OPTIONS	Yêu cầu thông tin về các tùy biến hiện có
GET	Lấy tài liệu được xác định trong URL
HEAD	Lấy thông tin thô về tài liệu được xác định trong URL
POST	Cung cấp thông tin cho sever
PUT	Tải tài liệu lên sever
DELETE	Xoá tài liệu nằm ở vị trí URL trên sever
TRACE	Phản hồi lại thông điệp yêu cầu
CONNECT	Được sử dụng bởi các Proxy

### b. Thông điệp trả lời

Giống như các thông điệp yêu cầu, các thông điệp trả lời bắt đầu bằng một dòng `START_LINE`. Trong trường hợp này, dòng `START_LINE` sẽ chỉ ra phiên bản HTTP đang được sử dụng, một mã 3 ký số xác định yêu cầu là thành công hay thất bại, và một chuỗi ký tự chỉ ra lý do của câu trả lời này.

*Ví dụ:* `START_LINE`.

Khi đó dòng `HTTP/1.1 202 Accepted` chỉ ra server đã có thể thoả mãn yêu cầu của người dùng. Còn dòng `HTTP/1.1 404 Not Found` chỉ ra rằng, server không thể tìm thấy tài liệu như yêu cầu. Có năm loại mã trả lời tổng quát với ký số đầu tiên xác định loại mã như trong bảng 5.4.

**Bảng 5.4.** Bảng các mã lệnh

Mã	Loại	Lý do
1xx	Thông tin	Đã nhận được yêu cầu, đang tiếp tục xử lý
2xx	Thành công	Thao tác đã được tiếp nhận, hiểu và chấp nhận được
3xx	Chuyển hướng	Cần thực hiện thêm thao tác để hoàn tất yêu cầu được đặt ra
4xx	Lỗi client	Yêu cầu do cú pháp sai hoặc không thể được đáp ứng
5xx	Lỗi sever	Sever thất bại trong việc đáp ứng một yêu cầu hợp lệ

Cũng giống như các thông điệp yêu cầu, các thông điệp trả lời có thể chứa một hoặc nhiều dòng trong phần MESSAGE\_HEADER. Những dòng này cung cấp thêm thông tin cho client. Ví dụ, dòng Header Location chỉ ra rằng, URL được yêu cầu đang có ở vị trí khác. Trong tình huống chung nhất, thông điệp trả lời cũng sẽ mang theo nội dung trang Web được yêu cầu. Trang này là một tài liệu HTML, nhưng vì nó có thể chứa dữ liệu không phải dạng văn bản (ví dụ như ảnh GIF), dữ liệu này có thể được mã hoá theo dạng MIME. Một số dòng trong phần MESSAGE\_HEADER cung cấp thêm thông tin về nội dung của trang Web, bao gồm: Content - Length (số byte trong phần nội dung), Expires (thời điểm mà nội dung trang Web được xem như lỗi thời), Last - Modified (thời điểm được sửa đổi lần cuối cùng)...

### 5.2.1.3. Các kết nối TCP

Nguyên tắc chung của giao thức HTTP là client nối kết đến cổng TCP số 80 tại server, server luôn "lắng nghe" trên cổng này để sẵn sàng phục vụ client. Phiên bản đầu tiên (HTTP/1.0) thiết lập một nối kết riêng cho mỗi hạng mục dữ liệu cần tải về từ server. Có thể thấy rằng, đây là cơ chế không mấy hiệu quả: Các thông điệp dùng để thiết lập và giải phóng nối kết sẽ phải được trao đổi qua lại giữa client và server, dẫn đến khi mà tất cả client muốn lấy thông tin mới nhất của một trang Web, server sẽ bị quá tải.

Cải tiến quan trọng nhất trong phiên bản HTTP/1.1 là nó cho phép các kết nối lâu dài - client và server sẽ trao đổi nhiều thông điệp yêu cầu/trả lời trên cùng một kết nối TCP. Kết nối lâu dài có hai cái lợi. Thứ nhất, nó làm giảm thiểu chi phí cho việc thiết lập/giải phóng nối kết. Thứ hai, do client gửi nhiều thông điệp yêu cầu qua một kết nối TCP, dẫn đến cơ chế điều khiển tắc nghẽn của TCP sẽ hoạt động hiệu quả hơn.

Tuy nhiên, kết nối lâu dài cũng có cái giá phải trả. Vấn đề phát sinh ở chỗ: Không ai trong client và server biết được kết nối đó sẽ kéo dài bao lâu. Điều này thực sự gây khó khăn cho phía server, bởi vì tại mỗi thời điểm, nó phải đảm bảo duy trì kết nối đến cả ngàn client. Giải pháp cho vấn đề này là: server sẽ mãn kỳ và cắt nối kết nếu nó không nhận được một yêu cầu cụ thể

nào từ phía client trong một khoảng thời gian định trước. Ngoài ra, cả client và server phải theo dõi xem phía bên kia có chủ động cắt nối kết hay không, và lấy đó làm cơ sở để tự cắt nối kết của mình. Nhắc lại rằng, cả hai bên phải cắt nối kết thì nối kết TCP mới thực sự kết thúc.

#### 5.2.1.4. Bộ đệm Web (Proxy server)

Một trong những lĩnh vực nghiên cứu tích cực nhất hiện nay về Internet là làm sao để trữ tạm các trang Web một cách hiệu quả. Việc trữ tạm mang lại nhiều lợi ích. Từ phía client, việc nạp và hiển thị một trang Web từ bộ đệm gần đây là nhanh hơn rất nhiều so với từ một server nào đó ở nửa vòng Trái Đất. Đối với server, có thêm một bộ đệm để can thiệp vào và phục vụ giúp yêu cầu của người dùng sẽ giảm bớt tải trên server. Việc trữ đệm có thể được cài đặt tại nhiều nơi khác nhau.

*Ví dụ:* Trình duyệt Web có thể trữ tạm những trang Web mới được nạp về gần đây, để khi người dùng duyệt lại những trang Web đó, trình duyệt sẽ không phải nối kết ra Internet để lấy chúng về, mà dùng bản trữ sẵn. Ví dụ khác, một khu vực làm việc (site) có thể thuê một máy làm nhiệm vụ trữ tạm các trang Web, để những người dùng sau có thể sử dụng các bản trữ sẵn của những người dùng trước. Yêu cầu của hệ thống này là mọi người dùng trong site phải biết địa chỉ của máy tính làm nhiệm vụ bộ trữ tạm, và họ chỉ đơn giản là liên hệ với máy tính này để tải các trang Web về theo yêu cầu. Người ta thường gọi máy tính làm nhiệm vụ trữ tạm các trang Web cho một site là proxy. Vị trí trữ đệm có thể di chuyển gần hơn đến phần lõi của Internet là các ISP. Trong tình huống này, các site nối kết tới ISP thường không hay biết gì về việc trữ tạm ở đây.

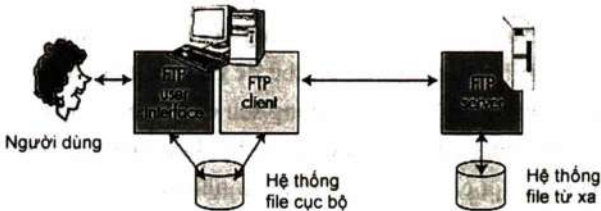
Khi các yêu cầu HTTP từ các site được chuyển phát đến router của ISP, router liền kiểm tra URL được yêu cầu có giống với các URL được trữ sẵn hay không. Nếu có, router sẽ trả lời ngay; nếu không, router sẽ chuyển yêu cầu đến server thật sự và cũng không quên lưu vào bộ đệm của mình thông điệp trả lời từ phía server đó. Việc trữ tạm là đơn giản; tuy nhiên, bộ đệm phải đảm bảo những thông tin trữ đệm không quá cũ. Để làm được điều này, các Web server phải gán "ngày hết hạn" (tức là trường Expires trong header) cho mọi trang Web mà nó phục vụ cho client. Nhân đó, các bộ đệm



cũng lưu lại thông tin này. Và từ đó, các bộ đệm sẽ không cần phải kiểm tra tình cập nhật của trang Web đó cho đến khi ngày hết hạn đến. Tại thời điểm một trang Web hết hạn, bộ đệm sẽ dùng lệnh HEAD hoặc lệnh GET có điều kiện (GET với trường If – Modified – Since) trong phần header được đặt để kiểm tra rằng, nó có một phiên bản mới nhất của trang Web kia. Tổng quát hơn, cần phải có "các chỉ thị hướng dẫn" cho việc trữ đệm và các chỉ thị này phải được tuân thủ tại mọi bộ đệm. Các chỉ thị sẽ chỉ ra có nên trữ đệm một tài liệu hay không, trữ nó bao lâu, một tài liệu phải tươi như thế nào...

### 5.2.2. File transfer: FTP

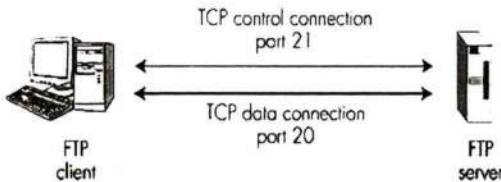
FTP (*File Transfer Protocol*) là một giao thức truyền file từ một thiết bị chủ này tới một thiết bị chủ khác. Giao thức này được quy định trước năm 1971 (khi Internet vẫn chỉ là thí nghiệm) nhưng vẫn còn được dùng rất phổ biến. FTP được quy định trong chuẩn RFC 959. Hình 5.5 đưa ra mô hình tổng quát về các dịch vụ cung cấp bởi FTP.



Hình 5.5. Mô hình dịch vụ FTP

Một phiên FTP thông thường, người dùng trên một máy và muốn truyền file tới máy đích. Người dùng phải có một account (tài khoản) và password (mật khẩu) để xác định người dùng. Sau khi cung cấp những thông tin xác minh như trên, người dùng có thể truyền file từ hệ thống cục bộ tới hệ thống từ xa và ngược lại. Người dùng tương tác với FTP thông qua tác nhân người dùng. Người sử dụng trước tiên phải cung cấp tên của máy chủ và tên của máy đích, tạo thành tiến trình FTP trên máy chủ thiết lập kết nối TCP tới tiến trình FTP trên server trên máy đích. Người dùng sau đó cung cấp thông tin xác minh và password thông qua kết nối TCP với các câu lệnh FTP. Khi server xác nhận người dùng, người dùng sao chép một hoặc nhiều file được lưu trữ trong hệ thống file cục bộ tới hệ thống máy từ xa và ngược lại.

Hai phương thức truyền file HTTP và FTP đều có chung các đặc điểm, ví dụ như chúng đều chạy trên giao thức TCP. Tuy nhiên, hai giao thức lớp ứng dụng này đều có các điểm khác biệt quan trọng. Điểm khác biệt nổi bật nhất là FTP sử dụng các kết nối đồng thời cho việc truyền file, kết nối điều khiển, kết nối dữ liệu. Kết nối điều khiển được sử dụng cho việc gửi thông tin điều khiển giữa hai thiết bị chủ – thông tin như là xác minh người dùng, password, câu lệnh để thay đổi thư mục máy đích và các câu lệnh gửi và nhận file. Đường kết nối dữ liệu thực tế chỉ dùng để gửi file. Do FTP sử dụng một kết nối điều khiển độc lập, FTP gửi thông tin điều khiển ra ngoài dải băng tần. HTTP gửi yêu cầu và nhận thông tin phản hồi trên cùng kết nối TCP mà nó tự chuyển đổi file. Kết nối điều khiển và dữ liệu FTP được minh họa trên hình 5.6.



**Hình 5.6.** Kết nối điều khiển và kết nối dữ liệu

Khi người dùng bắt đầu phiên làm việc FTP với thiết bị từ xa, FTP đầu tiên thiết lập kết nối điều khiển TCP ở cổng 21 của máy chủ. Bên phía client, FTP gửi thông tin xác minh người dùng trên kết nối điều khiển, câu lệnh thay đổi thư mục trên máy ở xa. Khi người dùng yêu cầu truyền file, FTP mở kết nối dữ liệu TCP trên cổng 20 của server. FTP gửi chính xác một file trên đường kết nối dữ liệu. Nếu trong cùng phiên làm việc, người dùng muốn truyền nhiều file, FTP mở thêm các đường kết nối dữ liệu khác.

Do đó, với FTP kết nối điều khiển vẫn còn được mở trong suốt phiên làm việc của người dùng; tuy nhiên, khi truyền một file mới kết nối mới phải được thiết lập, do đó kết nối FTP là kết nối không liên tục. Trong suốt phiên làm việc của người dùng, FTP server phải duy trì trạng thái về người dùng. Trong thực tế, server kết hợp kết nối điều khiển với một tài khoản người dùng nhất định và server luôn phải theo dõi sự thay đổi thư mục người dùng trên máy đích. Ngược lại, HTTP không duy trì trạng thái thông tin người dùng.



**Bảng 5.5.** Các lệnh cơ bản của FTP

Lệnh	Tham số	Ý nghĩa
FTP	Host - name	Nối kết đến FTP có địa chỉ Host - name
USER	User - name	Cung cấp tên người dùng cho FTP sever để thực hiện quá trình chứng thực
ASCII		Chỉ định kiểu dữ liệu truyền nhận là ký tự
BINARY		Chỉ định kiểu dữ liệu truyền nhận là nhị phân
LS		Xem nội dung thư mục từ xa
CD	Remote - dir	Chuyển đến thư mục khác trong hệ thống tập tin từ xa
GET	Remote - file Local - file	Tải tập tin Remote – file trên FTP sever về hệ thống tập tin cục bộ và đặt tên là Local - file
PUT	Local - file Remote - file	Nạp tập tin cục bộ Local - file lên sever và đặt tên là Remote - file
MKDIR	Dir - name	Tạo một thư mục có tên Dir - name trên hệ thống tập tin từ xa
RMDIR	Dir - name	Xoá thư mục có tên Dir - name trên hệ thống tập tin từ xa
QUIT		Đóng nối kết FTP và thoát khỏi chương trình FTP

### 5.2.3. Hệ thống tên miền DNS (Domain Name System)

#### 5.2.3.1. Giới thiệu

##### a. Lịch sử hình thành của DNS

Vào những năm 1970, mạng ARPAnet của Bộ Quốc phòng Mỹ rất nhỏ và dễ dàng quản lý các liên kết vài trăm máy tính với nhau. Do đó, mạng chỉ cần một file HOSTS.TXT chứa tất cả thông tin cần thiết về máy tính trong mạng và giúp các máy tính chuyển đổi được thông tin địa chỉ và tên mạng cho tất cả máy tính trong mạng ARPAnet một cách dễ dàng. Đó chính là bước khởi đầu của hệ thống tên miền, gọi tắt là DNS (*Domain Name System*). Khi mạng máy tính ARPAnet ngày càng phát triển, thì việc quản lý thông tin chỉ dựa vào một file HOSTS.TXT là rất khó khăn và không khả thi.

Vi thông tin bổ sung và sửa đổi vào file HOSTS.TXT ngày càng nhiều, nhất là khi ARPAnet phát triển hệ thống máy tính dựa trên giao thức TCP/IP, dẫn đến sự phát triển tăng vọt của mạng máy tính:

- Lưu lượng và trao đổi trên mạng tăng lên.
- Tên miền trên mạng và địa chỉ ngày càng nhiều.
- Mật độ máy tính ngày càng cao, do đó đảm bảo phát triển ngày càng

khó khăn. Đến năm 1984, Paul Mockpetris thuộc viện USC's Information Sciences Institute phát triển một hệ thống quản lý tên miền mới (miêu tả trong chuẩn RFC 882 – 883), gọi là DNS (*Domain Name System*) và được phát triển, hiệu chỉnh, bổ sung tính năng để đảm bảo yêu cầu ngày càng cao của hệ thống (hiện nay DNS được tiêu chuẩn theo chuẩn RFC 1034 – 1035).

### **b. Mục đích của hệ thống DNS**

Máy tính khi kết nối vào mạng Internet thì được gán cho một địa chỉ IP xác định. Địa chỉ IP của mỗi máy là duy nhất và có thể giúp máy tính có thể xác định đường đi đến một máy tính khác một cách dễ dàng. Như đối với người dùng thì địa chỉ IP là rất khó nhớ. Do vậy cần phải sử dụng một hệ thống để giúp cho máy tính tính toán đường đi một cách dễ dàng và đồng thời cũng giúp người dùng dễ nhớ.

Hệ thống DNS ra đời nhằm giúp cho người dùng có thể chuyển đổi từ địa chỉ IP khó nhớ mà máy tính sử dụng sang một tên dễ nhớ cho người sử dụng và đồng thời nó giúp cho hệ thống Internet dễ dàng sử dụng để liên lạc và ngày càng phát triển.

Hệ thống DNS sử dụng hệ thống cơ sở dữ liệu phân tán và phân cấp hình cây do đó việc quản lý sẽ dễ dàng và cũng rất thuận tiện cho việc chuyển đổi từ tên miền sang địa chỉ IP và ngược lại. Cũng giống như mô hình quản lý cá nhân của một đất nước mỗi cá nhân sẽ có một tên xác định đồng thời cũng có địa chỉ chứng minh thư để giúp quản lý con người một cách dễ dàng hơn (nhưng khác là tên miền không được trùng nhau còn tên người thì vẫn có thể trùng nhau). Mỗi cá nhân đều có một số căn cước để quản lý.

Mỗi một địa chỉ IP tương ứng với một tên miền. Vậy tóm lại tên miền là (domain name) gì? những tên gọi nhớ như home.vnn.vn hoặc www.cnn.com thì được gọi là tên miền (*Domain name hoặc DNS name*). Nó giúp cho người sử dụng dễ dàng nhớ vì nó ở dạng chữ mà người bình thường có thể hiểu và sử dụng hàng ngày.



**Hình 5.7.** Mô tả địa chỉ cho DNS

Hệ thống DNS đã giúp cho mạng Internet thân thiện hơn với người sử dụng do đó mạng Internet phát triển bùng nổ một vài năm lại đây. Theo thống kê trên thế giới vào thời điểm tháng 7/2000 số lượng tên miền được đăng ký là 93.000.000

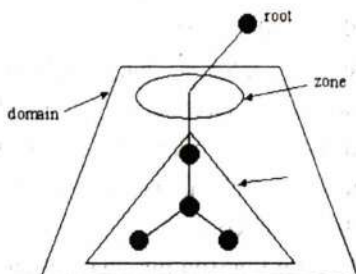
Tóm lại, mục đích của hệ thống DNS là:

- Địa chỉ IP khó nhớ cho người sử dụng nhưng dễ dàng với máy tính.
- Tên thì dễ nhớ với người sử dụng như không dùng được với máy tính.
- Hệ thống DNS giúp chuyển đổi từ tên miền sang địa chỉ IP và ngược lại giúp người dùng dễ dàng sử dụng hệ thống máy tính.

### 5.2.3.2. DNS server và cấu trúc cơ sở dữ liệu tên miền

#### a. Cấu trúc cơ sở dữ liệu

Cơ sở dữ liệu của hệ thống DNS là hệ thống cơ sở dữ liệu phân tán và phân cấp hình cây. Với Root server là đỉnh của cây và sau đó các domain được phân nhánh dần xuống dưới và phân quyền quản lý. Khi một client truy vấn một tên miền nó sẽ lần lượt đi từ root phân cấp lần lượt xuống dưới để đến DNS quản lý domain cần truy vấn.



**Hình 5.8.** Cấu trúc cơ sở dữ liệu DNS

Cấu trúc của dữ liệu được phân cấp hình cây root quản lý toàn bộ sơ đồ và phân quyền quản lý xuống dưới và tiếp đó các tên miền lại được tiếp tục chuyển xuống cấp thấp hơn (*Delegate*) xuống dưới.

#### – Zone

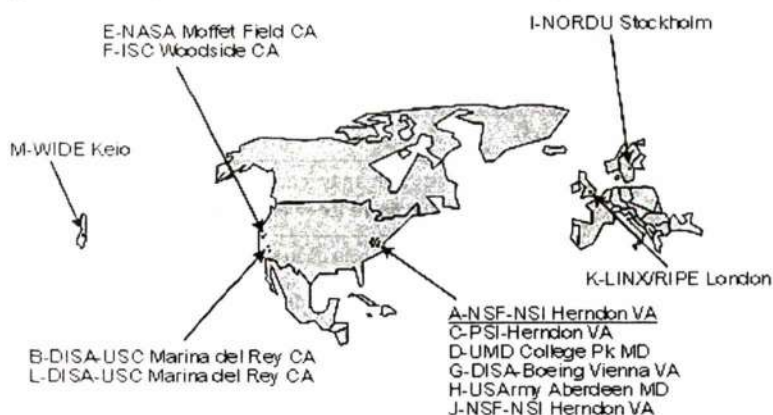
Hệ thống DNS cho phép phân chia tên miền để quản lý và nó chia hệ thống tên miền ra thành zone và trong zone quản lý tên miền được phân chia đó và nó chứa thông tin về domain cấp thấp hơn và có khả năng chia thành các zone cấp thấp hơn và phân quyền cho các DNS server khác quản lý.

*Tỉ dụ:* zone ".com" thì DNS server quản lý zone ".com" chứa thông tin về các bản ghi có đuôi là ".com" và có khả năng chuyển quyền quản lý (delegate) các zone cấp thấp hơn cho các DNS khác quản lý như ".microsoft.com" là vùng (zone) do microsoft quản lý.

#### – Root Server

Là server quản lý toàn bộ cấu trúc của hệ thống DNS. Root server không chứa dữ liệu thông tin về cấu trúc hệ thống DNS mà nó chỉ chuyển quyền quản lý xuống cho các server cấp thấp hơn và do đó root server có khả năng xác định đường đến của một domain tại bất cứ đâu trên mạng.

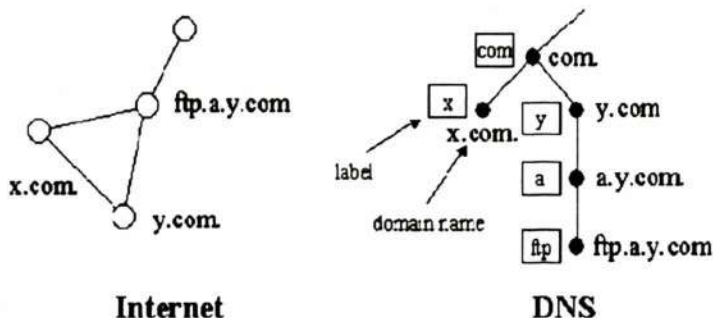
Hiện nay trên thế giới có khoảng 13 root server quản lý toàn bộ hệ thống Internet (vị trí của root server như trên hình 5.9).



**Hình 5.9.** Hệ thống DNS trên thế giới



Hệ thống cơ sở dữ liệu của DNS là hệ thống dữ liệu phân tán hình cây, cấu trúc đó là cấu trúc logic trên mạng Internet.



**Hình 5.10.** Cấu trúc của DNS trên mạng Internet

Về mặt vật lý, hệ thống DNS nằm trên mạng Internet không có cấu trúc hình cây nhưng nó được cấu hình phân cấp logic phân cấp hình cây phân quyền quản lý. Một DNS server có thể nằm bất cứ vị trí nào trên mạng Internet nhưng được cấu hình logic để phân cấp chuyển tên miền cấp thấp hơn xuống cho các DNS server khác nằm bất cứ vị trí nào trên mạng Internet (về nguyên tắc ta có thể đặt DNS tại bất cứ vị trí nào trên mạng Internet. Nhưng tốt nhất là đặt DNS tại vị trí nào gần với các client để dễ dàng truy vấn đến, đồng thời cũng gần với vị trí của DNS server cấp cao hơn trực tiếp của nó).

**Bảng 5.6.** Các bản ghi trên DNS

Tên trường	Tên đầy đủ	Mục đích
SOA	Satrt of Authority	Xác định máy chủ DNS có thẩm quyền cung cấp thông tin về tên miền xác định trên DNS.
NS	Name Sever	Chuyển quyền quản lý tên miền xuống một DNS thấp hơn.
A	Host	Ảnh xạ xác định địa chỉ IP của một host.
MX	Mail Exchanger	Xác định host có quyền quản lý thư điện tử cho một tên miền xác định.
PTR	Pointer	Xác định chuyển từ địa chỉ IP sang tên miền.
CNAME	Canonical NAME	Thường sử dụng xác định dịch vụ Web hosting.



Mỗi một tên miền đều được quản lý bởi ít nhất một DNS server và trên đó ta khai các bản ghi của tên miền trên DNS server. Các bản ghi đó sẽ xác định địa chỉ IP của tên miền hoặc các dịch vụ xác định trên Internet như web, thư điện tử...

### b. Cấu trúc của một tên miền

– Domain sẽ có dạng: lable.lable.label...lable.

– Độ dài tối đa của một tên miền là 255 ký tự.

– Mỗi một Lable tối đa là 63 ký tự. Lable phải bắt đầu bằng chữ hoặc số và chỉ được phép chứa chữ, số, dấu trừ (-), dấu chấm (.) mà không được chứa các ký tự khác.

– Phân loại tên miền

Hầu hết tên miền được chia thành các loại sau:

+ Arpa: tên miền ngược (chuyển đổi từ địa chỉ IP sang tên miền reverse domain).

+ Com: các tổ chức thương mại.

+ Edu: các cơ quan giáo dục.

+ Gov: các cơ quan chính phủ.

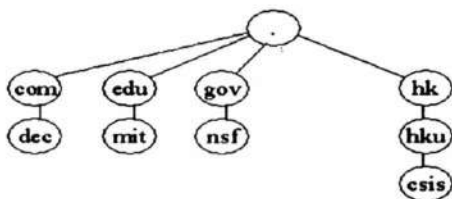
+ Mil: các tổ chức quân sự, quốc phòng.

+ Net: các trung tâm mạng lớn.

+ Org: các tổ chức khác.

+ Int: các tổ chức đa chính phủ (ít được sử dụng).

Ngoài ra, hiện nay trên thế giới sử dụng loại tên miền có hai ký tự cuối để xác định tên miền thuộc quốc gia nào (được xác định trong chuẩn ISO3166).



Hình 5.11. Hình các loại tên miền được xác định trong chuẩn ISO3166

**Bảng 5.7.** Bảng các cấp trong DNS

Loại tên	Miêu tả	Vi dụ
Gốc (domain root)	Nó là đỉnh của nhánh cây của tên miền. Nó xác định kết thúc của domain (fully qualified domain names FQDNs).	Đơn giản nó chỉ là dấu chấm (.) sử dụng tại cuối của tên ví như "example.microsoft.com."
Tên miền cấp một (Top-level domain)	Là hai hoặc ba ký tự xác định nước/khu vực hoặc các tổ chức.	".com", xác định tên sử dụng trong xác định là tổ chức thương mại.
Tên miền cấp hai (Second-level domain)	Nó rất đa dạng trên Internet, nó có thể là tên của một công ty, một tổ chức hay một cá nhân.v.v. đăng ký trên Internet.	"microsoft.com.", là tên miền cấp hai đăng ký là công ty Microsoft.
Tên miền cấp nhỏ hơn (Subdomain)	Chia nhỏ thêm ra của tên miền cấp hai xuống thường được sử dụng như chi nhánh, phòng ban của một cơ quan hay một chủ đề nào đó.	"example.microsoft.com." là phần quản lý tài liệu ví dụ của microsoft.

– Một số chú ý khi đặt tên miền

+ Tên miền nên đặt giới hạn từ cấp 3 đến cấp 4, hoặc cấp 5 vì nếu nhiều hơn việc quản trị sẽ khó khăn.

+ Sử dụng tên miền là phải duy nhất trong mạng Internet.

+ Nên đặt tên đơn giản gợi nhớ và tránh đặt tên quá dài.

### 5.2.3.3. Phân loại DNS server và đồng bộ dữ liệu giữa các DNS server

Có ba loại DNS server sau:

#### • Primary server

Nguồn xác thực thông tin chính thức cho các domain mà nó được phép quản lý. Thông tin về tên miền do nó được phân cấp quản lý thì được lưu trữ tại đây và sau đó có thể được chuyển sang cho các secondary server. Các tên

miền do primary server quản lý thì được tạo và sửa đổi tại primary server và sau đó được cập nhật đến các secondary server.

- **Secondary server**

DNS được khuyến nghị nên sử dụng ít nhất là hai DNS server để lưu cho mỗi một zone. Primary DNS server quản lý các zone và secondary server được sử dụng để lưu trữ dự phòng cho zone cho primary server. Secondary DNS server được khuyến nghị dùng nhưng không nhất thiết phải có. Secondary server được phép quản lý domain nhưng dữ liệu về domain không phải tạo tại secondary server mà nó được lấy về từ primary server.

Secondary server có thể cung cấp hoạt động ở chế độ không có tải trên mạng. Khi lượng truy vấn zone tăng cao tại primary server nó sẽ chuyển bớt tải sang secondary server hoặc khi primary server bị sự cố thì secondary sẽ hoạt động thay thế cho đến khi primary server hoạt động trở lại. Secondary server nên được sử dụng tại nơi gần với client để có thể phục vụ cho việc truy vấn tên miền một cách dễ dàng. Nhưng không nên cài đặt secondary server trên cùng một subnet hoặc cùng một kết nối với primary server. Vì điều đó sẽ là một giải pháp tốt để sử dụng secondary server để dự phòng cho primary server, vì có thể kết nối đến primary server bị hỏng thì cũng không ảnh hưởng gì đến secondary server. Primary server luôn luôn duy trì một lượng lớn dữ liệu và thường xuyên thay đổi hoặc thêm vào các zone.

Do đó, DNS server sử dụng một cơ chế cho phép chuyển các thông tin từ primary server sang secondary server và lưu giữ nó trên đĩa. Các thông tin nhận dữ liệu về các zone có thể sử dụng giải pháp lấy toàn bộ (*Full*) hoặc lấy phần thay đổi (*Incremental*). Nhiều secondary DNS server sẽ tăng độ ổn định hoạt động của mạng và việc lưu trữ thông tin của tên miền một cách đảm bảo như một điều cần quan tâm là dữ liệu của zone được chuyển trên mạng từ primary server đến các secondary server sẽ làm tăng lưu lượng đường truyền và yêu cầu thời gian để đồng bộ dữ liệu trên các secondary server.

- **Caching-only server**

Mặc dù tất cả các DNS server đều có khả năng lưu trữ dữ liệu trên bộ nhớ cache của máy để trả lời truy vấn một cách nhanh chóng, caching-only

server là loại DNS server chỉ sử dụng cho việc truy vấn, lưu giữ câu trả lời dựa trên thông tin trên cache của máy và cho kết quả truy vấn. Chúng không hề quản lý một domain nào và thông tin mà nó chỉ giới hạn những gì được lưu trên cache của server.

Khi nào thì sử dụng caching-only server?. Khi mà server bắt đầu chạy thì nó không có thông tin lưu trong cache. Thông tin sẽ được cập nhật theo thời gian khi các client server truy vấn dịch vụ DNS. Nếu bạn sử dụng kết nối mạng WAN tốc độ thấp thì việc sử dụng caching-only DNS server là một giải pháp tốt, nó cho phép giảm lưu lượng thông tin truy vấn trên đường truyền.

*Chú ý:* Caching-only DNS server không chứa zone nào và cũng không quản lý bất kỳ domain nào. Nó sử dụng bộ nhớ cache của mình để lưu các truy vấn DNS của client. Thông tin sẽ được lưu trong cache để trả lời cho các truy vấn đến của client. Caching-only DNS có khả năng trả lời các truy vấn nhưng không quản lý hoặc tạo bất cứ zone hoặc domain nào DNS server nói chung được khuyến nghị là được cấu hình sử dụng TCP/IP và dùng địa chỉ IP tĩnh. Đồng bộ dữ liệu giữa các DNS server (*zone transfer*).

### **Cách thức truyền dữ liệu:**

#### *– Truyền toàn bộ zone*

Bởi vì tầm quan trọng của hệ thống DNS và việc quản lý các domain thuộc zone phải được đảm bảo, do đó một zone thường được đặt trên hơn một DNS server để tránh lỗi khi truy vấn tên miền thuộc zone đó. Nói cách khác, nếu chỉ có một server quản lý zone và khi server không trả lời truy vấn thì các tên miền trong zone đó sẽ không được trả lời và không còn tồn tại trên Internet.

Do đó, cần có nhiều DNS server cùng quản lý một zone và có cơ chế để chuyển dữ liệu của các zone và đồng bộ nó từ một DNS server này đến các DNS server khác. Khi một DNS server mới được thêm vào mạng thì nó được cấu hình như một secondary server mới cho một zone đã tồn tại. Nó sẽ tiến hành nhận toàn bộ (full) zone từ DNS server khác. Như DNS server thế



hệ đầu tiên thường dùng giải pháp lấy toàn bộ cơ sở dữ liệu về zone khi có các thay đổi trong zone.

– *Truyền phân thay đổi (Incremental zone)*

Truyền chỉ những thay đổi (*Incremental zone transfer*) của zone được miêu tả chi tiết trong tiêu chuẩn RFC 1995. Nó là phần bổ sung cho chuẩn sao chép DNS zone. Incremental transfer thì được hỗ trợ bởi cả DNS server là nguồn lấy thông tin và DNS server nhận thông tin về zone, nó cung cấp giải pháp hiệu quả cho việc đồng bộ nhưng thay đổi hoặc thêm bớt zone.

Giải pháp ban đầu cho DNS yêu cầu cho việc thay đổi dữ liệu về zone là truyền toàn bộ dữ liệu của zone sử dụng truy vấn AXFR. Với việc chi chuyển các thay đổi (incremental transfer) sẽ sử dụng truy vấn IXFR được sử dụng thay thế cho AXFR. Nó cho phép secondary server chỉ lấy về như zone thay đổi để đồng bộ dữ liệu. Với trao đổi IXFR zone, thì có sự khác nhau giữa versions của nguồn dữ liệu và bản sao của nó. Nếu cả hai bản đều có cùng version (xác định bởi số serial trong khai báo tại phần đầu của zone SOA "start of authority") thì việc truyền dữ liệu của zone sẽ không được thực hiện. Nếu số serial cho dữ liệu nguồn lớn hơn số serial của secondary server thì nó sẽ thực hiện chuyển những thay đổi với các bản ghi nguồn (Resource record – RR) của zone.

Để truy vấn IXFR thực hiện thành công và các thay đổi được gửi thì tại DNS server nguồn của zone phải lưu giữ các phần thay đổi để sử dụng truyền đến nơi yêu cầu của truy vấn IXFR. Incremental sẽ cho phép lưu lượng truyền dữ liệu là ít và thực hiện nhanh hơn.

SOA vdc-hn01.vnn.vn. postmaster.vnn.vn.

82802 ; serial number  
 ; refresh every 30 mins  
 ; retry every hour  
 ; expire after 24 hours  
 ; minimum TTL 2 hours

DNS vdc-hn01.vnn.vn.

DNS hcm-server1.vnn.vn.



Zone transfer sẽ xảy ra khi có những hành động sau xảy ra:

- + Khi quá trình làm mới của zone kết thúc (refresh expire)
- + Khi secondary server được thông báo zone đã thay đổi tại server nguồn quản lý zone.
- + Khi dịch vụ DNS bắt đầu chạy tại secondary server.

Tại secondary server yêu cầu chuyển zone.

Sau đây là các bước yêu cầu từ secondary server đến DNS server chứa zone để yêu cầu lấy dữ liệu về zone mà nó quản lý.

Trong khi cấu hình mới DNS server, nó sẽ gửi truy vấn yêu cầu gửi toàn bộ zone ("all zone" transfer (AXFR) request) đến DNS server quản lý chính dữ liệu của zone).

DNS server chính quản lý dữ liệu của zone sẽ trả lời và chuyển toàn bộ dữ liệu về zone đến secondary (destination) server mới cấu hình. zone thì được chuyển đến DNS server yêu cầu căn cứ vào version được xác định bằng số Serial tại phần khai báo (*start of authority SOA*). Tại phần SOA cũng có chứa các thông số xác định thời gian làm mới lại zone...

Khi thời gian làm mới (refresh interval) của zone hết, thì DNS server nhận dữ liệu sẽ truy vấn yêu cầu làm mới zone tới DNS server chính chứa dữ liệu zone.

DNS server chính quản lý dữ liệu sẽ trả lời truy vấn và gửi lại dữ liệu. Trả lời sẽ bao gồm cả số serial của zone hiện tại tại DNS server chính.

DNS server nhận dữ liệu về zone sẽ kiểm tra số serial trong trả lời và quyết định sẽ làm thế nào với zone:

- + Nếu giá trị của số serial bằng với số hiện tại tại DNS server nhận trả lời thì nó sẽ kết luận rằng sẽ không cần chuyển dữ liệu về zone đến. Và nó sẽ thiết lập lại với các thông số cũ và thời gian để làm mới lại bắt đầu.

- + Nếu giá trị của số serial tại DNS server chính lớn hơn giá trị hiện tại tại dữ liệu DNS nơi nhận thì nó kết luận rằng zone cần phải được cập nhật và việc chuyển zone là cần thiết.

+ Nếu DNS server nơi nhận kết luận rằng zone cần phải thay đổi và nó sẽ gửi truy vấn IXFR tới DNS server chính để yêu cầu gửi zone.

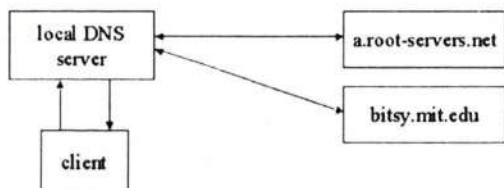
DNS server chính sẽ trả lời với việc gửi những thay đổi của zone hoặc toàn bộ zone:

+ Nếu DNS server chính có hỗ trợ việc gửi những thay đổi của zone thì nó sẽ gửi những phần thay đổi (*Incremental zone transfer (IXFR) of the zone*).

+ Nếu nó không hỗ trợ thì nó sẽ gửi toàn bộ zone (*Full AXFR transfer of the zone*).

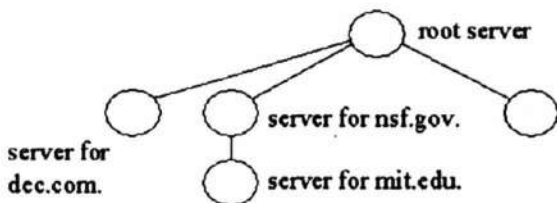
#### 5.2.3.4. Hoạt động của hệ thống DNS

Hệ thống DNS hoạt động tại lớp 4 của mô hình OSI nó sử dụng truy vấn bằng giao thức UDP và mặc định là sử dụng cổng 53 để trao đổi thông tin về tên miền.



Hình 5.12. Miêu tả hoạt động của DNS

Hoạt động của hệ thống DNS là chuyển đổi tên miền sang địa chỉ IP và ngược lại. Hệ thống cơ sở dữ liệu của DNS là hệ thống cơ sở dữ liệu phân tán, các DNS server được phân quyền quản lý các tên miền xác định và chúng liên kết với nhau để cho phép người dùng có thể truy vấn một tên miền bất kỳ (có tồn tại) tại bất cứ điểm nào trên mạng một cách nhanh nhất.



Hình 5.13. Cách thức truy vấn trong DNS

Như đã trình bày các DNS server phải biết ít nhất một cách để đến được root server và ngược lại. Như trên hình 5.13 muốn xác định được tên miền mit.edu thì root server phải biết DNS server nào được phân quyền quản lý tên miền mit.edu để chuyển truy vấn đến.

Nói tóm lại tất cả các DNS server đều được kết nối một cách logic với nhau: Tất cả các DNS server đều được cấu hình để biết ít nhất một cách đến root server. Một máy tính kết nối vào mạng phải biết làm thế nào để liên lạc với ít nhất là một DNS server.

### • Hoạt động của DNS

Khi DNS client cần xác định cho một tên miền nó sẽ truy vấn DNS. Truy vấn DNS và trả lời của hệ thống DNS cho client sử dụng thủ tục UDP cổng 53, UDP hoạt động ở mức thứ 3 (network) của mô hình OSI, UDP là thủ tục phi kết nối (connectionless), tương tự như dịch vụ gửi thư bình thường bạn cho thư vào thùng thư và hy vọng có thể chuyển đến nơi bạn cần gửi tới.

Mỗi một message truy vấn được gửi đi từ client bao gồm ba phần thông tin:

- Tên của miền cần truy vấn (tên đầy đủ FQDN).
- Xác định loại bản ghi là mail, web...
- Lớp tên miền (phần này thường được xác định là IN Internet, ở đây không đi sâu vào phần này).

*Ví dụ:* tên miền truy vấn đầy đủ như "hostname.example.microsoft.com." và loại truy vấn là địa chỉ A. Client truy vấn DNS hỏi: Có bản ghi địa chỉ A cho máy tính có tên là "hostname.example.microsoft.com" khi client nhận được câu trả lời của DNS server nó sẽ xác định địa chỉ IP của bản ghi A. Có một số giải pháp để trả lời các truy vấn DNS. Client có thể tự trả lời bằng cách sử dụng các thông tin đã được lưu trữ trong bộ nhớ cache của nó từ những truy vấn trước đó. DNS server có thể sử dụng các thông tin được lưu trữ trong cache của nó để trả lời hoặc DNS server có thể hỏi một DNS server khác lấy thông tin đó để trả lời lại client.

Các bước của một truy vấn gồm có hai phần như sau:

– Truy vấn sẽ bắt đầu ngay tại client computer để xác định câu trả lời. Khi ngay tại client không có câu trả lời, câu hỏi sẽ được chuyển đến DNS server để tìm câu trả lời.

– Tự tìm câu trả lời truy vấn.

Bước đầu tiên của quá trình xử lý một truy vấn. Tên miền sử dụng một chương trình trên máy tính truy vấn để tìm câu trả lời cho truy vấn. Nếu truy vấn có câu trả lời thì quá trình truy vấn kết thúc. Ngay tại máy tính truy vấn thông tin được lấy từ hai nguồn sau:

– Trong file HOSTS được cấu hình ngay tại máy tính. Các thông tin ánh xạ từ tên miền sang địa chỉ được thiết lập ở file này được sử dụng đầu tiên. Nó được tải ngay lên bộ nhớ cache của máy khi bắt đầu chạy DNS client.

– Thông tin được lấy từ các câu trả lời của truy vấn trước đó. Theo thời gian các câu trả lời truy vấn được lưu giữ trong bộ nhớ cache của máy tính và nó được sử dụng khi có một truy vấn lặp lại một tên miền trước đó.

#### • Truy vấn DNS server

Khi DNS server nhận được một truy vấn. Đầu tiên nó sẽ kiểm tra câu trả lời liệu có phải là thông tin của bản ghi mà nó quản lý trong các zone của server. Nếu truy vấn phù hợp với bản ghi mà nó quản lý thì nó sẽ sử dụng thông tin đó để trả lời (*Authoritatively answer*) và kết thúc truy vấn. Nếu không có thông tin về zone của nó phù hợp với truy vấn. Nó sẽ kiểm tra các thông tin được lưu trong cache liệu có các truy vấn tương tự nào trước đó phù hợp không nếu có thông tin phù hợp nó sẽ sử dụng thông tin đó để trả lời và kết thúc truy vấn. Nếu truy vấn không tìm thấy thông tin phù hợp để trả lời từ cả cache và zone mà DNS server quản lý thì truy vấn sẽ tiếp tục. Nó sẽ nhờ DNS server khác để trả lời truy vấn đến khi tìm được câu trả lời.

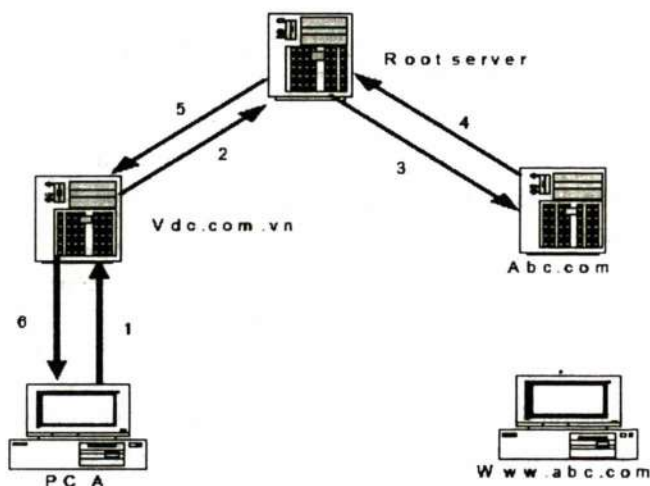
Các cách để DNS server liên lạc với nhau xác định câu trả lời.

– Trường hợp Root server kết nối trực tiếp với server tên miền cần truy vấn.

Trong trường hợp root server biết được DNS server quản lý tên miền cần truy vấn. Thì các bước của truy vấn sẽ như sau:



*Bước 1:* PC A truy vấn DNS server tên miền vdc.com.vn. (là local name server) tên miền www.abc.com.



*Hình 5.14.* Root server kết nối trực tiếp với server tên miền cần truy vấn

*Bước 2:* DNS server tên miền vdc.com.vn không quản lý tên miền www.abc.com, do vậy nó sẽ chuyển truy vấn lên root server.

*Bước 3:* Root server sẽ xác định được rằng DNS server quản lý tên miền www.abc.com là server DNS.abc.com và nó sẽ chuyển truy vấn đến DNS server DNS.abc.com để trả lời.

*Bước 4:* DNS server DNS.abc.com sẽ xác định bản ghi www.abc.com và trả lời lại root server.

*Bước 5:* Root server sẽ chuyển câu trả lời lại cho server vdc.com.vn.

*Bước 6:* DNS server vdc.com.vn sẽ chuyển câu trả lời về cho PC A và từ đó PC A có thể kết nối đến PC B (quản lý [www.abc.com](http://www.abc.com)).

– Trường hợp root server không kết nối trực tiếp với server tên miền cần truy vấn.

Trong trường hợp không kết nối trực tiếp thì root server sẽ hỏi server trung gian (phân lớp theo hình cây) để xác định được đến server tên miền quản lý tên miền cần truy vấn.



*Bước 1:* PC A truy vấn DNS server vdc.com.vn (local name server) tên miền www.acb.com.sg.

*Bước 2:* DNS server vdc.com.vn không quản lý tên miền www.abc.com.sg vậy nó sẽ chuyển lên root server.

*Bước 3:* Root server sẽ không xác định được DNS server quản lý trực tiếp tên miền www.abc.com.sg nó sẽ căn cứ vào cấu trúc của hệ thống tên miền để chuyển đến DNS quản lý cấp cao hơn của tên miền abc.com.sg đó là com.sg và nó xác định được rằng DNS server DNS.com.sg quản lý tên miền com.sg.

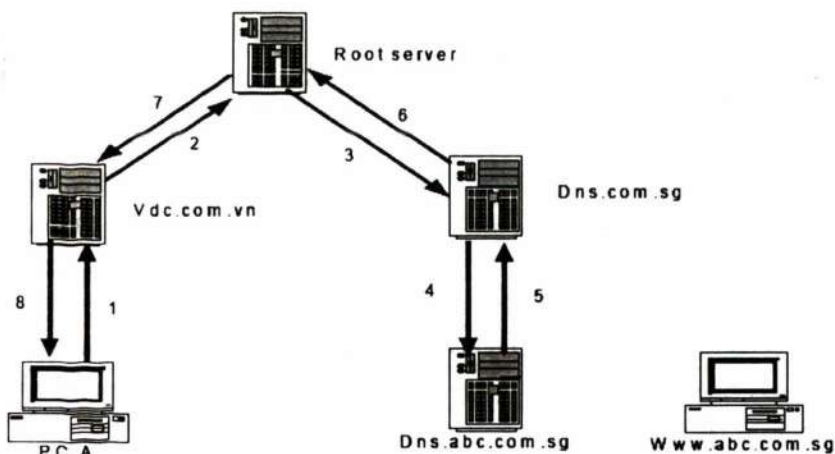
*Bước 4:* DNS.com.sg sau đó sẽ xác định được rằng DNS server DNS.abc.com.sg có quyền quản lý tên miền www.abc.com.sg.

*Bước 5:* DNS.abc.com.sg sẽ lấy bản ghi xác định cho tên miền www.abc.com.sg để trả lời DNS server DNS.com.sg.

*Bước 6:* DNS.com.sg sẽ lại chuyển câu trả lời lên root server.

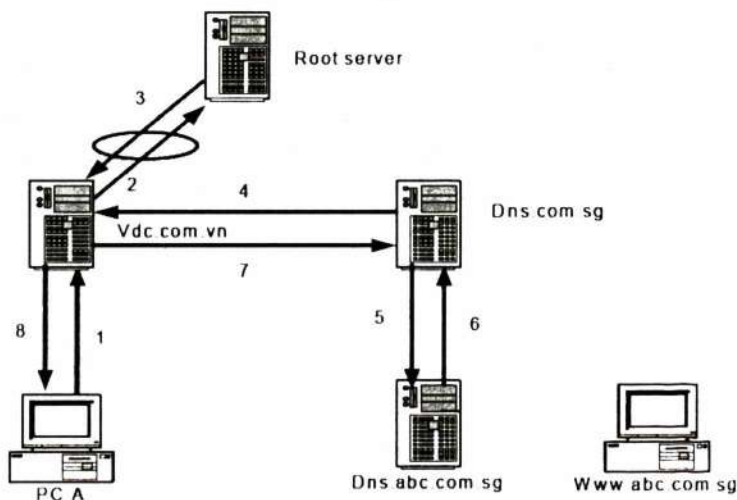
*Bước 7:* Root server sẽ chuyển câu trả lời trở lại DNS server vdc.com.vn.

*Bước 8:* DNS server vdc.com.vn sẽ trả lời về PC A câu trả lời và PC A đã kết nối được đến host quản lý tên miền www.abc.com.sg.



Hình 5.15. Root server không kết nối trực tiếp với server tên miền cần truy vấn

Khi các truy vấn lặp đi lặp lại thì hệ thống DNS có khả năng thiết lập chuyển quyền trả lời đến DNS trung gian mà không cần phải qua root server và nó cho phép thời gian truy vấn được giảm đi.



Hình 5.16. Truy vấn DNS

#### • Hoạt động của DNS cache

Khi DNS server xử lý các truy vấn của client và sử dụng các truy vấn lặp lại, nó sẽ xác định và lưu lại các thông tin quan trọng của tên miền mà client truy vấn. Thông tin đó sẽ được ghi lại trong bộ nhớ cache của DNS server. Cache lưu giữ thông tin là giải pháp hữu hiệu tăng tốc độ truy vấn thông tin cho các truy vấn thường xuyên của các tên miền hay được sử dụng và làm giảm lưu lượng thông tin truy vấn trên mạng. DNS server khi thực hiện các truy vấn đệ quy cho client thì DNS server sẽ tạm thời lưu trong cache bản ghi thông tin (*Resource Record – RR*) lấy được từ DNS server lưu trữ thông tin về truy vấn đó. Sau đó một client khác truy vấn yêu cầu thông tin của đúng bản ghi đó thì nó sẽ lấy thông tin bản ghi (RR) lưu trong cache để trả lời.

Khi thông tin được lưu trong cache, các bản ghi RR được ghi trong cache sẽ được cung cấp thời gian sống (*TTL – Time – To – Live*). Thời gian

sống của một bản ghi trong cache là thời gian mà nó tồn tại trong cache và được dùng để trả lời cho các truy vấn của client khi truy vấn tên miền trong bản ghi đó. Thời gian sống (TTL) được khai khi cấu hình cho các zone. Giá trị mặc định nhỏ nhất của thời gian sống (*Minimum TTL*) là 3600 giây (1 giờ) như giá trị này ta có thể thay đổi khi cấu hình zone. Hết thời gian sống bản ghi sẽ được xóa khỏi bộ nhớ cache.

#### 5.2.4. Dịch vụ E-mail

E-mail là một trong những ứng dụng mạng lâu đời nhất nhưng lại phổ dụng nhất. Thử nghĩ khi bạn muốn gửi thông điệp đến một người bạn ở đâu kia của thế giới, bạn muốn mang thư chạy bộ qua đó hay chỉ đơn giản lên máy tính gõ ít hàng và nhấn nút Send? Thật ra, những bậc tiền bối của mạng ARPANET đã không tiên đoán được E-mail sẽ là ứng dụng then chốt chạy trên mạng này, mục tiêu chính của họ là thiết kế hệ thống cho phép truy cập tài nguyên từ xa. Hệ thống E-mail ra đời không mấy nổi bật, nhưng đến bây giờ lại được sử dụng hằng ngày bởi hàng triệu người trên thế giới.

Mục tiêu của phần này là chỉ ra những nhân vật hoạt động trong hệ thống E-mail, vai trò của họ, giao thức mà họ sử dụng và khuôn dạng thông điệp mà họ trao đổi với nhau.

##### 5.2.4.1. Các thành phần của hệ thống E-mail

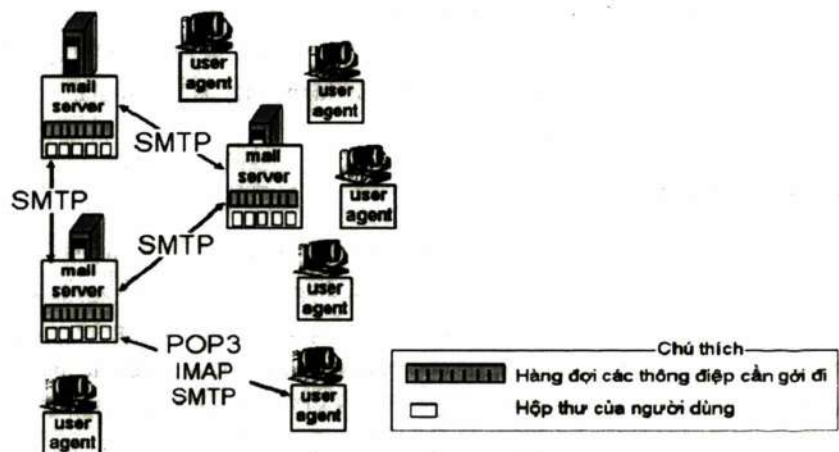
Một hệ thống E-mail thường có 3 thành phần chính: Bộ phận trợ giúp người dùng (*User Agent*), Mail Server và các giao thức mà các thành phần này dùng để giao tiếp với nhau.

Giao thức giữa các mail servers là SMTP (*Simple Mail Transfer Protocol*): Được các server dùng để chuyển thư qua lại với nhau. Ví dụ: nó giống như cách thức mà các trạm bưu điện dùng để chuyển các thùng thư của khách hàng cho nhau.

Giao thức giữa mail server và user agent bao gồm:

- POP3 (*Post Office Protocol version 3 [RFC 1939]*): Được user agent sử dụng để lấy thư về từ hộp thư của nó trên server.
- SMTP: Được user agent sử dụng để gửi thư ra server.

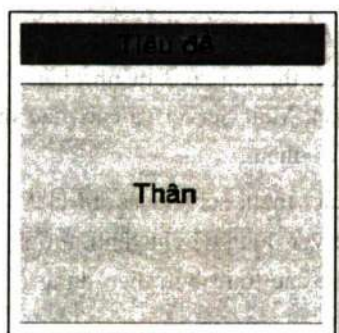
– IMAP: (*Internet Mail Access Protocol [RFC 1730]*): Có nhiều tính năng vượt trội hơn POP3. Ngoài ra, IMAP còn cho phép gửi mail.



Hình 5.17. Các thành phần của hệ thống e-mail

#### 5.2.4.2. Khuôn dạng của một e-mail

RFC 822 định nghĩa một E-mail gồm có hai phần: Phần tiêu đề (header) và phần thân (body).



Hình 5.18. Khuôn dạng của một e-mail

Cả hai phần đều được thể hiện dưới dạng ký tự ASCII. Lúc đầu, phần thân được quy định có khuôn dạng văn bản đơn giản. Sau này người ta để

ngiht một chuẩn mới gọi là MIME, có thể cho phép phần thân của E-mail chứa bất kỳ loại dữ liệu nào.

Phần tiêu đề bao gồm nhiều dòng thông tin, mỗi dòng kết thúc bằng hai ký tự <CRLF>. Phần tiêu đề được chia khối phần thân bởi một hàng rỗng. Mỗi một hàng tiêu đề chứa một cặp "tên" và "giá trị", cách nhau bởi dấu hai chấm (:). Người dùng có thể rất quen với nhiều hàng tiêu đề vì họ thường phải điền thông tin vào đó.

*Ví dụ:*

**Bảng 5.8.** Tên và giá trị của E-mail phiên bản đơn giản

Tên	Giá trị
From	Địa chỉ người gửi
To	Địa chỉ của người nhận
Subject:	Chủ đề thư
Date	Ngày gửi

RFC 822 được mở rộng năm 1993 (và được cập nhật lại năm 1996) để cho phép E-mail mang được nhiều loại dữ liệu: audio, video, hình ảnh, tài liệu Word... MIME (Multipurpose Internet Mail Extensions) về cơ bản có ba phần. Phần đầu tiên là tập các dòng header dùng để bổ túc cho phần header cũ của RFC 822. Theo nhiều cách, những dòng header này mô tả dữ liệu chứa trong phần thân.

Cụ thể như sau:

**Bảng 5.9.** Tên và giá trị của E-mail phiên bản mở rộng

Tên	Giá trị
MIME-Version:	Phiên bản MIME đang sử dụng
Content-Description:	Mô tả trong thư đang có dữ liệu gì
Content-Type:	Mô tả kiểu dữ liệu đang nằm trong thư
Content-Transfer-Encoding:	Mô tả cách thức mã hoá dữ liệu trong thư



Phần thứ hai là các định nghĩa cho một tập các kiểu nội dung (và kiểu con nếu có). Ví dụ một số kiểu mà MIME định nghĩa:

**Bảng 5.10.** Bảng một số kiểu trong E-mail

Kiểu	Ý nghĩa
image/gif	Ảnh dạng gif
image/jpeg	Ảnh dạng jpeg
text/plain	Văn bản đơn giản
text/richtext	Văn bản mở rộng (có đặt font chữ, được định dạng đậm, nghiêng hoặc gạch dưới...)
application	Dữ liệu trong thư được xuất ra từ một ứng dụng nào đó. Chẳng hạn: application/postscript: tài liệu Postscript (.ps) application/msword: tài liệu Microsoft Word (.doc)

MIME cũng định nghĩa kiểu multipart để chỉ ra cách mà phần thân của thư mang nhiều loại dữ liệu khác nhau như thế nào. Chỉ có một kiểu con của multipart là mixed với ý nói rằng trong phần thân của thư có nhiều mảnh dữ liệu khác nhau, độc lập với nhau và được sắp xếp theo một trình tự cụ thể. Mỗi mảnh dữ liệu sẽ có phần tiêu đề riêng để mô tả kiểu dữ liệu của mảnh đó. Phần thứ ba mô tả cách thức mã hoá các kiểu dữ liệu nói trên để có thể truyền chúng dưới dạng ASCII.

Lý do để mọi bức thư phải chứa các ký tự ASCII là vì để đi được đến đích, bức thư đó có thể phải trung chuyển qua nhiều gateway, mà các gateway này đều coi mọi bức thư dưới dạng ASCII. Nếu trong thư chứa bất kỳ ký tự nào khác ASCII thì thư sẽ bị đứt gãy nội dung. MIME sử dụng phương pháp mã hoá trực tiếp dữ liệu nhị phân thành các ký tự nhị phân, gọi là base64. Ý tưởng của base64 là ánh xạ 3 bytes dữ liệu nhị phân nguyên thủy thành 4 ký tự ASCII.

Giải thuật đơn giản như sau: Tập hợp 3 bytes dữ liệu nhị phân lại thành 24 bit, sau đó chia 24 bit này thành 4 cụm, một cụm 6 bit. Một cụm 6 bit được ánh xạ vào một trong 64 ký tự ASCII hợp lệ; ví dụ 0 ánh xạ thành A, 1 ánh xạ thành B... Nếu nhìn vào bức thư đã được mã hoá dạng base64,

người dùng sẽ thấy chỉ có 52 chữ cái cả hoa lẫn thường, 10 chữ số từ 0 đến 9 và các ký tự đặc biệt + và/.

Đối với những người dùng chỉ sử dụng trình đọc thư hỗ trợ duy nhất kiểu ký tự thì việc đọc những bức thư có kiểu base64 sẽ rất khó khăn. Vì lý do nhân đạo, MIME còn hỗ trợ kiểu mã hoá ký tự thường được gọi là 7-bit. 7-bit sẽ giữ nguyên dạng ký tự mà ta nhập vào.

#### 5.2.4.3. Chuyển thư

Tiếp đến, ta xem xét giao thức SMTP – giao thức được dùng để chuyển thư từ máy này đến máy kia. Để đặt SMTP vào đúng ngữ cảnh, chúng ta nên nhắc lại các nhân vật then chốt trong hệ thống E-mail.

*Thứ nhất*, người dùng tương tác với trình đọc thư (hay còn gọi là user agent) để soạn, lưu, tìm kiếm và đọc thư của họ. Hiện nay trên thị trường có nhiều phần mềm đọc thư, giống như hiện cũng đang có nhiều loại trình duyệt Web vậy.

*Thứ hai*, có trình xử lý thư (hay còn gọi là mail server) chạy trên một máy nào đó trong mạng nội bộ của người dùng. Có thể xem mail server như một bưu điện: Người dùng trao cho mail server các bức thư mà họ muốn gửi cho người dùng khác, mail server sử dụng giao thức SMTP trên TCP để chuyển các thư này đến mail server bên đích. Mail server bên đích nhận các thư đến và đặt chúng vào hộp thư của người dùng bên đích. Do SMTP là giao thức mà rất nhiều người có thể tự cài đặt, vì thế sẽ có nhiều sản phẩm mail server hiện có trên thị trường. Sản phẩm mail server thường được sử dụng nhất là sendmail, ban đầu được cài đặt trong hệ điều hành Berkeley Unix.

Tất nhiên mail server bên máy gửi có thể kết nối SMTP/TCP trực tiếp tới mail server bên máy nhận, nhưng trong thực tế, một bức thư có thể đi ngang qua vài mail gateways trước khi đến đích. Cũng giống như máy đích, mỗi mail gateway cũng chạy một mail server. Không phải ngẫu nhiên mà các nút chuyển thư trung gian được gọi là mail gateway. Công việc của chúng cũng giống như các IP gateway là lưu tạm và chuyển phát tiếp các bức thư của người dùng. Điểm khác nhau duy nhất giữa chúng là, mail gateway trữ tạm các bức thư trong đĩa, trong khi các IP gateway trữ tạm các gói tin IP trong bộ nhớ.

Có thể đặt câu hỏi: tại sao lại cần đến các mail gateways? Tại sao không dùng phương pháp nối kết SMTP/TCP trực tiếp từ bên gửi sang bên nhận? Lý do thứ nhất, người gửi không muốn kèm trong thư địa chỉ của máy đích. Thứ hai, không chắc lúc bên gửi thiết lập nối kết đến bên nhận, người dùng bên nhận đã bật sẵn máy! Do đó chỉ cần địa chỉ thư bên nhận là đủ.

Dù có bao nhiêu mail gateways trung gian trên đường đến đích vẫn không đáng lo lắng, bởi mỗi mail gateway trung gian sẽ nỗ lực sử dụng một kết nối SMTP độc lập đến gateway kế tiếp trên đường đi nhằm chuyển thư càng ngày càng đến gần người nhận.

SMTP là một giao thức đơn giản dùng các ký tự ASCII. Sau khi thiết lập nối kết TCP đến cổng 25 của máy đích (được coi là server), máy nguồn (được coi là client) chờ nhận kết quả trả về từ server. Server khởi đầu cuộc đối thoại bằng cách gửi một dòng văn bản đến client thông báo danh tính của nó và khả năng tiếp nhận thư. Nếu server không có khả năng nhận thư tại thời điểm hiện tại, client sẽ huỷ bỏ nối kết và thử thiết lập lại nối kết sau.

Nếu server sẵn sàng nhận thư, client sẽ thông báo thư đó từ đâu đến và ai sẽ là người nhận. Nếu người nhận đó tồn tại, server sẽ thông báo cho client tiếp tục gửi thư. Sau đó client gửi thư và server báo nhận cho thư đó. Sau khi cả hai bên hoàn tất phiên truyền nhận, kết nối sẽ được đóng lại

**Bảng 5.11. Bảng thông báo gửi và nhận thư**

LỆNH CỦA CLIENT	
HELO	Câu chào và xưng danh của client
MAIL FROM	Địa chỉ mail của người gửi
RCPT To	Địa chỉ mail của người nhận
DATA	Bắt đầu truyền nội dung của thư
QUIT	Huỷ nối liên kết
TRẢ LỜI CỦA CLIENT	
250	Yêu cầu hợp lệ
550	Yêu cầu không hợp lệ, không tồn tại hộp thư như client đã chỉ ra
354	Cho phép bắt đầu nhập thư vào. Kết thúc thư bằng <CRLF>.<CRLF>
221	Sever đang đóng kết nối TCP

#### 5.2.4.4. Phân phát thư

Khi đứng về góc độ người dùng thư, họ sẽ dùng user agent để gửi và nhận thư. User agent dùng giao thức SMTP để gửi thư đi, dùng giao thức POP3 hoặc IMAP để nhận thư về.

##### • POP3

Một phiên làm việc theo giao thức POP3 bắt đầu tại user agent. User agent khởi động một nối kết TCP đến cổng 110 của mail server. Khi kết nối thực hiện xong, phiên làm việc POP3 sẽ trải qua theo thứ tự ba kỳ:

- Chứng thực.
- Giao dịch dữ liệu.
- Cập nhật.

Kỳ chứng thực buộc người dùng thực hiện thủ tục đăng nhập bằng cách nhập vào hai lệnh sau:

**Bảng 5.12.** Bảng lệnh khai báo dùng cho người dùng E-mail

Lệnh	Ý nghĩa
USER <tên người dùng>	Khai báo tên người dùng
PASS <mật khẩu>	Khai báo mật khẩu

Báo trả của mail server sẽ là một trong hai câu sau:

**Bảng 5.13.** Bảng lệnh báo của mail sever

Trả lời	Ý nghĩa
+OK <chú thích>	Khai báo của người dùng là đúng
-ERR <chú thích>	Khai báo của người dùng là sai và lời giải thích

Trong kỳ giao dịch, người dùng có thể xem danh sách thư chưa nhận về, nhận thư về và xoá thư trong hộp thư của mình khi cần thiết. Các lệnh mà người dùng thường sử dụng để giao dịch với server là:



**Bảng 5.14.** Bảng lệnh dùng để giao dịch giữa người dùng và mail sever

Lệnh	Ý nghĩa
LIST [<số thứ tự thư>]	Nếu dùng LIST không tham số, server sẽ trả về toàn bộ danh sách các thư chưa nhận. Nếu có tham số là số thứ tự thư cụ thể, server sẽ trả về thông tin của chỉ bức thư đó
RETR <số thứ tự thư>	Tải lá thư có số thứ tự <số thứ tự thư> về
DELE <số thứ tự thư>	Xoá lá thư số <số thứ tự thư> khỏi hộp thư
Quit	Hoàn tất giai đoạn giao dịch và huỷ nối kết TCP

Các trả lời của server có thể là các số liệu mà client yêu cầu hoặc các thông báo +OK, -ERR như trong phần đăng nhập.

#### • IMAP (*Internet Message Access Protocol*)

Với những người dùng có một tài khoản E-mail trên một ISP và người dùng này thường truy cập E-mail trên một PC thì giao thức POP3 hoạt động tốt. Tuy nhiên, một sự thật trong ngành công nghệ máy tính, khi một thứ gì đó đã hoạt động tốt, người ta lập tức đòi hỏi thêm nhiều tính năng mới. Điều đó cũng xảy ra đối với hệ thống E-mail.

*Ví dụ:* Người ta chỉ có một tài khoản E-mail, nhưng họ lại muốn ngồi đâu cũng truy cập được nó. POP3 cũng làm được việc này bằng cách đơn giản tải hết các E-mail xuống máy PC mà người dùng này đang ngồi làm việc. Và dĩ nhiên là thư từ của người dùng này nằm rải rác khắp nơi. Sự bất tiện này khơi mào cho sự ra đời của giao thức phân phối thư mới, IMAP (*Internet Message Access Protocol*), được định nghĩa trong RFC 2060.

Không giống như POP2, IMAP coi các thông điệp mặc nhiên nằm trên server vô hạn và trên nhiều hộp thư. IMAP còn đưa ra cơ chế cho phép đọc các thông điệp hoặc một phần của thông điệp, một tính năng hữu ích khi người dùng kết nối đến server bằng đường truyền tốc độ chậm như điện thoại nhưng lại đọc các E-mail có âm thanh, hình ảnh... Với quan niệm cho rằng người dùng không cần tải thư về lưu trên PC, IMAP cung cấp các cơ chế cho phép tạo, xoá và sửa đổi nhiều hộp thư trên server. Cung cách làm



việc của IMAP cũng giống như POP3, ngoại trừ trong IMAP có rất nhiều lệnh. IMAP server sẽ lắng nghe trên cổng 143. Cũng nên chú ý rằng, không phải mọi ISP đều hỗ trợ cả hai giao thức POP3 và IMAP.

**Bảng 5.15.** Bảng so sánh các tính năng của POP3 và IMAP

Tính năng	POP3	IMAP
Giao thức được định nghĩa ở đâu?	RFC 1939	RFC 2060
Cổng TCP được dùng	110	143
E-mail được lưu ở đâu	PC của người dùng	Sever
E-mail được đọc ở đâu	Off-line	On-line
Thời gian nối kết	ít	Nhiều
Sử dụng tài nguyên của server	Tối thiểu	Nhiều hơn
Nhiều hộp thư	Không	Đúng
Ai lưu phông hờ các hộp thư	Người dùng	ISP
Tốt cho người dùng di động	không	có
Kiểm soát của người dùng đối với việc tải thư về.	ít	Tốt
Tải một phần thư	Không	có
Quota đĩa có là vấn đề không?	Không	Thỉnh thoảng
Dễ cài đặt	Có	Không
Được hỗ trợ rộng rãi	Có	Đang phát triển

## CÂU HỎI VÀ BÀI TẬP

1. Hãy nêu giao thức lớp ứng dụng.
2. Trình bày giao thức HTTP và dịch vụ WWW. Cho ví dụ minh họa.
3. Trình bày giao thức FTP và dịch vụ truyền file trên Internet.
4. Trình bày mô hình giao thức gửi và nhận thư điện tử.
5. Cài đặt DNS Server cho Window 2003.

6. Cài đặt dịch vụ E-mail sever.
7. Cài đặt dịch vụ FTP và Web sever.
8. Bạn hiểu thế nào là quá trình truyền thông trên mạng?
9. Trình bày cách thức chuyển thư trong dịch vụ E-mail.
10. Nêu khuôn dạng của một E-mail.
11. Phân tích các thành phần của E-mail.
12. Trình bày giao thức http.
13. Nêu các lệnh cơ bản trong FTP.
14. Trình bày dịch vụ (Service) lớp ứng dụng.

# Chương 6

## LỚP TRUYỀN TẢI

### MỤC TIÊU

Sau khi học xong chương này người học phải có khả năng sau:

- ✓ Trình bày được nguyên lý cơ bản trong lớp truyền tải, phương thức truyền dẫn có kết nối UDP và phương thức truyền dẫn không kết nối TCP.
- ✓ Phân biệt được phương thức truyền dẫn có kết nối UDP và phương thức truyền dẫn không kết nối TCP.
- ✓ Phát huy được khả năng tư duy sáng tạo.

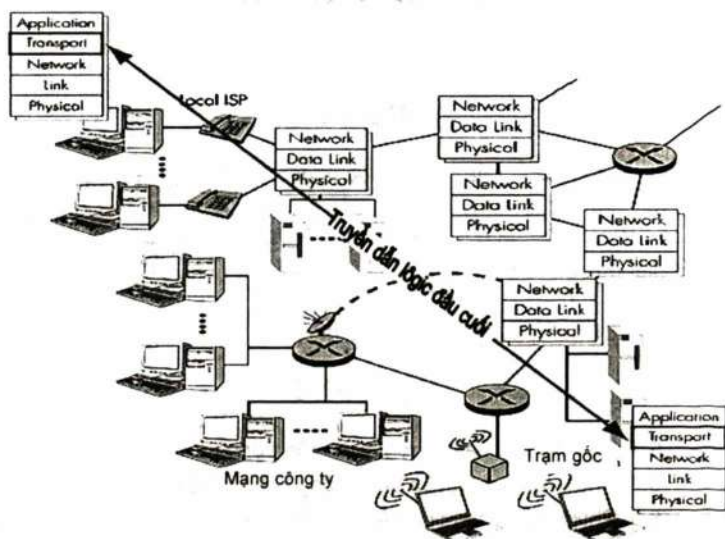
### NỘI DUNG

- 6.1. Các dịch vụ và nguyên lý cơ bản trong lớp truyền tải
  - 6.2. Phương thức truyền dẫn không kết nối UDP
  - 6.3. Phương thức truyền dẫn có kết nối TCP
  - 6.4. Điều khiển luồng trong TCP
  - 6.5. Điều khiển lỗi
- Câu hỏi và bài tập.

## 6.1. CÁC DỊCH VỤ VÀ NGUYÊN LÝ CƠ BẢN TRONG LỚP TRUYỀN TẢI

Lớp truyền dẫn làm nhiệm vụ thiết lập, duy trì và hủy bỏ các cuộc giao tiếp giữa hai máy, đảm bảo việc truyền dữ liệu truyền giống hoàn toàn dữ liệu nhận. Dữ liệu qua các mạng con có thể bị lỗi, tập tin lớp truyền dẫn thực hiện cải thiện chất lượng dịch vụ, đảm bảo dữ liệu được truyền một cách chính xác và truyền lại nếu như phát hiện thấy lỗi. Lớp dữ liệu quản lý việc gửi, xác định thứ tự dữ liệu, độ ưu tiên của dữ liệu đó. Giao thức lớp

truyền dẫn cung cấp quá trình truyền thông logic giữa các tiến trình ứng dụng trên các hệ thống chủ khác nhau. Các tiến trình ứng dụng sử dụng đường truyền logic để gửi các thông tin cho nhau mà không cần quan tâm đến các chi tiết của việc truyền các thông tin này trên các đường truyền vật lý.



**Hình 6.1.** Lớp truyền dẫn cung cấp các đường truyền logic giữa các ứng dụng

Trên hình mô tả giao thức truyền dẫn chạy trên các hệ thống đầu - cuối nhưng không chạy trên các thiết bị chuyên mạch. Router chỉ tác động đến các trường trong PDU lớp 3. Bên phía phát, lớp truyền dẫn chuyển đổi dữ liệu nhận được thành PDU lớp 4. Điều này được thực hiện bằng cách chia nhỏ dữ liệu lớp ứng dụng thành các khối tin nhỏ hơn và thêm vào các thông tin đầu trang để tạo thành các PDU.

Lớp truyền dẫn sau đó chuyển PDU lớp 4 tới lớp mạng (network). Bên phía nhận, lớp truyền dẫn nhận PDU lớp 4 từ lớp mạng, loại bỏ header, kết hợp các thông tin lại thành 1 gói tin và chuyển tới tiến trình ứng dụng phía nhận. Mạng máy tính có thể sử dụng một hoặc nhiều các giao thức lớp truyền dẫn trên các ứng dụng mạng. Ví dụ Internet có hai giao thức - TCP và UDP. Mỗi giao thức cung cấp một số các dịch vụ nhất định cho ứng dụng.

### 6.1.1. Tổng quan lớp truyền dẫn trong mạng Internet

Trong mạng Internet, nơi sử dụng mô hình mạng TCP/IP, sử dụng hai giao thức lớp truyền dẫn khác nhau là UDP (*User Datagram Protocol*) và TCP (*Transmission Control Protocol*). Khi thiết kế các chương trình ứng dụng, người thiết kế ứng dụng phải sử dụng một trong hai giao thức truyền dẫn này. Chúng ta biết rằng đơn vị dữ liệu giao thức (PDU) lớp 4 là segment, tuy nhiên trong các tài liệu về Internet (như RFCs) thường gọi PDU của TCP là segment còn PDU của UDP là datagram. Tuy vậy trong các tài liệu của Internet cũng sử dụng khái niệm datagram cho PDU lớp network.

Trong tài liệu này chúng ta sử dụng chung khái niệm segment cho PDU của cả TCP và UDP. Giao thức lớp network là IP – *Internet Protocol*. IP cung cấp đường truyền logic giữa hai hệ thống. Dịch vụ của giao thức IP là best-effort. Điều này có nghĩa là IP sẽ cố gắng hết mức để chuyển các đoạn dữ liệu đi nhưng không có sự đảm bảo trong việc truyền dữ liệu. Trong thực tế, IP không đảm bảo chuyển đoạn dữ liệu, không đảm bảo thứ tự các đoạn dữ liệu cũng như tính chính xác của dữ liệu bên trong đoạn dữ liệu. Chính vì vậy IP được coi là dịch vụ không tin cậy (*Unreliable service*). Chúng ta cũng đều biết được rằng tất cả các hệ thống đầu - cuối đều có một địa chỉ IP duy nhất.

Nhiệm vụ cơ bản của UDP và TCP là mở rộng việc chuyển tin giữa hai hệ thống đầu - cuối thành các dịch vụ chuyển dữ liệu giữa hai tiến trình chạy trên các hệ thống đầu - cuối khác nhau, được gọi là application multiplexing và demultiplexing. TCP và UDP cùng thực hiện việc kiểm tra, phát hiện lỗi để đảm bảo tính chính xác. Trong thực tế UDP cũng như IP là các dịch vụ không tin cậy: không đảm bảo dữ liệu gửi từ một tiến trình đến được chính xác một tiến trình đích. TCP thì ngược lại, cung cấp thêm nhiều dịch vụ cho lớp ứng dụng. Đầu tiên và trước nhất là dịch vụ truyền dữ liệu tin cậy. Sử dụng việc điều khiển luồng, đánh số thứ tự, thông báo phản hồi (Ack) và sử dụng bộ định thời. TCP đảm bảo rằng dữ liệu được chuyển từ tiến trình truyền đến tiến trình nhận một cách chính xác và đúng thứ tự.

Như vậy, TCP đã biến dịch vụ IP không tin cậy giữa các hệ thống đầu - cuối thành dịch vụ truyền dữ liệu tin cậy giữa các tiến trình. TCP còn sử

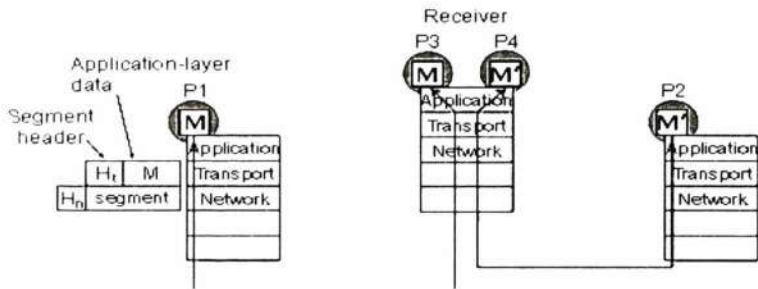


dụng điều khiển tắc nghẽn, nó ngăn chặn bất kỳ một kết nối TCP nào mà làm cho đường truyền trở lên quá tải và thực hiện chuyển các quá trình truyền thông giữa hai hệ thống bằng cách kiểm tra lưu lượng kết nối. Trong thực tế, TCP cho phép các kết nối cùng được chia sẻ băng thông trên cùng đường truyền. Điều này được thực hiện bằng cách quy định tốc độ và lưu lượng truyền ở phía phát. Ngược lại, lưu lượng của UDP không được kiểm soát, do đó các ứng dụng sử dụng UDP có thể truyền dữ liệu tại bất kỳ tốc độ nào mà đường truyền cho phép.

### 6.1.2. Dồn kênh, phân kênh

Trong phần này chúng ta mô tả về các ứng dụng mạng sử dụng ghép kênh và phân kênh. Phải nhấn mạnh lại là dồn kênh và phân kênh là các ứng dụng cần thiết cho tất cả các mạng máy tính. Tại phía máy đích, lớp truyền dẫn nhận đoạn dữ liệu từ lớp mạng phía dưới. Lớp truyền dẫn sau đó thực hiện chuyển dữ liệu trong các đoạn tới các tiến trình trong các ứng dụng tương ứng trên hệ thống.

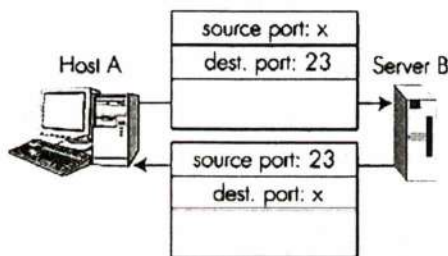
Giả thiết bạn đang ngồi trước máy tính và bạn đang tải một trang Web trong khi đang chạy một phiên FTP và hai phiên Telnet. Do đó bạn có tất cả bốn tiến trình ứng dụng: hai tiến trình telnet, một tiến trình FTP và một tiến trình HTTP. Khi lớp truyền dẫn trên máy tính nhận dữ liệu từ lớp mạng phía dưới, nó cần phải chuyển hướng dữ liệu nhận được tới một trong số bốn tiến trình trên. Bây giờ chúng ta xem xét xem việc chuyển hướng diễn ra như thế nào. Mỗi đoạn dữ liệu lớp truyền dẫn đều có một trường thông tin xác định tiến trình nào sẽ được truyền dữ liệu tới. Ở đầu - cuối phía thu, lớp truyền dẫn có thể kiểm tra các trường này để quyết định tiến trình nhận và chuyển hướng đoạn dữ liệu đó tới tiến trình tương ứng. Việc chuyển dữ liệu trong các đoạn trong lớp truyền dẫn tới tiến trình trong các ứng dụng tương ứng được gọi là quá trình phân kênh. Việc thu thập dữ liệu ở phía máy nguồn từ các tiến trình ứng dụng khác nhau, đóng gói dữ liệu với các thông tin thêm vào để tạo ra các phân đoạn và chuyển các phân đoạn đó đến lớp mạng phía dưới thì được gọi là quá trình ghép kênh. Ghép kênh và phân kênh được mô tả trong hình 6.2.



Hình 6.2. Ghép kênh và phân kênh

UDP và TCP thực hiện phân kênh và ghép kênh bằng cả hai trường thông tin trong header của đoạn dữ liệu: số hiệu cổng nguồn và số hiệu cổng đích. Hai trường này được minh họa trên hình 6.3. Khi cùng làm việc, các trường xác định duy nhất một tiến trình ứng dụng chạy trên máy đích.

Số hiệu cổng là một số 16 bit có giá trị trong khoảng từ 0 tới 65535. Các số hiệu cổng thay đổi từ 0 tới 1023 được gọi là các cổng "well known - thông dụng" và là các cổng dành riêng, có nghĩa là chúng được sử dụng cho các giao thức lớp ứng dụng thông dụng như HTTP, FTP.



Hình 6.3. Trường số hiệu cổng nguồn và cổng đích trong đoạn dữ liệu

Số hiệu cổng của HTTP là 80, FTP là 21. Danh sách số hiệu các cổng thông dụng được mô tả trong chuẩn RFC 1700. Khi phát triển một ứng dụng mới, chúng ta phải gắn ứng dụng đó với một cổng nhất định. Đối với mỗi loại ứng dụng khác nhau chạy trên các hệ thống đầu - cuối có một số hiệu cổng duy nhất, như vậy là tại sao đoạn dữ liệu trong lớp ứng dụng lại có hai trường thông tin biểu diễn số hiệu cổng nguồn và cổng đích? Câu trả lời rất đơn giản đó là: một hệ thống đầu - cuối có thể chạy hai tiến trình cùng loại

trong cùng một thời điểm, do đó số hiệu cổng đích của một ứng dụng không thể xác định được một tiến trình nhất định.

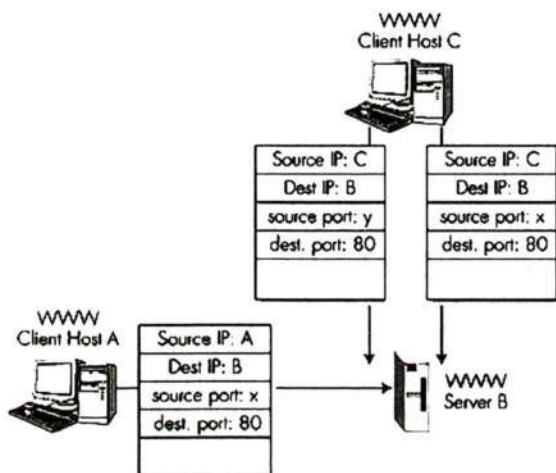
*Ví dụ:* Web server có thể sinh ra một tiến trình cho tất cả các yêu cầu xử lý khi server nhận được nhiều hơn một yêu cầu, khi đó server chạy nhiều hơn một tiến trình với cổng là 80. Do đó để xác định được một tiến trình duy nhất mà dữ liệu đã được định trước thì đòi hỏi phải có hai số hiệu cổng.

Như vậy cách tạo ra số hiệu cổng thứ hai như thế nào? Cổng nào là cổng đi ra trong trường thông tin cổng của đoạn dữ liệu? Cổng nào là trường thông tin cổng đích của đoạn tin? Chúng ta biết rằng các ứng dụng mạng thường tổ chức dưới dạng chủ/khách. Thông thường một hệ thống chủ bắt đầu ứng dụng là các máy khách. Giả thiết có một ứng dụng sử dụng cổng 23 (Telnet), đoạn dữ liệu nằm trên máy khách (trên hệ thống bắt đầu phiên làm việc telnet) và được gửi đến server. Như vậy đâu là số hiệu cổng nguồn và đâu là số hiệu cổng đích cho đoạn dữ liệu này? Đối với trường hợp số hiệu cổng đích, đoạn dữ liệu này có số hiệu cổng đích là 23. Đối với số hiệu cổng nguồn, máy khách sử dụng một số mà không được sử dụng bởi bất kỳ một tiến trình khác trên máy (phần mềm lớp truyền dẫn tự động thực hiện việc lựa chọn số hiệu cổng và chuyển đến các nhà phát triển ứng dụng). Giả thiết máy khách lựa chọn số hiệu cổng là X. Khi đó mỗi đoạn thông tin mà tiến trình telnet gửi đến server đều có số hiệu cổng nguồn là X và số hiệu cổng đích là 23. Khi đoạn tin đến server, số hiệu cổng nguồn và đích cho phép server chuyển dữ liệu trong đoạn tin đó tới đúng tiến trình ứng dụng. Số hiệu cổng đích 23 xác định tiến trình Telnet và số hiệu cổng nguồn là X xác định một tiến trình Telnet nhất định. Trong trường hợp các đoạn dữ liệu chuyển từ server tới máy khách. Số hiệu cổng nguồn là 23 và số hiệu cổng đích bây giờ là X. Khi đoạn dữ liệu tới đích, số hiệu cổng nguồn và cổng đích được client sử dụng để chuyển dữ liệu từ đoạn dữ liệu tới tiến trình dữ liệu tương ứng mà được xác định bằng cặp số hiệu cổng. Hình 6.4 minh họa cho quá trình này.

Bây giờ chắc bạn thắc mắc nếu hai client thiết lập phiên làm việc tới server mà chúng sử dụng chung số hiệu cổng X thì chuyện gì sẽ xảy ra? Điều này xảy ra khi Web server đáp ứng nhiều yêu cầu trang Web đồng



thời. Như vậy làm thế nào để server có thể phân kênh các đoạn dữ liệu khi hai phiên làm việc có cùng cặp số hiệu cổng? Ở đây, server cũng sử dụng địa chỉ IP trong gói tin IP mà mang đoạn dữ liệu đó. Tình huống này được minh họa trên hình 6.4, máy C bắt đầu hai phiên làm việc HTTP với server B và máy A bắt đầu một phiên HTTP với B. A, C và B đều có địa chỉ IP duy nhất. Máy C sử dụng hai cổng nguồn là X và Y cho việc thực hiện kết nối HTTP tới B. Do A cũng chọn số hiệu cổng nguồn và hoàn toàn độc lập với máy C, vì vậy A có thể chọn số hiệu cổng nguồn là X cho kết nối HTTP. Tuy nhiên server B vẫn có thể phân kênh một cách chính xác do hai kết nối có các địa chỉ IP nguồn khác nhau.



*Hình 6.4. Sử dụng số hiệu cổng nguồn/đích trên các ứng dụng client/server 186*

Như vậy, ta đã xem xét cách thức lớp truyền dẫn có thể dồn kênh và phân kênh trong các ứng dụng mạng.

## 6.2. PHƯƠNG THỨC TRUYỀN DẪN KHÔNG KẾT NỐI UDP

UDP được định nghĩa trong chuẩn RFC 768. Ngoài chức năng ghép kênh/phân kênh và kiểm tra lỗi UDP không cung cấp thêm một dịch vụ nào khác. Thực tế nếu một nhà phát triển ứng dụng chọn UDP thay cho TCP có

nghĩa là ứng dụng đó chủ yếu làm việc với IP. UDP lấy thông tin từ tiến trình ứng dụng, gắn vào số hiệu cổng nguồn và cổng đích cho việc ghép kênh/dồn kênh và thêm vào hai trường thông tin nhỏ khác và chuyển đoạn dữ liệu kết quả đến lớp mạng. Lớp mạng đóng gói các đoạn dữ liệu vào trong một đơn vị dữ liệu IP và cố gắng nhất để chuyển các đoạn dữ liệu đó đến phía máy nhận. Nếu các đoạn dữ liệu đến được phía thiết bị nhận, UDP sử dụng số hiệu cổng và địa chỉ IP đích để chuyển dữ liệu trong các đoạn đến đúng tiến trình tương ứng.

Chú ý UDP không sử dụng quá trình bắt tay trước khi truyền dữ liệu, vì vậy UDP được gọi là phương pháp không kết nối. DNS là một giao thức lớp ứng dụng sử dụng UDP. Như vậy câu hỏi đặt ra đó là tại sao một nhà phát triển ứng dụng lại lựa chọn cách xây dựng chương trình chạy trên nền UDP hơn là TCP. Có phải TCP luôn tốt hơn UDP do TCP cung cấp dịch vụ truyền file tin cậy trong khi UDP thì không? Câu trả lời là không, nhiều ứng dụng sử dụng UDP thì phù hợp hơn bởi nhiều lý do.

Không phải thiết lập kết nối. Như chúng ta đã biết, TCP sử dụng phương pháp bắt tay ba bước trước khi bắt đầu truyền dữ liệu. UDP luôn bắt đầu truyền mà không cần bất kỳ sự chuẩn bị nào về đường truyền do đó UDP không có thời gian trễ để thiết lập đường truyền. Đó có thể là lý do chính để DNS sử dụng UDP chứ không phải là TCP.

Không có trạng thái kết nối. TCP duy trì trạng thái kết nối trên các hệ thống đầu - cuối. Thông tin trạng thái này bao gồm bộ đệm thu và bộ đệm nhận, các thông số của điều khiển tắc nghẽn, chuỗi số và các thông báo. UDP thì ngược lại không duy trì trạng thái kết nối và không theo dõi bất kỳ một thông số nào. Vì vậy mà một server thường hỗ trợ cho nhiều các máy khách chạy trên UDP nhiều hơn so với trường hợp chạy TCP.

Thông tin phân đầu gói tin thêm vào nhỏ hơn. Đoạn dữ liệu TCP sử dụng 20 byte để làm header trong khi UDP chỉ sử dụng có 8 byte. Tốc độ truyền không được kiểm soát. TCP có cơ chế điều khiển luồng làm giảm tốc độ truyền của phía phát khi một hoặc một số đường truyền giữa bên phát và bên nhận bắt đầu xuất hiện hiện tượng tắc nghẽn. Điều này ảnh hưởng đến một vài ứng dụng thời gian thực, những ứng dụng có thể chấp nhận tỷ lệ



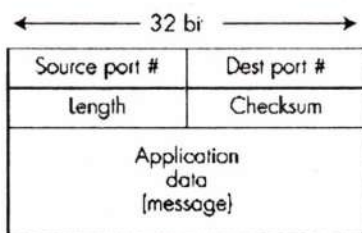
mất gói nhất định nhưng lại yêu cầu một tốc độ truyền nhất định. Ngược lại, tốc độ truyền dẫn của UDP phụ thuộc tốc độ truyền dữ liệu của phía phát, dung lượng của thiết bị nguồn (CPU, tốc độ đồng hồ...) và băng thông cho phép của kết nối.

Mặc dù hiện nay các ứng dụng đa phương tiện chạy trên UDP nhưng chúng có thể có tranh chấp do UDP không có cơ chế điều khiển luồng để đảm bảo rằng mạng không rơi vào tình trạng chỉ có một số ít các đường truyền là hoạt động. Nếu mọi người đều bắt đầu luồng video tốc độ cao mà không sử dụng bất kỳ một cơ chế điều khiển luồng nào thì có thể dẫn đến tình trạng có quá nhiều các gói tin đến router và không thể có một ai có thể làm việc được.

Do đó việc thiếu cơ chế điều khiển tắc nghẽn trong UDP có thể là một vấn đề nghiêm trọng. Nhiều nhà nghiên cứu đưa ra nhiều cơ chế mới để bắt buộc mọi nguồn phát kể cả nguồn UDP thực hiện việc điều khiển tắc nghẽn thích ứng. Trước khi mô tả về cấu trúc khung đoạn dữ liệu UDP chúng ta đề cập đến khả năng để các ứng dụng có thể truyền thông tin cậy khi sử dụng UDP. Điều này có thể được thực hiện khi khả năng tin cậy được thiết kế sẵn trong các ứng dụng.

### 6.2.1. Cấu trúc đoạn dữ liệu UDP

Cấu trúc đoạn dữ liệu được mô tả trong chuẩn RFC 768. Dữ liệu lớp ứng dụng có trong trường dữ liệu của gói tin UDP. Header của UDP chỉ có bốn trường, mỗi trường chiếm 2 byte. Số hiệu cổng cho phép thiết bị đích có thể chuyển dữ liệu ứng dụng tới đúng tiến trình chạy trên thiết bị đích. Trường checksum (kiểm tra tổng) được bên thu sử dụng để kiểm tra xem có lỗi hay không. Trên thực tế, kiểm tra tổng được tính toán dựa vào một vài trường thông tin trong khung IP được thêm vào



Hình 6.5. UDP segment structure

đoạn dữ liệu UDP. Trường độ dài chỉ ra chiều dài của đoạn dữ liệu bao gồm cả header.

- SourcePort#: Địa chỉ cổng nguồn, là số hiệu của tiến trình gửi gói tin đi.
- DestPort#: Địa chỉ cổng đích, là số hiệu của tiến trình sẽ nhận gói tin.
- Length: Tổng chiều dài của segment, tính luôn cả phần header.
- Checksum: Là phần kiểm tra lỗi. UDP sẽ tính toán phần kiểm tra lỗi tổng hợp trên phần header, phần dữ liệu và cả phần header ảo. Phần header ảo chứa 3 trường trong IP header: địa chỉ IP nguồn, địa chỉ IP đích, và trường chiều dài của UDP.
- Data: Phần dữ liệu hai bên gửi cho nhau.

## 6.2.2. UDP Checksum

Phần kiểm tra (*Checksum*) của UDP được sử dụng để phát hiện lỗi. UDP phía phát thực hiện việc chia đoạn dữ liệu thành các từ mã 16 bit, sau đó thực hiện phép toán cộng các từ mã này (cộng không nhớ). Kết quả cuối cùng được lưu trong trường UDP checksum.

*Ví dụ:* Ta có các từ mã 16 bit như sau:

```
0110 0110 0110 0110;
0101 0101 0101 0101;
0000 1111 0000 1111.
```

Thực hiện phép toán cộng từng từ mã một:

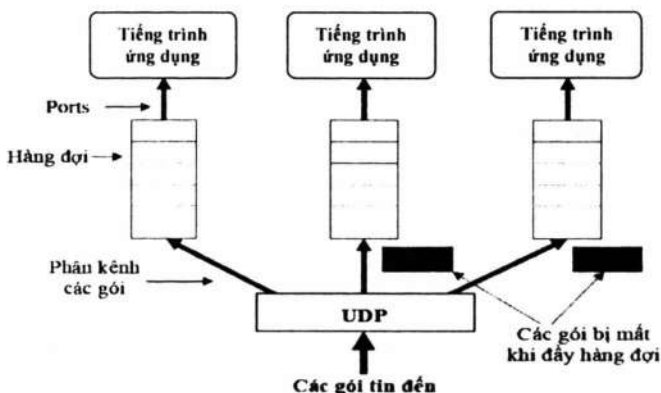
```
  0110 0110 0110 0110
+  0101 0101 0101 0101
-----
=  1011 1011 1011 1011
```

Kết quả cộng với từ mã thứ 3:

```
  1011 1011 1011 1011
+  0000 1111 0000 1111
-----
=  1100 1010 1100 1010
```

Do đó checksum của UDP là: 0011 1100 0011 1100, nếu có lỗi xảy ra thì checksum nhận được với checksum tính toán được của gói tin nhận sẽ khác nhau.

Do không có sự đảm bảo việc tất cả các đường truyền từ nguồn đến đích đều thực hiện kiểm tra lỗi và một trong số các lớp bên dưới lại sử dụng các giao thức mà không thực hiện kiểm tra lỗi. Do IP hỗ trợ chạy trên tất cả các giao thức lớp 2 như vậy lớp transport nên có cơ chế kiểm tra lỗi để đảm bảo an toàn dữ liệu. Mặc dù UDP cung cấp cơ chế kiểm tra lỗi, tuy nhiên nó không thể thực hiện việc khôi phục được các lỗi này. UDP có thể đơn giản là loại bỏ các đoạn dữ liệu bị lỗi hoặc là cho đoạn dữ liệu bị lỗi qua cùng với thông tin cảnh báo gửi tới ứng dụng.



Hình 6.6. Quá trình xử lý của các tiến trình của UDP

Khi một thông điệp được truyền đến, thông điệp được ở cuối hàng đợi. Nên khi hàng đợi đầy, thông điệp sẽ bị loại bỏ. Không có kỹ thuật điều khiển luồng được sử dụng tại đây. Một tiến trình ứng dụng muốn nhận một thông điệp nó lấy từ hàng đợi. Nếu hàng đợi rỗng tiến trình sẽ đợi cho đến khi thông điệp đến hàng đợi.

### 6.3. PHƯƠNG THỨC TRUYỀN DẪN CÓ KẾT NỐI TCP

Giao thức TCP (*Transmission Control Protocol* – "Giao thức điều khiển truyền vận") là một trong các giao thức cốt lõi của bộ giao thức TCP/IP. Sử dụng TCP, các ứng dụng trên các máy chủ được nối mạng có thể tạo các "kết nối" với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một

cách đáng tin cậy và đúng thứ tự. TCP còn phân biệt giữa dữ liệu của nhiều ứng dụng (chẳng hạn, dịch vụ Web và dịch vụ thư điện tử) đồng thời chạy trên cùng một máy chủ. TCP hỗ trợ nhiều giao thức ứng dụng phổ biến nhất trên Internet và các ứng dụng kết quả, trong đó có WWW, thư điện tử và Secure Shell.

Trong bộ giao thức TCP/IP, TCP là tầng trung gian giữa giao thức IP bên dưới và một ứng dụng bên trên. Các ứng dụng thường cần các kết nối đáng tin cậy kiểu đường ống để liên lạc với nhau, trong khi đó, giao thức IP không cung cấp những dòng kiểu đó, mà chỉ cung cấp dịch vụ chuyển gói tin không đáng tin cậy. TCP làm nhiệm vụ của tầng giao vận trong mô hình OSI đơn giản của các mạng máy tính. Các ứng dụng gửi các dòng gồm các byte 8-bit tới TCP để chuyển qua mạng. TCP phân chia dòng byte này thành các đoạn (segment) có kích thước thích hợp (thường được quyết định dựa theo kích thước của đơn vị truyền dẫn tối đa (MTU) của tầng liên kết dữ liệu của mạng mà máy tính đang nằm trong đó). Sau đó, TCP chuyển các gói tin thu được tới giao thức IP để gửi nó qua một liên mạng tới môđun TCP tại máy tính đích. TCP kiểm tra để đảm bảo không có gói tin nào bị thất lạc bằng cách gán cho mỗi gói tin một "số thứ tự" (*sequence number*). Số thứ tự này còn được sử dụng để đảm bảo dữ liệu được trao cho ứng dụng đích theo đúng thứ tự. Môđun TCP tại đầu kia gửi lại "tin báo nhận" (*acknowledgement*) cho các gói tin đã nhận được thành công; một "đồng hồ" (timer) tại nơi gửi sẽ báo time-out nếu không nhận được tin báo nhận trong khoảng thời gian bằng một round-trip time (RTT), và dữ liệu (được coi là bị thất lạc) sẽ được gửi lại. TCP sử dụng checksum (giá trị kiểm tra) để xem có byte nào bị hỏng trong quá trình truyền hay không; giá trị này được tính toán cho mỗi khối dữ liệu tại nơi gửi trước khi nó được gửi, và được kiểm tra tại nơi nhận.

### 6.3.1. Cấu trúc gói tin TCP

Một gói tin TCP bao gồm:

- Header
- Dữ liệu



- Source port: Số hiệu của cổng tại máy tính gửi.
- Destination port: Số hiệu của cổng tại máy tính nhận.
- Sequence number: Trường này có 2 nhiệm vụ. Nếu cờ SYN bật thì nó là số thứ tự gói ban đầu và byte đầu tiên được gửi có số thứ tự này cộng thêm 1. Nếu không có cờ SYN thì đây là số thứ tự của byte đầu tiên.

- Acknowledgement number: Nếu cờ ACK bật thì giá trị của trường chính là số thứ tự gói tin tiếp theo mà bên nhận cần.

- Data offset: Trường có độ dài 4 bit quy định độ dài của phần header (tính theo đơn vị từ 32 bit). Phần header có độ dài tối thiểu là 5 từ (160 bit) và tối đa là 15 từ (480 bit).

- Reserved: Dành cho tương lai và có giá trị là 0.
- Flags (*hay Control bits*): Bao gồm 6 cờ:
  - URG: Cờ cho trường Urgent pointer
  - ACK: Cờ cho trường Acknowledgement
  - PSH: Hàm Push
  - RST: Thiết lập lại đường truyền
  - SYN: Đồng bộ lại số thứ tự
  - FIN: Không gửi thêm số liệu
  - Window: Số byte có thể nhận bắt đầu từ giá trị của trường báo nhận (ACK)

- Checksum: 16 bit kiểm tra cho cả phần header và dữ liệu. Phương pháp sử dụng được mô tả trong RFC 793: 16 bit của trường kiểm tra là tổng của tổng tất cả các từ 16 bit trong gói tin. Trong trường hợp số octet (khối 8 bit) của header và dữ liệu là lẻ thì octet cuối được bổ sung với các

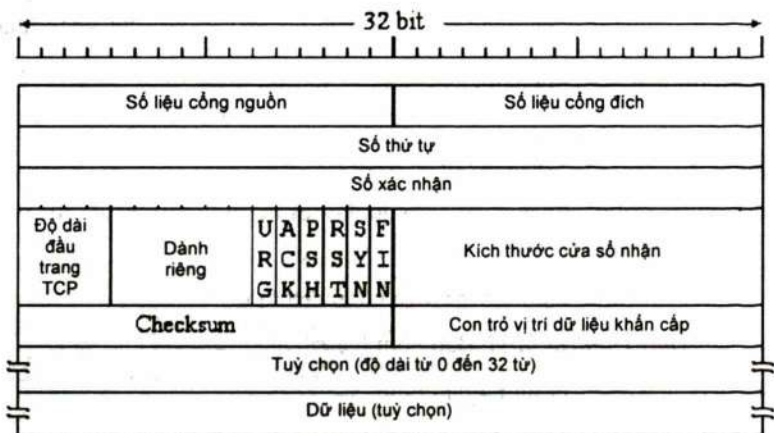
IP nguồn		↑ Giá đầu trang ↓
IP đích		
Giao thức	Chiều dài	
Đầu trang		↑ Đoạn TCP ↓
Dữ liệu		

Hình 6.7. Cấu trúc gói tin TCP



bit 0. Các bit này không được truyền. Khi tính tổng, giá trị của trường kiểm tra được thay thế bằng 0.

- Urgent pointer: Nếu cờ URG bật thì giá trị trường này chính là số từ 16 bit mà số thứ tự gói tin (sequence number) cần dịch trái.
- Options: Đây là trường tùy chọn. Nếu có thì độ dài là bội số của 32 bit data (độ dài thay đổi): chứa dữ liệu của tầng trên, có độ dài tối đa ngầm định là 536 byte. Giá trị này có thể điều chỉnh bằng cách khai báo trong vùng options.



Hình 6.8. Cấu trúc của gói tin TCP segment

### 6.3.2. Hoạt động của giao thức

Một tiến trình ứng dụng trong một host truy nhập vào các dịch vụ của TCP cung cấp thông qua một cổng (port) như sau:

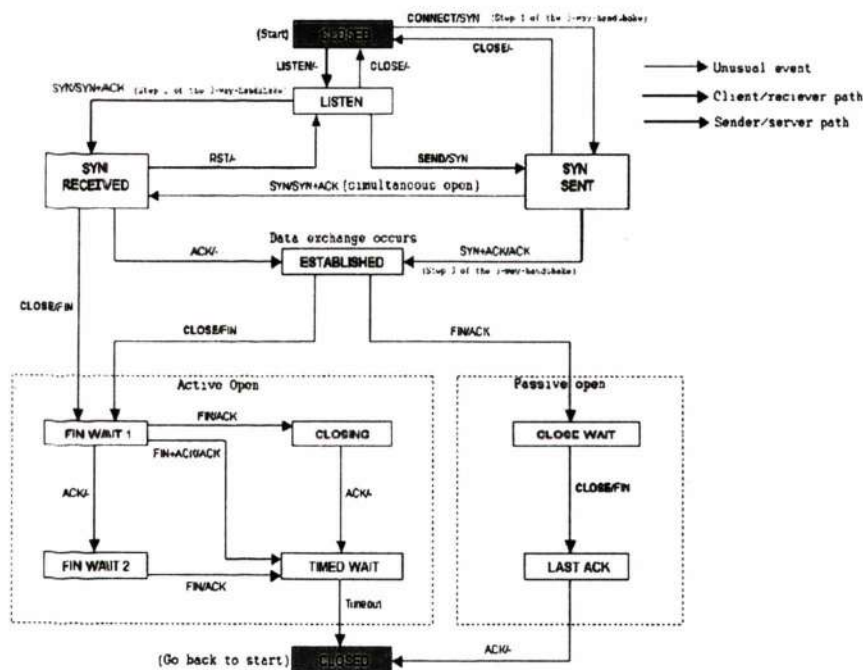
Một cổng kết hợp với một địa chỉ IP tạo thành một socket duy nhất trong liên mạng. TCP được cung cấp nhờ một liên kết logic giữa một cặp socket. Một socket có thể tham gia nhiều liên kết với các socket ở xa khác nhau. Trước khi truyền dữ liệu giữa hai trạm cần phải thiết lập một liên kết TCP giữa chúng và khi kết thúc phiên truyền dữ liệu thì liên kết đó sẽ được giải phóng. Cũng giống như ở các giao thức khác, các thực thể ở tầng trên sử dụng TCP thông qua các hàm dịch vụ nguyên thủy (service primitives),

hay còn gọi là các lời gọi hàm (function call). Không như giao thức UDP – giao thức có thể lập tức gửi gói tin mà không cần thiết lập kết nối, TCP đòi hỏi thiết lập kết nối trước khi bắt đầu gửi dữ liệu và kết thúc kết nối khi việc gửi dữ liệu hoàn tất.

Cụ thể, các kết nối TCP có ba pha:

- Thiết lập kết nối
- Truyền dữ liệu
- Kết thúc kết nối.

Trước khi miêu tả các pha này, ta cần lưu ý các trạng thái khác nhau của một socket:



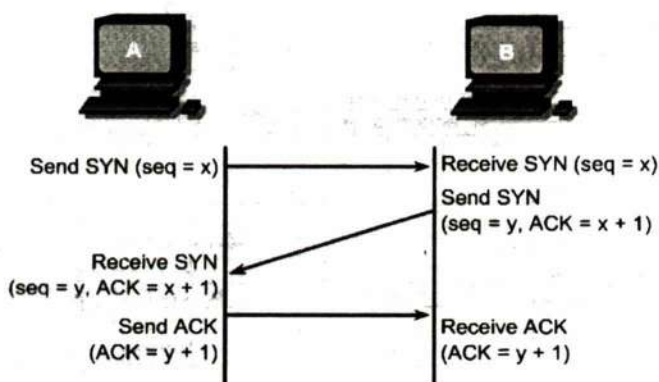
Hình 6.9. Sơ đồ trạng thái của TCP

- LISTEN: đang đợi yêu cầu kết nối từ một TCP và cổng bất kỳ ở xa (trạng thái này thường do các TCP server đặt).

- SYN-SENT: đang đợi TCP ở xa gửi một gói tin TCP với các cờ SYN và ACK được bật (trạng thái này thường do các TCP client đặt).
- SYN-RECEIVED: đang đợi TCP ở xa gửi lại một tin báo nhận sau khi đã gửi cho TCP ở xa đó một tin báo nhận kết nối (connection acknowledgment) (thường do TCP server đặt).
- ESTABLISHED: công đã sẵn sàng nhận/gửi dữ liệu với TCP ở xa (đặt bởi TCP client và server).
- TIME-WAIT: đang đợi qua đủ thời gian để chắc chắn là TCP ở xa đã nhận được tin báo nhận về yêu cầu kết thúc kết nối của nó. Theo RFC 793, một kết nối có thể ở tại trạng thái TIME-WAIT trong vòng tối đa 4 phút.

### 6.3.2.1. Thiết lập kết nối

Thiết lập kết nối TCP được thực hiện trên cơ sở phương thức bắt tay ba bước (*Tree – way Handsake*).



**Hình 6.10.** Quá trình kết nối theo 3 bước

Yêu cầu kết nối luôn được tiến trình trạm khởi tạo, bằng cách gửi một gói TCP với cờ SYN = 1 và chứa giá trị khởi tạo số tuần tự ISN của client. Giá trị ISN này là một số 4 byte không dấu và được tăng mỗi khi kết nối được yêu cầu (giá trị này quay về 0 khi nó tới giá trị 232). Trong thông điệp SYN này còn chứa số hiệu cổng TCP của phần mềm dịch vụ mà tiến trình trạm muốn kết nối (bước 1).

Mỗi thực thể kết nối TCP đều có một giá trị ISN mới số này được tăng theo thời gian. Vì một kết nối TCP có cùng số hiệu cổng và cùng địa chỉ IP được dùng lại nhiều lần, do đó việc thay đổi giá trị INS ngăn không cho các kết nối dùng lại các dữ liệu đã cũ (stale) vẫn còn được truyền từ một kết nối cũ và có cùng một địa chỉ kết nối. Khi thực thể TCP của phần mềm dịch vụ nhận được thông điệp SYN, nó gửi lại gói SYN cùng giá trị ISN của nó và đặt cờ ACK = 1 trong trường hợp sẵn sàng nhận kết nối. Thông điệp này còn chứa giá trị ISN của tiến trình trạm trong trường hợp số tuần tự thu để báo rằng thực thể dịch vụ đã nhận được giá trị ISN của tiến trình trạm (bước 2).

Tiến trình trạm trả lời lại gói SYN của thực thể dịch vụ bằng một thông báo trả lời ACK cuối cùng. Bằng cách này, các thực thể TCP trao đổi một cách tin cậy các giá trị ISN của nhau và có thể bắt đầu trao đổi dữ liệu. Không có thông điệp nào trong ba bước trên chứa bất kỳ dữ liệu gì, tất cả thông tin trao đổi đều nằm trong phần tiêu đề của thông điệp TCP (bước 3).

### 6.3.2.2. Kết thúc kết nối

Khi có nhu cầu kết thúc kết nối, thực thể TCP, ví dụ cụ thể A gửi yêu cầu kết thúc kết nối với FIN = 1. Vì kết nối TCP là song công (full-duplex) nên mặc dù nhận được yêu cầu kết thúc kết nối của A (A thông báo hết số liệu gửi) thực thể B vẫn có thể tiếp tục truyền số liệu cho đến khi B không còn số liệu để gửi và thông báo cho A bằng yêu cầu kết thúc kết nối với FIN = 1 của mình. Khi thực thể TCP đã nhận được thông điệp FIN và sau khi đã gửi thông điệp FIN của chính mình, kết nối TCP thực sự kết thúc.

### 6.3.2.3. Truyền dữ liệu

- Một số đặc điểm cơ bản của TCP để phân biệt với UDP:
  - Truyền dữ liệu không lỗi (do có cơ chế sửa lỗi/truyền lại).
  - Truyền các gói dữ liệu theo đúng thứ tự.
  - Truyền lại các gói dữ liệu mất trên đường truyền.
  - Loại bỏ các gói dữ liệu trùng lặp.
- Cơ chế hạn chế tắc nghẽn đường truyền

Ở hai bước đầu tiên trong ba bước bắt tay, hai máy tính trao đổi một số thứ tự gói ban đầu (*Initial Sequence Number – ISN*). Số này có thể chọn một



cách ngẫu nhiên. Số thứ tự này được dùng để đánh dấu các khối dữ liệu gửi từ mỗi máy tính. Sau mỗi byte được truyền đi, số này lại được tăng lên. Nhờ vậy ta có thể sắp xếp lại chúng khi tới máy tính kia bất kể các gói tới nơi theo thứ tự thế nào. Trên lý thuyết, mỗi byte gửi đi đều có một số thứ tự và khi nhận được thì máy tính nhận gửi lại tin báo nhận (ACK). Trong thực tế chỉ có byte dữ liệu đầu tiên được gán số thứ tự trong trường số thứ tự của gói tin và bên nhận sẽ gửi tin báo nhận bằng cách gửi số thứ tự của byte đang chờ.

*Ví dụ:* Máy tính A gửi 4 byte với số thứ tự ban đầu là 100 (theo lý thuyết thì 4 byte sẽ có thứ tự là 100, 101, 102, 103) thì bên nhận sẽ gửi tin báo nhận có nội dung là 104 vì đó là thứ tự của byte tiếp theo nó cần. Bằng cách gửi tin báo nhận là 104, bên nhận đã ngầm thông báo rằng nó đã nhận được các byte 100, 101, 102 và 103. Trong trường hợp 2 byte cuối bị lỗi thì bên nhận sẽ gửi tin báo nhận với nội dung là 102 vì 2 byte 100 và 101 đã được nhận thành công. Giả sử ta có 10.000 byte được gửi đi trong 10 gói tin 1.000 byte và có 1 gói tin bị mất trên đường truyền. Nếu gói bị mất là gói đầu tiên thì bên gửi sẽ phải gửi lại toàn bộ 10 gói vì không có cách nào để bên nhận thông báo nó đã nhận được 9 gói kia. Vấn đề này được giải quyết trong giao thức SCTP (*Stream Control Transmission Protocol* – "*Giao thức điều khiển truyền vận dòng*") với việc bổ sung báo nhận chọn lọc.

Số thứ tự và tin báo nhận giải quyết được các vấn đề về lặp gói tin, truyền lại những gói bị hỏng/mất và các gói tin đến sai thứ tự. Để phục vụ mục đích kiểm tra, các gói tin có trường giá trị tổng kiểm (checksum). Với trình độ hiện tại, kỹ thuật kiểm tra tổng trong TCP không đủ mạnh. Các tầng liên kết dữ liệu với xác suất lỗi bit cao có thể cần được bổ sung các khả năng phát hiện lỗi tốt hơn. Nếu như TCP được thiết kế vào thời điểm hiện tại, nhiều khả năng nó sẽ bao gồm trường kiểm tra độ dư tuần hoàn – CRC (*cyclic redundancy check*) với độ dài 32 bit. Điểm yếu này một phần được bù đắp bằng CRC hay những kỹ thuật khác tại tầng thứ 2 (trong mô hình 7 lớp OSI) ở bên dưới cả TCP và IP như trong các giao thức điểm - điểm (PPP) hoặc Ethernet. Tuy nhiên, điều này cũng không có nghĩa là trường kiểm tra tổng của TCP là không cần thiết: thống kê cho thấy các sai sót do



cả phần cứng và phần mềm gây ra giữa các điểm áp dụng kỹ thuật kiểm tra CRC là khá phổ biến và kỹ thuật kiểm tra tổng có khả năng phát hiện phần lớn các lỗi (đơn giản) này.

Điểm cuối cùng là khả năng hạn chế tắc nghẽn. Tin báo nhận (hoặc không có tin báo nhận) là tín hiệu về tình trạng đường truyền giữa 2 máy tính. Từ đó, hai bên có thể thay đổi tốc độ truyền nhận dữ liệu phù hợp với điều kiện. Vấn đề này thường được đề cập là điều khiển lưu lượng, kiểm soát tắc nghẽn. TCP sử dụng một số cơ chế nhằm đạt được hiệu suất cao và ngăn ngừa khả năng nghẽn mạng. Các cơ chế này bao gồm: cửa sổ trượt (*sliding window*), thuật toán slow-start, thuật toán tránh nghẽn mạng (*congestion avoidance*), thuật toán truyền lại và phục hồi nhanh... Hiện nay, vấn đề cải tiến TCP trong môi trường truyền dẫn tốc độ cao đang là một hướng nghiên cứu được quan tâm.

### 6.3.3. Điều khiển lưu lượng trong TCP

Giao thức TCP được xây dựng dựa trên các khái niệm được Cerf và Kahn đưa ra đầu tiên. Đó là giao thức hướng kết nối, kiểu đầu - cuối - đầu - cuối, tin cậy, được thiết kế phù hợp với kiến trúc phân lớp các giao thức. Giao thức TCP cung cấp sự truyền thông tin cậy giữa hai tiến trình chạy trên hai máy tính ở các mạng khác nhau nhưng kết nối với nhau. Giao thức TCP có khả năng gửi và nhận liên tiếp các đơn vị dữ liệu chiều dài có thể thay đổi, được gọi là phân đoạn (segment), mỗi phân đoạn được đóng gói trong một "phong bì" và tạo nên một gói số liệu IP. Khi thiết kế TCP người ta giả thiết rằng TCP sẽ nhận được các dịch vụ vận chuyển gói số liệu không tin cậy (không có sự biên nhận cho các gói số liệu) do các giao thức ở các tầng bên dưới cung cấp. Về nguyên tắc, TCP phải có khả năng hoạt động bên trên một miền rất rộng các hệ thống truyền thông, từ các mạng có các đường truyền cố định, tới các mạng chuyển mạch gói và các mạng chuyển mạch cứng.

#### 6.3.3.1. Khởi động chậm

Thuật toán khởi động chậm (*SS, Slow Start*) khắc phục nguyên nhân thứ nhất dẫn tới việc vi phạm nguyên lý "Bảo toàn các gói số liệu": Tăng dần

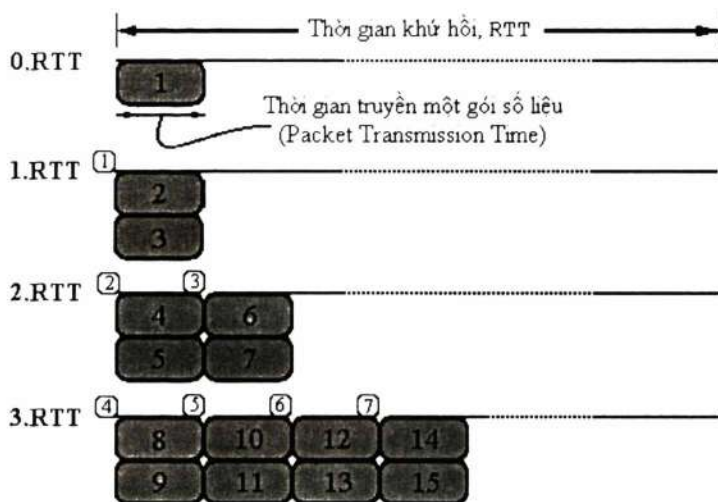
lượng dữ liệu đang được vận chuyển trong mạng để đạt tới sự cân bằng. Thuật toán được trình bày cụ thể như sau:

- Bổ sung thêm tham số cửa sổ tắc nghẽn *cwnd* (*congestion window*) vào tập trạng thái của mỗi kết nối.
- Khi bắt đầu phát hoặc bắt đầu lại việc phát sau khi có gói số liệu bị mất, đặt *cwnd* bằng một gói số liệu.
- Mỗi khi nhận được một biên nhận mới, tăng *cwnd* lên một gói số liệu.
- Khi gửi, gửi số lượng gói số liệu là min của kích thước cửa sổ mà hai bên đã thỏa thuận và *cwnd*.

Thực ra, theo cơ chế khởi động chậm, cửa sổ tăng lên theo hàm mũ, nó đạt tới kích thước  $W$  sau thời gian bằng  $RTT \cdot \log_2 W$ , trong đó  $RTT$  là thời gian khứ hồi và  $W$  tính bằng đơn vị gói số liệu. Điều này có nghĩa là cửa sổ mở đủ nhanh để ảnh hưởng không đáng kể đến hiệu năng, ngay cả trên các đường truyền có tích số dài thông  $\times$  độ trễ lớn.

Theo thuật toán này, người gửi sẽ truyền dữ liệu với tốc độ khi cao nhất là gấp đôi giá trị cực đại có thể của đường truyền. Chính vì vậy, giai đoạn khởi động chậm cần phải được kết thúc khi cửa sổ  $W$  đạt tới một ngưỡng nhất định. Hình 6.11 minh họa sự tăng của cửa sổ trong cơ chế khởi động chậm. Trục thời gian được cắt thành các đoạn có chiều dài bằng khoảng thời gian khứ hồi  $RTT$ , các đoạn này được chồng lên nhau theo chiều đứng, hướng trên – dưới ứng với chiều tăng của thời gian. Các gói số liệu được biểu diễn bằng các hình chữ nhật màu xám, bên trong là số thứ tự của gói số liệu. Các hình vuông nhỏ, không tô màu, có đánh số, biểu diễn cho các gói số liệu biên nhận tương ứng. Trên hình vẽ có thể thấy rõ, mỗi khi có một biên nhận trở về, hai gói số liệu sẽ được phát ra: một gói tương ứng với biên nhận (vì mỗi biên nhận cho biết rằng đã có một gói số liệu rời khỏi mạng, do đó cần gửi đi một gói thế chỗ cho nó), còn gói thứ hai là do biên nhận đã làm tăng cửa sổ lên một đơn vị gói số liệu. Hai gói số liệu này được vẽ chồng lên nhau, thể hiện rằng chúng cần được phát đi đồng thời, tuy nhiên, trong thực tế chỉ có thể phát chúng đi lần lượt, do đó trong khi một gói số liệu đang được phát đi, gói số liệu còn lại trong cửa sổ sẽ phải nằm chờ

trong hàng đợi. Khi cửa sổ mở rộng đến kích thước  $W$ , thì trong hàng đợi có thể có đến  $2 \times W$  gói số liệu đang xếp hàng chờ được gửi đi.



Hình 6.11. Sự tăng của cửa sổ trong cơ chế khởi động chậm

### 6.3.3.2. Tính thời gian khứ hồi một cách thông minh

Tính thời gian khứ hồi một cách thông minh là cách khắc phục nguyên nhân thứ hai dẫn tới việc vi phạm nguyên lý "Bảo toàn các gói số liệu", đó là việc đưa vào mạng một gói tin mới trước khi có một gói tin cũ ra khỏi mạng. Có hai sai lầm dẫn đến nguyên nhân thứ hai này, cách giải quyết chúng được trình bày dưới đây.

Cách giải quyết sai lầm thứ nhất: tính ước lượng thời gian khứ hồi bằng một bộ lọc dải thông thấp để tránh cho đại lượng này khởi tăng giáng quá mạnh nhằm duy trì sự cân bằng. Đặc tả cho giao thức TCP, RFC-793 gợi ý tính ước lượng thời gian khứ hồi như sau:

$$RTT \leftarrow \alpha \cdot RTT + (1 - \alpha) \cdot M \quad (1)$$

Trong đó:  $RTT$  là ước lượng thời gian khứ hồi trung bình;  $M$  là số đo thời gian khứ hồi nhận được từ gói số liệu đã được biên nhận gần nhất và  $\alpha$  là hệ số làm trơn của bộ lọc, giá trị mà người ta gợi ý nên sử dụng là  $\alpha = 0,9$ .



Sau khi ước lượng về RTT đã được cập nhật, thì thời gian hết giờ để phát lại gói số liệu tiếp theo, RTO (*Retransmit timeout*) được tính như sau:

$$RTO = \beta \cdot RTT. \quad (2)$$

Cần phải chọn  $\beta$  sao cho việc phát lại do hết giờ không bị sai lầm do thăng giáng của thời gian khứ hồi; nghĩa là làm cho xác suất thời gian khứ hồi của một gói tin lớn hơn RTO là rất nhỏ. Chính vì vậy,  $\beta$  cần được chọn không quá nhỏ, có thể sẽ dẫn đến việc phát lại vô ích, khi gói tin vẫn đang ở trong mạng;  $\beta$  cũng không được chọn quá lớn, có thể sẽ dẫn đến việc phát lại quá chậm trễ, gói tin bị mất từ lâu, mà bên gửi vẫn chờ cho hết giờ rồi mới phát lại.

Trong các phiên bản TCP được cài đặt đầu tiên, người ta thường chọn  $\beta$  là một số cố định bằng 2. Tuy nhiên, các nghiên cứu thực nghiệm sau này cho thấy rằng, RTT thăng giáng trong một miền tương đối rộng, vì vậy không nên chọn  $\beta$  theo cách đơn giản như trên. Công trình đầu tiên đề xuất việc cải tiến thuật toán tính RTO của Jacobson được công bố năm 1988. Ông đã đề xuất cách làm cho  $\beta$  xấp xỉ tỷ lệ với độ lệch chuẩn của hàm mật độ xác suất thời gian đến của biên nhận. Cụ thể là, sử dụng độ lệch trung bình như một ước lượng rẻ (*cheap estimator*) của độ lệch chuẩn. Thuật toán này đòi hỏi phải tính một biến nữa là độ lệch được làm trơn D, như sau:

$$D = \alpha \cdot D + (1 - \alpha) \cdot |RTT - M|. \quad (3)$$

Trong đó, các tham số RTT và M hoàn toàn tương tự như trong biểu thức (1), còn  $\alpha$  ở đây không nhất thiết phải có cùng giá trị như tham số  $\alpha$  trong biểu thức đó. Jacobson đã chỉ ra rằng, mặc dù D không hoàn toàn giống độ lệch chuẩn, nhưng nó cũng là một xấp xỉ đủ tốt. Cách tính D như trên nhằm đạt được tốc độ cao nhất, chỉ sử dụng các phép tính cộng, trừ và dịch trên các số nguyên. Ngày nay, các phiên bản TCP đều sử dụng thuật toán này và tính thời gian hết giờ để phát lại như sau:

$$RTO = RTT + 4 \cdot D \quad (4)$$

Sử dụng hệ số 4 có hai ưu điểm, thứ nhất là việc nhân với 4 sẽ được thực hiện bởi phép dịch, có tốc độ thực hiện cao; thứ hai là, xác suất một gói tin được biên nhận chậm hơn RTO là rất nhỏ, có thể bỏ qua.

Cách giải quyết sai lầm thứ hai: rút lui theo hàm mũ. Đây là cách giải quyết duy nhất đúng đắn, bởi vì theo cơ chế khởi động chậm, cửa sổ gửi tăng lên theo hàm mũ, cho nên cũng cần phải rút lui theo cách này cho dù nhanh khi đã có dấu hiệu của tắc nghẽn. TCP sẽ đặt đồng hồ phát lại bằng khoảng thời gian rút lui và khoảng đó sẽ được tăng gấp đôi cứ mỗi lần bị hết giờ liên tiếp.

**Bảng 6.1.** Thuật toán Slow Start (SS)

- Thực thể phát sử dụng thêm biến:
  - + cwnd (congestion window) – kích thước cửa sổ phát
  - + ssthresh (ss threshold) – giới hạn trên của cwnd, nếu vượt qua → tắc nghẽn.
- Bắt đầu phát,  $cwnd := 1$ , đó là tốc độ "an toàn nhất"
- Nhận được mỗi ack, tăng cwnd lên 1 để thăm dò (Additive Increase)
  - + Không tăng cwnd quá Window Size mà bên nhận thông báo.
  - + Thực chất, cwnd tăng lên theo hàm mũ (theo thời gian).
- Khi  $cwnd \geq ssthresh$ , chuyển sang CA

### 6.3.3.3. Tránh tắc nghẽn

Thuật toán tránh tắc nghẽn – CA (*Congestion Avoidance*) nhằm khắc phục nguyên nhân thứ ba dẫn tới việc vi phạm nguyên lý "Bảo toàn các gói số liệu". Một chiến lược tránh tắc nghẽn đã được đề xuất bao gồm hai thành phần: thứ nhất là các chính sách của mạng: mạng phải có khả năng gửi tín hiệu đến cho các thực thể cuối của các kết nối (endpoint), báo cho chúng biết là tắc nghẽn đang xảy ra hoặc sắp xảy ra; thứ hai là các chính sách của endpoint: các endpoint phải có chính sách giảm lưu lượng đưa vào mạng nếu nhận được các tín hiệu báo và tăng thêm lưu lượng đưa vào mạng nếu không nhận được tín hiệu báo này. Chính sách của endpoint đối với tắc nghẽn: thích ứng với đường truyền.

Đó chính là chính sách tăng theo cấp số cộng, giảm theo cấp số nhân, chính sách đó được giải thích như sau:



- Mỗi khi xảy ra sự kiện hết giờ, đặt giá trị cửa sổ tắc nghẽn cwnd bằng một phần hai giá trị cửa sổ hiện thời. Đó là sự giảm theo cấp số nhân.

- Mỗi khi nhận được một biên nhận cho gói số liệu mới, tăng cwnd thêm một lượng bằng  $1/cwnd$ , đây là sự tăng theo cấp số cộng. (Trong giao thức TCP, kích thước cửa sổ và kích thước gói số liệu được tính bằng byte, vì thế sự tăng nói trên được chuyển thành  $maxseg * maxseg / cwnd$ , trong đó maxseg là kích thước gói số liệu cực đại và cwnd là cửa sổ tắc nghẽn, được tính bằng bytes).

- Khi gửi, sẽ gửi đi số gói số liệu bằng số bé hơn trong hai số: kích thước cửa sổ mà bên nhận đã đề nghị và cwnd.

Trong thực tế, các thuật toán Khởi động chậm (SS) và Tránh tắc nghẽn (CA) đã được triển khai thực hiện cùng với nhau như là một thuật toán, thuật toán này được cài đặt trong Tahoe TCP – một phiên bản của TCP.

- Chính sách của mạng đối với tắc nghẽn:

Đó là các chính sách làm cho mạng, cụ thể là các gateways gửi tin hiệu báo tắc nghẽn tới các máy tính trên mạng càng sớm càng tốt, nhưng đừng quá sớm, tránh cho mạng khỏi bị thiếu lưu lượng vận chuyển. Gateway chỉ phải làm công việc loại bỏ các gói số liệu để báo cho các thực thể đã gửi các gói số liệu rằng: chúng đã sử dụng quá phần tài nguyên mạng dành cho chúng.

Chính vì thế, các thuật toán tại gateway sẽ làm giảm tắc nghẽn ngay cả khi không phải sửa đổi giao thức giao vận ở các máy tính trên mạng, để thực hiện việc tránh tắc nghẽn. Đồng thời các máy tính trên mạng có triển khai thực hiện tránh tắc nghẽn sẽ nhận được phần dài thông hợp lý dành cho nó và chỉ bị mất một số lượng tối thiểu các gói số liệu. Bởi vì tắc nghẽn tăng lên theo hàm mũ, cho nên việc phát hiện sớm là quan trọng.

Nếu tắc nghẽn được phát hiện sớm, thì chỉ cần một vài điều chỉnh nhỏ đối với cửa sổ của người gửi cũng có thể giải quyết được vấn đề; ngược lại, sẽ phải điều chỉnh rất nhiều để mạng có thể chuyển hết đồng gói số liệu tắc nghẽn trong mạng ra ngoài. Tuy nhiên, do bản chất luôn thăng giáng mạnh của lưu lượng, phát hiện tắc nghẽn sớm một cách tin cậy là một việc khó.

**Bảng 6.2.** Thuật toán Congestion Avoidance

Dấu hiệu tắc nghẽn:

RTT tăng quá Timeout, là một giá trị mà thực tế sử dụng để phán đoán là gói tin đã bị mất.

Nhận được nhiều (3) Dup Ack (biên nhận lập), điều đó cho biết đã có nhiều gói tin không đúng thứ tự đến đích, nghĩa là đã có gói tin bị mất.

$cwnd := cwnd + 1/cwnd$  với mỗi ack.

Khi phát hiện dấu hiệu tắc nghẽn:

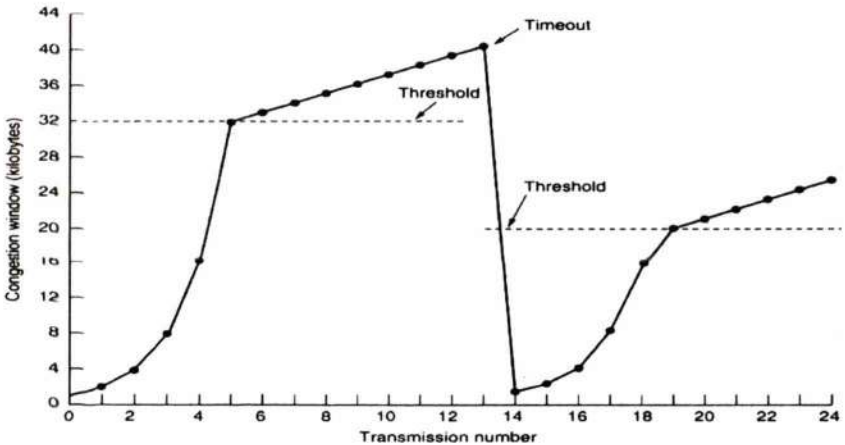
+  $ssthresh := cwnd/2, cwnd := 1$

+  $RTO = RTO * 2$  (Exponential backoff)

+  $\rightarrow SS$

*Nhận xét:*

- Trong giai đoạn CA, cwnd tăng tuyến tính:
- Đảm bảo tận dụng băng thông có thể sử dụng được.
- Vẫn thăm dò tiếp khả năng sử dụng băng thông nhiều hơn.
- cwnd bị giảm theo cấp số nhân (*Multiplicative Decreased*).

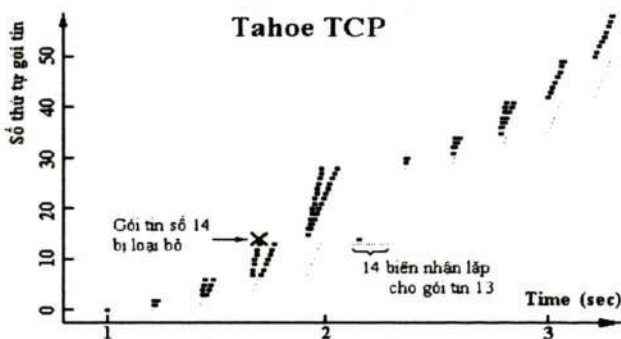


**Hình 6.12.** Minh họa thuật toán SS và CA

• Thuật toán Fast Retransmit (FRTX):

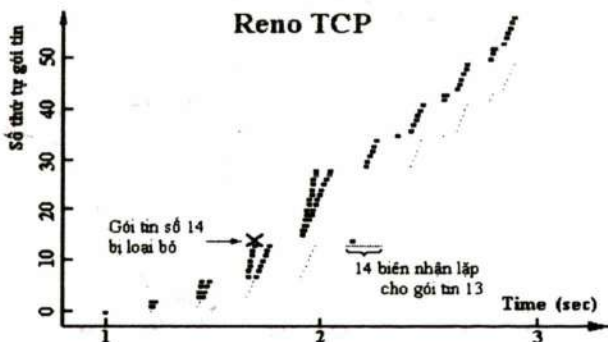
- Sau khi nhận được Dupack ( $\geq 3$ ), TCP thực hiện phát lại nhanh, không chờ bị Timeout, sau đó chuyển ngay về SS.

- Đây là một cách "dự đoán thông minh" rằng, gói tin đã bị mất.



Hình 6.13. Giao thức Tahoe TCP

- Thuật toán Fast Recovery (FRCV):
  - Cài tiến FRTX: thực hiện FRTX xong về CA chứ không về SS:
  - $ssthresh := cwnd/2$ , nhưng không nhỏ hơn 2 (gói tin)
  - $cwnd := cwnd + 3$ . Bên gửi "đoán": 3 dupack ứng với 3 gói tin đã được nhận đúng.
  - Với mỗi dupack nhận được thêm, tăng  $cwnd := cwnd + 1$



Hình 6.14. Giao thức Reno TCP

## 6.4. ĐIỀU KHIỂN LƯỜNG TRONG TCP

Điều khiển luồng định nghĩa lượng dữ liệu mà nguồn có thể gửi trước khi nhận một xác nhận từ đích. Trong trường hợp đặc biệt, giao thức lớp

giao vận có thể gửi một byte dữ liệu và đợi xác nhận trước khi gửi byte tiếp theo. Nhưng nếu làm như vậy, quá trình gửi sẽ diễn ra rất chậm. Nếu dữ liệu phải đi qua đoạn đường dài thì nguồn sẽ ở trạng thái rồi trong khi đợi xác nhận.

Trong trường hợp đặc biệt khác, giao thức lớp giao vận có thể gửi tất cả dữ liệu nó có mà không quan tâm tới xác nhận. Làm như vậy sẽ tăng tốc độ truyền, nhưng có thể làm tràn ngập trạm đích (trạm đích không xử lý kịp). Bên cạnh đó, nếu một phần dữ liệu bị mất, bị nhân đôi, sai thứ tự hoặc bị hỏng thì trạm nguồn sẽ không biết.

TCP sử dụng một giải pháp cho cả hai trường hợp đặc biệt này. Nó định nghĩa một cửa sổ, đặt cửa sổ này lên bộ đệm gửi và chỉ gửi lượng dữ liệu bằng kích thước cửa sổ.

## 6.5. ĐIỀU KHIỂN LỖI

TCP là một giao thức giao vận tin cậy. Ngoài điều khiển luồng, TCP còn điều khiển lỗi. Điều khiển lỗi gồm các cơ chế phát hiện phân đoạn bị hỏng, bị mất, sai thứ tự hoặc nhân đôi. Nó cũng gồm cơ chế sửa lỗi sau khi chúng được phát hiện.

Phát hiện lỗi trong TCP được thực hiện thông qua việc sử dụng ba công cụ đơn giản: tổng kiểm tra, xác nhận và thời gian chờ (time-out). Mỗi phân đoạn có chứa một trường tổng kiểm tra để phát hiện phân đoạn lỗi. Nếu phân đoạn lỗi, nó sẽ bị máy thu bỏ đi.

TCP sử dụng phương pháp xác nhận để thông báo sự nhận các gói đã tới đích mà không hỏng. Không có xác nhận phù định (xác nhận gói hỏng) trong TCP. Nếu một phân đoạn không được xác nhận trước khi hết hạn thì nó được xem như bị hỏng hoặc bị mất trên đường đi.

Cơ chế sửa lỗi trong TCP cũng rất đơn giản. TCP nguồn đặt một bộ định thời cho mỗi phân đoạn được gửi đi. Bộ định thời được kiểm tra định kỳ. Khi nó tắt, phân đoạn tương ứng được xem như bị hỏng hoặc bị mất và nó sẽ được truyền lại.

## CÂU HỎI VÀ BÀI TẬP

1. Gói dữ liệu trong một tầng bất kỳ gồm những thành phần nào?
2. Nêu phương pháp truyền dẫn không kết nối TCP.
3. Nêu cấu trúc của một đoạn dữ liệu.
4. Nêu hoạt động của giao thức trong phương thức truyền dẫn có kết nối TCP.
5. Hãy so sánh hai giao thức TCP và UDP.
6. Trình bày cơ chế điều khiển nghẽn được áp dụng trong kết nối TCP.
7. Nêu cơ chế điều khiển luồng và điều khiển lỗi trong TCP.
8. Nêu cấu trúc đoạn dữ liệu của giao thức UDP.



## Chương 7

# LỚP MẠNG VÀ CÁC PHƯƠNG PHÁP ĐỊNH TUYẾN

### MỤC TIÊU

Sau khi học xong chương này người học phải có những khả năng sau:

- ✓ Trình bày được chức năng lớp mạng trong mô hình phân lớp.
- ✓ Nêu được giao thức thông điệp điều khiển ICMP, giao thức định tuyến.
- ✓ So sánh được các lớp địa chỉ IP thuộc các phân lớp A, B, C, D...
- ✓ Giải được các bài toán phân chia địa chỉ mạng.
- ✓ Phát huy khả năng tư duy sáng tạo của người học.

### NỘI DUNG

- 7.1. Chức năng lớp mạng trong mô hình phân lớp
  - 7.2. Giao thức liên mạng IP
  - 7.3. Giao thức phân giải địa chỉ (Address Resolution Protocol)
  - 7.4. Giao thức thông điệp điều khiển Internet ICMP
  - 7.5. Giao thức định tuyến
  - 7.6. Router
- Câu hỏi và bài tập.

## 7.1. CHỨC NĂNG LỚP MẠNG TRONG MÔ HÌNH PHÂN LỚP

Chúng ta đã xem xét cách thức xây dựng và vận hành của các mạng đơn lẻ sử dụng các nối kết điểm - điểm, các đường truyền chia sẻ và các bộ hoán chuyển (switch). Vấn đề phát sinh là có nhiều người muốn xây dựng hệ thống mạng riêng của họ theo nhiều kỹ thuật khác nhau nhưng lại muốn giao tiếp với nhau mà không quan tâm rằng họ đang hoạt động trên các hệ thống không đồng nhất.

Chương này sẽ trình bày về cách thức để nối kết những mạng không đồng nhất lại với nhau. Có hai vấn đề quan trọng cần phải quan tâm khi nối kết các mạng: tính không đồng nhất (*heterogeneity*) và phạm vi (*scale*) khác nhau của chúng. Giải thích một cách đơn giản, tính không đồng nhất là khi người dùng trên hai mạng khác kiểu nhau muốn giao tiếp với nhau. Phức tạp hơn một chút, ta có thể thấy việc nối kết các host trên các mạng khác nhau có thể sẽ đòi hỏi việc duyệt qua nhiều mạng trung gian, mà các mạng trung gian này lại có thể có kiểu khác nhau. Chúng có thể là mạng Ethernet, Token Ring hay mạng dạng điểm nối điểm, hoặc nhiều kiểu mạng hoán chuyển (switch) khác nhau, và chúng lại sử dụng các phương thức đánh địa chỉ riêng, các phương pháp truy cập đường truyền riêng và cả mô hình dịch vụ riêng nữa. Thách thức đối với vấn đề không đồng nhất là làm sao cung cấp cho người dùng một dịch vụ nối kết host-host dễ hiểu xuyên qua mớ hỗn độn các mạng không đồng nhất.

Để hiểu về vấn đề phạm vi mạng, ta lấy một ví dụ có giá trị là sự phát triển của mạng Internet, mạng có tốc độ phát triển gần gấp đôi sau mỗi năm trong vòng 20 năm qua. Kiểu phát triển chóng mặt này buộc chúng ta phải đối mặt với nhiều thách thức. Một trong số đó là việc vạch đường: Làm sao để tìm ra một đường đi hữu hiệu xuyên qua một mạng gồm cả triệu nút mạng? Thêm một vấn đề có liên quan đến vạch đường là phương pháp đánh địa chỉ, là cách gán cho mỗi nút trên mạng một định danh duy nhất.

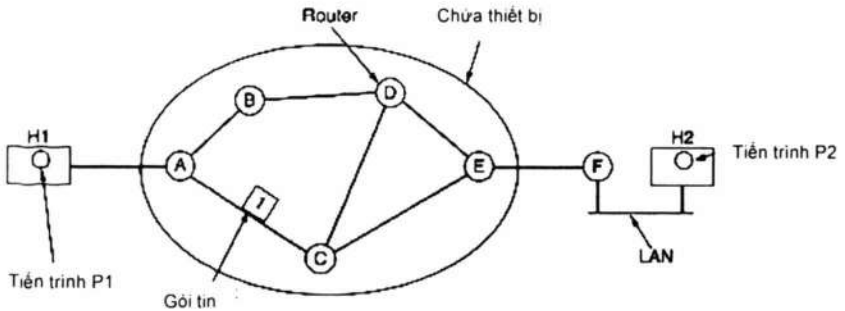
Tầng mạng có nhiệm vụ đưa các gói tin từ máy gửi qua các chặng đường để đến được máy nhận. Để đến được đích đến, gói tin có thể phải đi từng bước một qua nhiều router trung gian. Điều này thì trái ngược với tầng liên kết dữ liệu vốn chỉ chịu trách nhiệm truyền tải các khung đi từ đầu này đến đầu kia của một kênh truyền vật lý.

Để thực hiện được nhiệm vụ này, tầng mạng phải biết được hình trạng của mạng đường trục (subnet) và chọn đường thích hợp để cho gói tin đi. Nó phải chú ý đến việc chọn đường sao cho tránh được tình trạng tắc nghẽn trên một số đường truyền và router trong khi số khác thì đang rảnh rỗi.

**Các vấn đề liên quan đến việc thiết kế tầng mạng:**

**a) Kỹ thuật hoán chuyển lưu và chuyển tiếp (Store-and-Forward Switching)**

Xét một liên mạng như hình 7.1 dưới đây:



**Hình 7.1.** Môi trường của các giao thức lớp mạng  
(The environment of the network layer protocols)

Trong đó các router nằm trong hình oval được nối lại với nhau bằng các đường truyền theo kiểu điểm nối điểm được gọi là các thiết bị của nhà cung cấp đường truyền (*Carrier's equipment*). Các thiết bị nằm bên ngoài hình oval được gọi là các thiết bị của khách hàng (*Customer's Equipment*).

Máy tính H1 được nối trực tiếp vào router A của nhà cung cấp đường truyền bằng một đường nối kết thường trực (lease line). Máy H2 nối kết vào một mạng LAN cục bộ. Trong mạng LAN có router F thuộc sở hữu của khách hàng. F được nối với router E của nhà cung cấp cũng bằng một đường nối kết thường trực. Cho dù cách thức nối kết vào mạng của các máy tính có thể khác nhau như trường hợp máy H1 và H2, nhưng cách thức các gói tin của chúng được truyền đi đều giống nhau. Một máy tính có một gói tin cần truyền đi sẽ gửi gói tin đến router gần nó nhất, có thể là router trên LAN của nó hoặc router của nhà cung cấp đường truyền. Gói tin được lưu lại ở đó và được kiểm tra lỗi. Kế đến gói tin sẽ được chuyển đến một router kế tiếp trên đường đi đến đích của gói tin. Và cứ tiếp tục như thế cho đến khi đến được máy nhận gói tin. Đây chính là kỹ thuật lưu và chuyển tiếp.

### **b) Các dịch vụ cung cấp cho tầng vận chuyển**

Các dịch vụ tầng mạng cung cấp cho tầng vận chuyển cần được thiết kế hướng đến các mục tiêu sau:

- Các dịch vụ này cần nên độc lập với kỹ thuật của các router.
- Tầng vận chuyển cần được độc lập với số lượng, kiểu và hình trạng của các router hiện hành.
- Địa chỉ mạng cung cấp cho tầng vận chuyển phải có sơ đồ đánh số nhất quán cho dù chúng là LAN hay WAN.

Tầng mạng cung cấp hai dịch vụ chính là:

- Dịch vụ không nối kết (*Connectionless Service*): Các gói tin được đưa vào subnet một cách riêng lẻ và được vạch đường một cách độc lập nhau. Không cần thiết phải thiết lập nối kết trước khi truyền tin. Các gói tin trong trường hợp này được gọi là thư tín (Datagram) và subnet được gọi là Datagram Subnet.

- Dịch vụ định hướng nối kết (*Connection – Oriented Service*): Một đường nối kết giữa bên gửi và bên nhận phải được thiết lập trước khi các gói tin có thể được gửi đi. Nối kết này được gọi là mạch ảo (*Virtual Circuit*) tương tự như mạch vật lý được nối kết trong hệ thống điện thoại và subnet trong trường hợp này được gọi là virtual circuit subnet.

- Cài đặt dịch vụ không nối kết (*Implementation of Connectionless Service*)

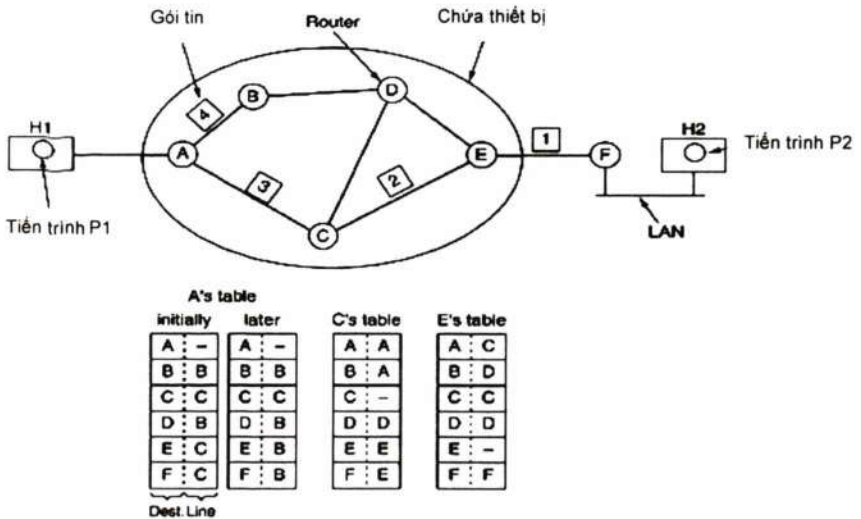
Xét hệ thống mạng như hình 7.2.

Giả sử rằng quá trình P1 có nhiều thông điệp cần gửi cho quá trình P2. Khi đó P1 sẽ gửi các thông điệp này cho tầng vận chuyển và yêu cầu tầng vận chuyển truyền sang quá trình P2 trên máy tính H2. Tầng vận chuyển sẽ gắn thêm tiêu đề (header) của nó vào thông điệp và chuyển các thông điệp xuống tầng mạng.

Giả sử rằng thông điệp gửi đi thì lớn gấp 4 lần kích thước tối đa của một gói tin, vì thế tầng mạng phải chia thông điệp ra thành 4 gói tin 1, 2, 3 và 4, và lần lượt gửi từng gói một đến router A bằng một giao thức diêm nối



điểm như PPP chẳng hạn. Mỗi router có một bảng thông tin cục bộ chỉ ra nơi nào có thể gửi các gói tin để có thể đến được những đích đến khác nhau trên mạng. Mỗi mục từ của bảng chứa 2 thông quan trọng nhất đó là Đích đến (Destination) và ngõ ra kế tiếp (Next Hop) cần phải chuyển gói tin đến để có thể đến được đích đến này. Ta gọi là bảng chọn đường (Routing Table).



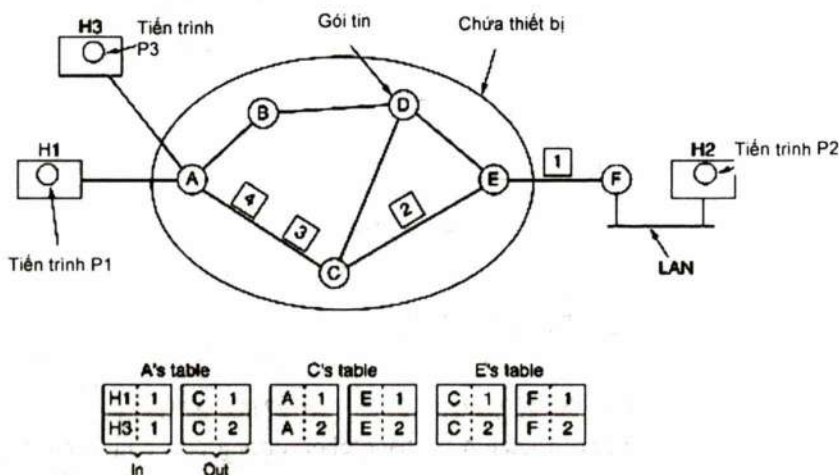
Hình 7.2. Định tuyến bên trong mạng truyền gói dữ liệu

Ví dụ: Lúc khởi đầu, router A có bảng chọn đường như hình 7.2 (lúc đầu). Khi gói tin 1, 2 và 3 đến router A, nó được lưu tạm thời để kiểm tra lỗi. Sau đó chúng được chuyển tiếp sang router C vì theo thông tin trong bảng chọn đường của A. Gói tin 1 sau đó tiếp tục được chuyển đến E và kế đến là F. Sau đó nó được gói lại trong một khung của tầng liên kết dữ liệu và được chuyển đến máy H2 bởi mạng LAN. Các gói tin 2 và 3 cũng có cùng đường đi tương tự. Sau đó, do một số sự cố về đường truyền, router A cập nhật lại bảng chọn đường của mình. Khi đó gói tin số 4 đến router A, nó sẽ chuyển gói tin này sang B để có thể đi được đến H2. Giải thuật chịu trách nhiệm quản lý thông tin trong bảng chọn đường cũng như thực hiện các quyết định về chọn đường được gọi là Giải thuật chọn đường (Routing algorithm).



- Cài đặt dịch vụ định hướng nối kết (*Connection – Oriented Service*)

Đối với dịch vụ nối kết định hướng chúng ta cần một mạch ảo trên subnet. Mục đích của việc sử dụng mạch ảo là để tránh phải thực hiện việc chọn lại đường đi mới cho mỗi gói tin gửi đến cùng một đích. Khi một nối kết được thực hiện, một đường đi từ máy tính gửi đến máy tính nhận được chọn như là một phần của giai đoạn thiết lập nối kết (*Connection setup*) và được lưu trong bảng chọn đường của các router nằm trên đường đi. Khi nối kết kết thúc, mạch ảo bị xoá. Với dịch vụ định hướng nối kết, mỗi gói tin có mang một số định dạng để xác định mạch ảo mà nó thuộc về.



**Hình 7.3.** Định tuyến bên trong mạng truyền sử dụng kênh ảo

Máy tính H1 thực hiện một nối kết với máy tính H2 qua nối kết số 1. Nối kết này được ghi nhận trong mục từ đầu tiên trong bảng chọn đường của các router. Dòng đầu tiên trong bảng chọn đường của router A nói rằng: những gói tin mang số nhận dạng nối kết số 1 đến từ máy H1 phải được gửi sang router C với số nhận dạng nối kết là 1. Tương tự, cho các mục từ đầu tiên của router C và E. Điều gì xảy ra nếu máy tính H3 muốn nối kết với máy tính H2. Nó chọn số nhận dạng nối kết là 1, vì đây là nối kết đầu tiên đối với H3, và yêu cầu subnet thiết lập mạch ảo. Điều này đã làm cho các router phải thêm vào mục từ số 2 trong bảng chọn đường. Đối với router A,

số nhận dạng nối kết với H3 là 1, trùng với nối kết với H1, không làm router A lẫn lộn vì A có thêm thông tin máy gửi là H1 hay H3. Tuy nhiên, đối với các router C, E và F thì không thể phân biệt được đâu là nối kết của H1 và đâu là nối kết của H3 nếu sử dụng số nhận dạng nối kết là 1 cho cả 2 nối kết. Chính vì thế A đã gán một số nhận dạng khác, là số 2, cho các gói tin gửi đến C có nguồn gốc từ H3.

- So sánh giữa Datagram subnet và Virtual-circuit subnet.

**Bảng 7.1.** Bảng so sánh ưu nhược điểm của hai dịch vụ không kết nối và hướng kết nối

Vấn đề	Mạng chuyển mạch gói	Mạng chuyển mạch kênh
Thiết lập nối kết	Không cần	Cần thiết
Đánh địa chỉ	Mỗi gói tin chứa đầy đủ địa chỉ gửi và nhận	Mỗi gói tin chỉ chứa số nhận dạng nối kết có kích thước nhỏ.
Thông tin trạng thái	Router không cần phải lưu giữ thông tin trạng thái của các nối kết	Mỗi nối kết phải được lưu lại trong bảng chọn đường của router.
Chọn đường	Mỗi gói tin có đường đi khác nhau	Đường đi được chọn khi mạch ảo được thiết lập, sau đó tất cả các gói tin đều đi trên đường này.
Ảnh hưởng khi router bị hỏng	Không bị ảnh hưởng, ngoại trừ gói tin đang trên đường truyền bị hỏng	Tất cả các mạch ảo đi qua router bị hỏng đều bị kết thúc
Chất lượng dịch vụ	Khó đảm bảo	Có thể thực hiện dễ dàng nếu có đủ tài nguyên gán trước cho từng nối kết
Điều khiển tắc nghẽn	Khó điều khiển	Có thể thực hiện dễ dàng nếu có tài nguyên gán cho từng kết nối

## 7.2. GIAO THỨC LIÊN MẠNG IP (INTERNET PROTOCOLS)

Các giao thức liên mạng là bộ giao thức cho các hệ thống mở nổi tiếng nhất trên thế giới, vì chúng có thể được sử dụng để giao tiếp qua bất kỳ các

liên mạng nào cũng như thích hợp cho các giao tiếp trong mạng LAN và mạng WAN.

Các giao thức liên mạng bao gồm một bộ các giao thức truyền thông, trong đó nổi tiếng nhất là Giao thức điều khiển truyền tải (*TCP – Transmission Control Protocol*) và Giao thức liên mạng (*IP – Internet Protocol*) hoạt động ở tầng 4 và tầng 3 trên mô hình OSI.

Ngoài hai giao thức này, bộ giao thức IP còn đặc tả nhiều giao thức cho lớp ứng dụng, ví dụ như giao thức cho dịch vụ thư điện tử, giao thức mô phỏng thiết bị đầu - cuối và giao thức truyền tải tập tin. Bộ giao thức liên mạng lần đầu tiên được phát triển vào giữa những năm của thập niên 70 khi Văn phòng các dự án nghiên cứu chuyên sâu của Bộ Quốc phòng Mỹ (*DARPA – Defense Advanced Research Projects Agency*) quan tâm đến việc xây dựng một mạng chuyển mạch gói (*packet-switched network*) cho phép việc trao đổi thông tin giữa các hệ thống máy tính khác nhau của các viện nghiên cứu trở nên dễ dàng hơn. Với ý tưởng nối các hệ thống máy tính không đồng nhất lại với nhau, DARPA đã cấp kinh phí nghiên cứu cho đại học Stanford, Bolt, Beranek, and Newman (BBN) về vấn đề này. Kết quả của những nỗ lực phát triển của dự án này là bộ giao thức Liên mạng đã được hoàn thành vào những năm cuối của thập niên 70 của thế kỷ XX. Sau đó TCP/IP được tích hợp vào hệ điều hành UNIX phiên bản BSD (*Berkeley Software Distribution*) trở thành nền tảng cho mạng Internet và dịch vụ WWW (*World Wide Web*).

Giao thức liên mạng, thường gọi là giao thức IP (*Internet Protocol*) là một giao thức mạng hoạt động ở tầng 3 của mô hình OSI, nó quy định cách thức định địa chỉ các máy tính và cách thức chuyển tải các gói tin qua một liên mạng. IP được đặc tả trong bảng báo cáo kỹ thuật có tên RFC (*Request For Comments*) mã số 791 và là giao thức chủ yếu trong Bộ giao thức liên mạng. Cùng với giao thức TCP, IP trở thành trái tim của bộ giao thức Internet. IP có hai chức năng chính: cung cấp dịch vụ truyền tải dạng không nối kết để chuyển tải các gói tin qua một liên mạng; và phân mảnh cũng như tập hợp lại các gói tin để hỗ trợ cho tầng liên kết dữ liệu với kích thước đơn vị truyền dữ liệu là khác nhau.

### 7.2.1. Định dạng gói tin IP

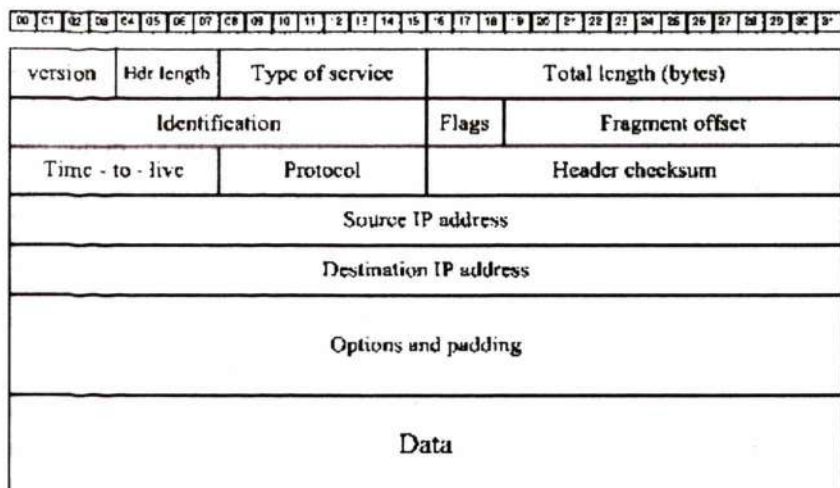
Ý nghĩa của các trường được mô tả như sau:

- Version (phiên bản): Chỉ ra phiên bản của giao thức IP dùng để tạo datagram, được sử dụng để máy gửi, máy nhận, các bộ định tuyến cùng thống nhất về định dạng lược đồ dữ liệu. Ở đây phiên bản là IPv4.

- IP header length (độ dài tiêu đề IP): Cung cấp thông tin về độ dài của tiêu đề datagram được tính theo các từ 32 bit.

- Type of service (loại dịch vụ): Trường loại phục vụ dài 8 bit gồm 2 phần, trường ưu tiên và kiểu phục vụ. Trường ưu tiên gồm 3 bit dùng để gán mức ưu tiên cho datagram, cung cấp cơ chế cho phép điều khiển các gói tin qua mạng. Các bit còn lại dùng để xác định kiểu lưu lượng datagram tin khi nó chuyển qua mạng như đặc tính thông, độ trễ và độ tin cậy. Tuy nhiên, bản thân mạng Internet không đảm bảo chất lượng dịch vụ, vì vậy trường này chỉ mang tính yêu cầu chứ không mang tính đòi hỏi đối với các bộ định tuyến.

- Total length (tổng độ dài): Trường này gồm 16 bit, nó sử dụng để xác định chiều dài của toàn bộ IP datagram.



Hình 7.4. Cấu trúc gói tin Ipv4



– Identification (nhận dạng): Trường nhận dạng dài 16 bit. Trường này được máy chủ dùng để phát hiện và nhóm các đoạn bị chia nhỏ ra của gói tin. Các bộ định tuyến sẽ chia nhỏ các datagram nếu như đơn vị truyền tin lớn nhất của gói tin (*MTU – Maximum Transmission Unit*) lớn hơn MTU của môi trường truyền.

– Flags (cờ): Chứa 3 bit được sử dụng cho quá trình điều khiển phân đoạn, bit đầu tiên chỉ thị tới các bộ định tuyến cho phép hoặc không cho phép phân đoạn gói tin, 2 bit giá trị thấp được sử dụng điều khiển phân đoạn, kết hợp với trường nhận dạng để xác định được gói tin nhận sau quá trình phân đoạn. Fragment offset: mạng thông tin về số lần chia một gói tin, kích thước của gói tin phụ thuộc vào mạng cơ sở truyền tin, tức là độ dài gói tin không thể vượt quá MTU của môi trường truyền.

– Time - to - live (thời gian sống): Được dùng để ngăn việc các gói tin lặp vòng trên mạng. Nó có vai trò như một bộ đếm ngược, tránh hiện tượng các gói tin đi quá lâu trong mạng. Bất kỳ gói tin nào có thời gian sống bằng 0 thì gói tin đó sẽ bị bộ định tuyến huỷ bỏ và thông báo lỗi sẽ được gửi về trạm phát gói tin.

– Protocol (giao thức): Trường này được dùng để xác nhận giao thức tầng kế tiếp mức cao hơn đang sử dụng dịch vụ IP dưới dạng con số.

– Header checksum: trường kiểm tra tổng header có độ dài 16 bit, được tính toán trong tất cả các trường của tiêu đề IPv4. Một gói tin khi đi qua các bộ định tuyến. Các trường trong phần tiêu đề có thể bị thay đổi, vì vậy trường này cần phải được tính toán và cập nhật lại để đảm bảo độ tin cậy của thông tin định tuyến.

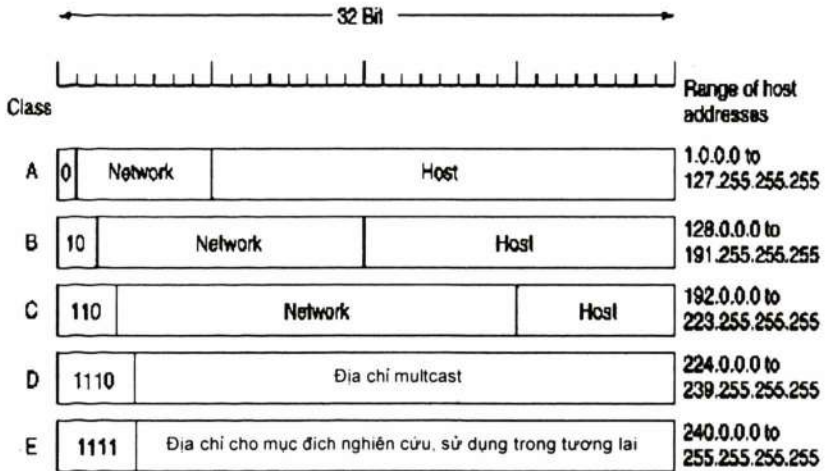
– Source Address – Destination Address (địa chỉ nguồn và địa chỉ đích): Được các bộ định tuyến và các gateway sử dụng để định tuyến các đơn vị số liệu, luôn luôn đi cùng với gói tin từ nguồn tới đích.

– Option and Padding (tuỳ chọn và đệm): Có độ dài thay đổi, dùng để thêm thông tin chọn và chèn đầy đảm bảo số liệu bắt đầu trong phạm vi 32 bit.



### 7.2.2. Cấu trúc địa chỉ IP

Mỗi máy tính trên mạng TCP/IP phải được gán một địa chỉ luận lý có chiều dài 32 bit, gọi là địa chỉ IP. 32 bit của địa chỉ IP được chia thành 2 phần: Phần nhận dạng mạng (network id) và phần nhận dạng máy tính (Host id). Phần nhận dạng mạng được dùng để nhận dạng một mạng và phải được gán bởi Trung tâm thông tin mạng Internet (InterNIC – Internet Network Information Center) nếu muốn nối kết vào mạng Internet. Phần nhận dạng máy tính dùng để nhận dạng một máy tính trong một mạng.



Hình 7.5. Phân lớp địa chỉ IP

Để dễ dàng cho việc đọc và hiểu bởi con người, 32 bit của địa chỉ IP được nhóm lại thành 4 byte và được phân cách nhau bởi 3 dấu chấm (.). Giá trị của mỗi byte được viết lại dưới dạng thập phân, với giá trị hợp lệ nằm trong khoảng từ 0 đến 255.

Câu hỏi được đặt ra là bao nhiêu bit dành cho phần nhận dạng mạng và bao nhiêu bit dành cho phần nhận dạng máy tính. Người ta phân các địa chỉ ra thành 5 lớp: A, B, C, D và E. Trong đó, chỉ có lớp A, B và C được dùng cho các mục đích thương mại. Các bits có trọng số cao nhất chỉ định lớp mạng của địa chỉ. Hình 7.5 mô tả cách phân chia lớp cho các địa chỉ IP.

Bảng 7.2. Bảng mô tả thông tin chi tiết về các lớp

Lớp	Dạng	Mục đích	Các bit cao nhất	Khoảng địa chỉ	Số bit phân nhận dạng mạng/Số bit phân nhận dạng máy tính	Tổng số máy tính trong một mạng
A	N.H.H.H	Cho một số ít các tổ chức lớn	0	1.0.0.0 đến 126.0.0.0	7/24	16.777.214 ( $2^{24} - 2$ )
B	N.N.N.H	Cho các tổ chức có kích thước trung bình	10	128.1.0.0 đến 191.254.0.0	14/16	65.543 ( $2^{16} - 2$ )
C	N.N.N.H	Cho các tổ chức có kích thước nhỏ	110	192.0.1.0 đến 223.255.254.0	21/8	254 ( $2^8 - 2$ )
D		Truyền nhóm	1110	224.0.0.0 đến 239.255.255.255		
E		Dành cho thí nghiệm	1111	240.0.0.0 đến 254.255.255.255		

### 7.2.3. Một số địa chỉ IP đặc biệt

0 0	This host	
0 0 ... 0 0	Host	A host on this network
1 1	Broadcast on the local network	
Network	1 1 1 1 ... 1 1 1 1	Broadcast on a distant network
127	(Anything)	Loopback

Hình 7.6. Cấu trúc của địa chỉ IP đặc biệt

- Địa chỉ mạng (*Network Address*) là địa chỉ IP mà giá trị của tất cả các bits ở phần nhận dạng máy tính đều là 0, được sử dụng để xác định một mạng.

*Ví dụ:* 10.0.0.0; 172.18.0.0; 192.1.1.0

- Địa chỉ quảng bá (*Broadcast Address*) là địa chỉ IP mà giá trị của tất cả các bits ở phần nhận dạng máy tính đều là 1, được sử dụng để chỉ tất cả các máy tính trong mạng.

*Ví dụ:* 10.255.255.255, 172.18.255.255, 192.1.1.255

- Mặt nạ mạng chuẩn (*Netmask*) là địa chỉ IP mà giá trị của các bits ở phần nhận dạng mạng đều là 1, các bits ở phần nhận dạng máy tính đều là 0. Như vậy ta có 3 mặt nạ mạng tương ứng cho 3 lớp mạng A, B và C là:

- Mặt nạ mạng lớp A: 255.0.0.0
- Mặt nạ mạng lớp B: 255.255.0.0
- Mặt nạ mạng lớp C: 255.255.255.0

Ta gọi chúng là các mặt nạ mạng mặc định (*Default Netmask*)

*Lưu ý:* Địa chỉ mạng, địa chỉ quảng bá, mặt nạ mạng không được dùng để đặt địa chỉ cho các máy tính. Địa chỉ mạng 127.0.0.0 là địa chỉ được dành riêng để đặt trong phạm vi một máy tính. Nó chỉ có giá trị cục bộ (trong phạm vi một máy tính). Thông thường khi cài đặt giao thức IP thì máy tính sẽ được gán địa chỉ 127.0.0.1. Địa chỉ này thông thường để kiểm tra xem giao thức IP trên máy hiện tại có hoạt động không. Địa chỉ dành riêng cho mạng cục bộ không nối kết trực tiếp Internet: Các mạng cục bộ không nối kết trực tiếp vào mạng Internet có thể sử dụng các địa chỉ mạng sau để đánh địa chỉ cho các máy tính trong mạng của mình:

- Lớp A: 10.0.0.0
- Lớp B: 172.16.0.0 đến 172.32.0.0
- Lớp C: 192.168.0.0

- Ý nghĩa của netmask:

Với một địa chỉ IP và một Netmask cho trước, ta có thể dùng phép toán AND BIT để tính ra được địa chỉ mạng mà địa chỉ IP này thuộc về. Công thức như sau: Network Address = IP Address & Netmask

*Ví dụ:* Cho địa chỉ IP = 198.53.147.45 và Netmask = 255.255.255.0. Ta thực hiện phép toán AND BIT (&) hai địa chỉ trên.

**Bảng 7.3.** Bảng sau khi thực hiện phép toán AND BIT

	<i>Biểu diễn thập phân</i>	<i>Biểu diễn nhị phân</i>
IP Address	198.53.147.45	11000110 00110101 10010011 00101101
Netmask	255.255.255.0	11111111 11111111 11111111 00000000
Network Address	198.53.147.0	11000110 00110101 10010011 00000000

## 7.2.4. Phân chia địa chỉ mạng (Subnetting)

Phân mạng con là một kỹ thuật cho phép nhà quản trị mạng chia một mạng thành những mạng con nhỏ, nhờ đó có được các tiện lợi sau:

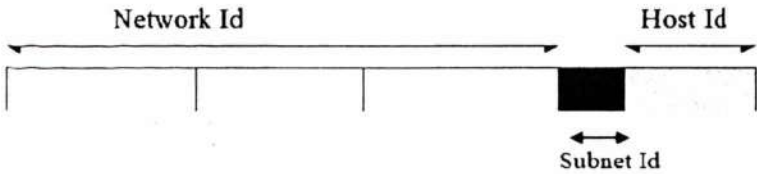
- Đơn giản hoá việc quản trị: Với sự trợ giúp của các router, các mạng có thể được chia ra thành nhiều mạng con (subnet) mà chúng có thể được quản lý như những mạng độc lập và hiệu quả hơn.
- Có thể thay đổi cấu trúc bên trong của mạng mà không làm ảnh hưởng đến các mạng bên ngoài. Một tổ chức có thể tiếp tục sử dụng các địa chỉ IP đã được cấp mà không cần phải lấy thêm khối địa chỉ mới.
- Tăng cường tính bảo mật của hệ thống: Phân mạng con sẽ cho phép một tổ chức phân tách mạng bên trong của họ thành một liên mạng nhưng các mạng bên ngoài vẫn thấy đó là một mạng duy nhất.
- Cô lập các luồng giao thông trên mạng: Với sự trợ giúp của các router, giao thông trên mạng có thể được giữ ở mức thấp nhất có thể.

### 7.2.4.1. Phương pháp phân mạng con

Nguyên tắc chung để thực hiện phân mạng con là: Phần nhận dạng mạng (Network Id) của địa chỉ mạng ban đầu được giữ nguyên. Phần nhận dạng máy tính của địa chỉ mạng ban đầu được chia thành 2 phần:

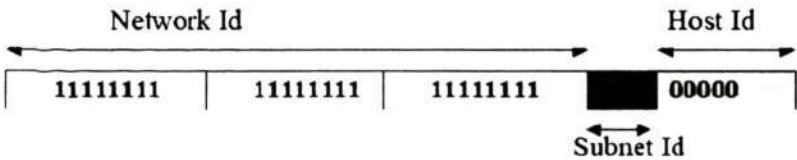
- Phần nhận dạng mạng con (*Subnet Id*).
- Phần nhận dạng máy tính trong mạng con (*Host Id*).





Hình 7.7. Địa chỉ mạng

Để phân mạng con, người ta phải xác định mặt nạ mạng con (*subnetmask*). Mặt nạ mạng con là một địa chỉ IP mà giá trị các bit ở phần nhận dạng mạng (*Network Id*) và phần nhận dạng mạng con (*Subnet Id*) đều là 1 trong khi giá trị của các bits ở phần nhận dạng máy tính (*Host Id*) đều là 0.



Hình 7.8. Mặt nạ mạng

Khi có được mặt nạ mạng con, ta có thể xác định địa chỉ mạng con (*Subnetwork Address*) mà một địa chỉ IP được tính bằng công thức sau:

$$\text{Subnetwork Address} = \text{IP} \& \text{Subnetmask}$$

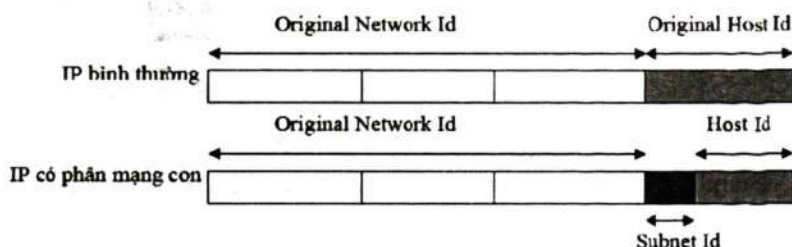
Có hai chuẩn để thực hiện phân mạng con là: Chuẩn phân lớp hoàn toàn (*Classfull standard*) và chuẩn vạch đường liên miền không phân lớp CIDR (*Classless Inter – Domain Routing*). Thực tế, CIDR chỉ mới được đa số các nhà sản xuất thiết bị và hệ điều hành mạng hỗ trợ nhưng vẫn chưa hoàn toàn chuẩn hoá.

**7.2.4.2. Phương pháp phân lớp hoàn toàn (Classfull Standard)**

Chuẩn này quy định địa chỉ IP khi phân mạng con sẽ gồm 3 phần:

- Phần nhận dạng mạng của địa chỉ ban đầu (*Network Id*).
- Phần nhận dạng mạng con (*Subnet Id*): Được hình thành từ một số bit có trọng số cao trong phần nhận dạng máy tính (*Host Id*) của địa chỉ ban đầu.
- Phần nhận dạng máy tính trong mạng con (*Host Id*) bao gồm các bit còn lại.

*Vi dụ:* Hình 7.9 mô tả cấu trúc địa chỉ IP lớp C khi thực hiện phân mạng con.



**Hình 7.9.** Mô tả địa chỉ IP lớp C

Số lượng bit thuộc phần nhận dạng mạng con xác định số lượng mạng con. Giả sử phần nhận dạng mạng con chiếm 4 bit. Như vậy, về mặt lý thuyết ta có thể phân ra thành  $2^4 = 16$  mạng con. Tuy nhiên phần nhận dạng mạng con gồm toàn bit 0 hoặc bit 1 không được dùng để đánh địa chỉ cho mạng con vì nó trùng với địa chỉ mạng và địa chỉ quảng bá của mạng ban đầu.

*Vi dụ:* Cho địa chỉ mạng lớp C: 192.168.1.0 với mặt nạ mạng mặc định là 255.255.255.0.

Xét trường hợp phân mạng con cho mạng trên sử dụng 2 bit để làm phần nhận dạng mạng con. Mặt nạ mạng trong trường hợp này là 255.255.255.192. Khi đó ta có các địa chỉ mạng con như bảng 7.4.

Ta nhận thấy rằng:

Địa chỉ mạng con thứ nhất 192.168.1.0 trùng với địa chỉ mạng ban đầu. Địa chỉ mạng con thứ tư 192.168.1.192 có địa chỉ quảng bá trùng với địa chỉ quảng bá của mạng ban đầu. Chính vì thế mà hai địa chỉ này (có phần nhận dạng mạng con toàn bit 0 hoặc toàn bit 1) không được dùng để đánh địa chỉ cho mạng con.

Nói tóm lại, với  $n$  bit làm phần nhận dạng mạng con ta chỉ có thể phân ra được  $2^n - 2$  mạng con mà thôi. Mỗi mạng con cũng có địa chỉ quảng bá. Đó là địa chỉ mà các bit ở phần nhận dạng máy tính đều có giá trị là 1.

**Bảng 7.4.** Bảng kết quả chia subnet lớp C 192.168.1.0

Địa chỉ IP	Biểu diễn dạng nhị phân	Biểu diễn dạng nhị phân			
Mạng ban đầu	192.168.1.0	1100 0000	1010 1000	0000 0001	0000 0000
Mạng con 1	192.168.1.0	1100 00001	1010 1000	0000 0001	0000 0000
Mạng con 2	192.168.1.64	1100 0000	1010 1000	0010 0001	0100 0000
Mạng con 3	192.168.1.128	1100 0000	1010 1000	0010 0001	1000 0000
Mạng con 4	192.168.1.192	1100 0000	1010 1000	0000 0001	1100 0000

*Ví dụ:*

**Bảng 7.5.** Bảng kết quả địa chỉ mạng và địa chỉ quảng bá sau khi chia subnet lớp C với địa chỉ 192.168.1.0

Địa chỉ IP	Biểu diễn dạng thập phân	Biểu diễn dạng nhị phân			
Mạng con 1	192.168.1.64	1100 0000	1010 1000	0010 0001	0100 0000
Địa chỉ quảng bá	192.168.1.127	1100 0000	1010 1000	0010 0001	0111 1111
Mạng con 2	192.168.1.128	1100 0000	1010 1000	0000 0001	1000 0000
Địa chỉ quảng bá	192.168.1.191	1100 0000	1010 1000	0000 0001	1011 1111

Như vậy quy trình phân mạng con có thể được tóm tắt như sau:

- Xác định số lượng mạng con cần phân, giả sử là N.
- Biểu diễn (N + 1) thành số nhị phân. Số lượng bit cần thiết để biểu diễn (N + 1) chính là số lượng bit cần dành cho phần nhận dạng mạng con.

Ví dụ  $N = 6$ , khi đó biểu diễn của  $(6 + 1)$  dưới dạng nhị phân là 111. Như vậy cần dùng 3 bit để làm phần nhận dạng mạng con.

- Tạo mặt nạ mạng con:

Liệt kê tất cả các địa chỉ mạng con có thể, trừ hai địa chỉ mà ở đó phần nhận dạng mạng con toàn các bit 0 và các bit 1. Chọn ra  $N$  địa chỉ mạng con từ danh sách các mạng con đã liệt kê.

### 7.2.4.3. Phương pháp vạch đường liên miền không phân lớp CIDR (Classless Inter – Domain Routing)

CIDR là một sơ đồ đánh địa chỉ mới cho mạng Internet hiệu quả hơn nhiều so với sơ đồ đánh địa chỉ cũ theo kiểu phân lớp A, B và C. CIDR ra đời để giải quyết hai vấn đề bức xúc đối với mạng Internet.

- Vấn đề thiếu địa chỉ IP:

Với sơ đồ đánh địa chỉ truyền thống, các địa chỉ được phân ra thành các lớp A, B và C. Mỗi địa chỉ có 2 phần, phần nhận dạng mạng và phần nhận dạng máy tính. Khi đó số lượng mạng và số máy tính tối đa cho từng mạng được thống kê như bảng 7.6.

**Bảng 7.6.** Bảng số lượng mạng và số máy tính tối đa trong mạng của các lớp địa chỉ A, B, C

Lớp mạng	Số lượng mạng	Số máy tính tối đa trong mạng
A	126	16.777.214
B	65.000	65.534
C	Hơn 2 triệu	254

Bởi vì các địa chỉ của mạng Internet thường được gán theo kích thước này dẫn đến tình trạng lãng phí. Trường hợp bạn cần 100 địa chỉ, bạn sẽ được cấp một địa chỉ lớp C. Như vậy còn 154 địa chỉ không được sử dụng. Chính điều này dẫn đến tình trạng thiếu địa chỉ IP cho mạng Internet. Theo thống kê, chỉ có khoảng 3% số địa chỉ đã được cấp phát được sử dụng đến. Chính vì thế sơ đồ đánh địa chỉ mới CIDR ra đời để khắc phục tình trạng trên.



Khi số lượng mạng trên mạng Internet tăng cũng đồng nghĩa với việc tăng số lượng router trên mạng. Trong những năm gần đây, người ta dự đoán rằng các router đường trục của mạng Internet đang nhanh chóng tiến đến mức ngưỡng tối đa số lượng router mà nó có thể chấp nhận được. Thậm chí với những công nghệ hiện đại dùng để sản xuất các router thì về mặt lý thuyết kích thước tối đa của một bảng chọn đường cũng chỉ đến 60.000 mục từ (đường đi). Nếu không có những cải tiến thì các router đường trục sẽ đạt đến con số này và như thế không thể mở rộng mạng Internet hơn nữa.

Để giải quyết hai vấn đề trên, cộng đồng Internet đã đưa ra các giải pháp sau:

- Sửa đổi lại cấu trúc cấp phát địa chỉ IP.

CIDR được sử dụng để thay thế cho sơ đồ cấp phát cũ với việc quy định các lớp A, B, C. Thay vì phần nhận dạng mạng được giới hạn với 8, 16 hoặc 24 bit, CIDR sử dụng phần nhận dạng mạng có tính tổng quát từ 13 đến 27 bit. Chính vì thế các khối địa chỉ có thể được gán cho mạng nhỏ nhất với 32 máy tính đến mạng lớn nhất hơn 500.000 máy tính. Điều này đáp ứng gần đúng yêu cầu đánh địa chỉ của các tổ chức khác nhau.

Một địa chỉ theo cấu trúc CIDR, gọi tắt là địa chỉ CIDR, bao gồm 32 bit của địa chỉ IP chuẩn cùng với một thông tin bổ sung về số lượng các bit được sử dụng cho phần nhận dạng mạng.

*Ví dụ:* Với địa chỉ CIDR 206.13.01.48/25 thì chuỗi số "/25" chỉ ra rằng 25 bit đầu tiên trong địa chỉ IP được dùng để nhận dạng duy nhất một mạng, số bit còn lại dùng để nhận dạng một máy tính trong mạng.

Bảng 7.7 so sánh giữa sơ đồ đánh địa chỉ theo kiểu CIDR và sơ đồ đánh địa chỉ theo chuẩn phân lớp hoàn toàn.

Kết hợp việc chọn đường có cấu trúc để giảm tối đa số lượng các mục từ trong bảng chọn đường.

Sơ đồ đánh địa chỉ theo CIDR cũng cho phép kết hợp các đường đi, ở đó mục từ trong bảng chọn đường ở mức cao có thể đại diện cho nhiều router ở mức thấp hơn trong các bảng chọn đường tổng thể. Sơ đồ này giống như hệ thống mạng điện thoại ở đó mạng được thiết lập theo kiến trúc phân

cấp. Một router ở mức cao (quốc gia), chỉ quan tâm đến mã quốc gia trong số điện thoại, sau đó nó sẽ vạch đường cho cuộc gọi đến router đường trực phụ trách mạng quốc gia tương ứng với mã quốc gia đó. Router nhận được cuộc gọi nhìn vào phần đầu của số điện thoại, mã tỉnh, để vạch đường cho cuộc gọi đến một mạng con tương ứng với mã tỉnh đó, và cứ như thế.

**Bảng 7.7.** Bảng cách đánh địa chỉ theo CIDR

Số bit nhận mạng trong địa chỉ CIDR	Lớp tương ứng trong chuẩn phân lớp hoàn toàn	Số lượng máy tính trong mạng
/27	1/8 Lớp C	32
/26	1/4 Lớp C	64
/25	1/2 Lớp C	128
/24	1 Lớp C	256
/23	2 Lớp C	512
/22	4 Lớp C	1.024
/21	8 Lớp C	2.048
/20	16 Lớp C	4.096
/19	32 Lớp C	8.192
/18	64 Lớp C	16.384
/17	128 Lớp C	
/16	256 Lớp C (= 1 Lớp B)	65.536
/15	512 Lớp C	131.072
/14	1024 Lớp C	262.144

Trong sơ đồ này, các router đường trực chỉ lưu giữ thông tin về mã quốc gia cho mỗi mục từ trong bảng chọn đường của mình, mỗi mục từ như thể đại diện cho một số không lồ các số điện thoại riêng lẻ chứ không phải là một số điện thoại cụ thể. Thông thường, các khối địa chỉ lớn được cấp

cho các nhà cung cấp dịch vụ Internet (IP- Internet Service Providers) lớn, sau đó họ lại cấp lại các phần trong khối địa chỉ của họ cho các khách hàng của mình. Hiện tại, mạng Internet sử dụng cả hai sơ đồ cấp phát địa chỉ Classfull standard và CIDR. Hầu hết các router mới đều hỗ trợ CIDR và những nhà quản lý Internet thì khuyến khích người dùng cài đặt sơ đồ đánh địa chỉ CIDR. Tham khảo thêm về CIDR ở địa chỉ <http://www.rfc-editor.org/rfcsearch.html> với các RFC liên quan sau:

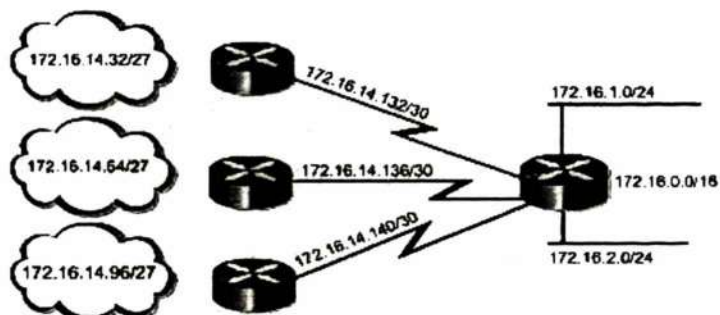
- RFC 1517: Applicability Statement for the Implementation of CIDR.
- RFC 1518: An Architecture for IP Address Allocation with CIDR.
- RFC 1519: CIDR: An Address Assignment and Aggregation Strategy.
- RFC 1520: Exchanging Routing Information Across Provider Boundaries in the CIDR Environment.

#### **7.2.4.4. Kỹ thuật phân chia theo VLSM – Variable Length Subnet Mask**

Khi mạng IP phát triển lớn hơn, người quản trị mạng phải có cách sử dụng không gian địa chỉ của mình một cách hiệu quả hơn. Một trong những kỹ thuật thường được sử dụng là VLSM. Với VLSM, người quản trị mạng có thể chia địa chỉ mạng có Subnet mask dài cho mạng có ít host và địa chỉ mạng có Subnet mask ngắn cho mạng có nhiều host.

Khi sử dụng VLSM thì hệ thống phải chạy giao thức định tuyến có hỗ trợ VLSM như: OSPF (*Open Shortest Path First*), EIGRP, RIPv2 và định tuyến cố định. VLSM cho phép một tổ chức sử dụng chiều dài SUBNET MASK khác nhau trong một địa chỉ mạng lớn. VLSM còn được gọi là "chia subnet trong một subnet lớn hơn" giúp tận dụng tối đa không gian địa chỉ. Giao thức định tuyến theo lớp địa chỉ thì đòi hỏi toàn bộ hệ thống mạng phải có cùng subnet mask.

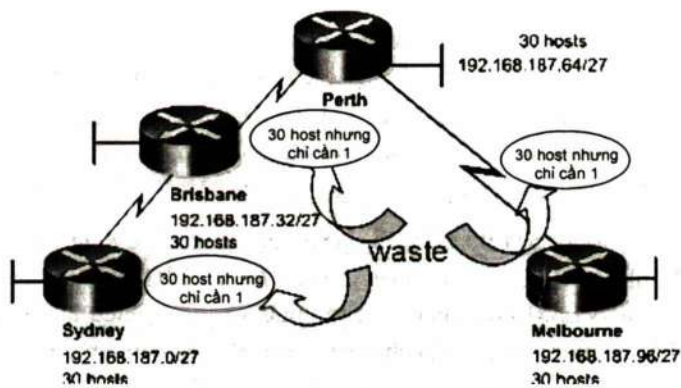
Với VLSM thì chúng ta có thể chia một địa chỉ mạng lớn thành nhiều địa chỉ mạng con có kích thước khác nhau: như địa chỉ mạng có 30 bit subnet mask, 255.255.255.252, để dành cho các kết nối mạng; địa chỉ mạng có 24 bit subnet mask, 255.255.255.0, để dành cho các mạng có dưới 254 user, các địa chỉ mạng có 22 bit subnet mask, 255.255.252.0 để dành cho các mạng có tới 1000 user.



Hình 7.10. Một ví dụ về chia địa chỉ IP theo VLSM

Với VLSM chúng ta có thể tận dụng subnet đầu tiên và subnet cuối cùng.

Ví dụ: Người quản trị mượn 3 bit để chia subnet cho địa chỉ lớp C 192.168.187.0.

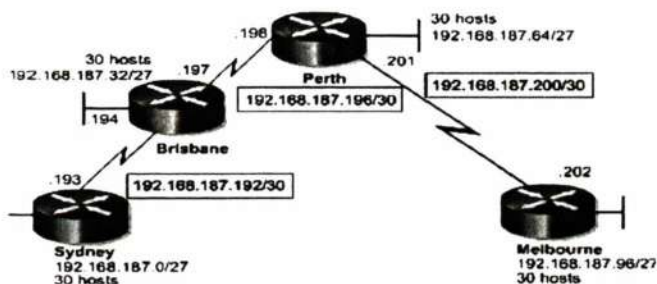


Subnet Number	Subnet Address	
Subnet 0	192.168.187.0	/27
Subnet 1	192.168.187.32	/27
Subnet 2	192.168.187.64	/27
Subnet 3	192.168.187.96	/27
Subnet 4	192.168.187.128	/27
Subnet 5	192.168.187.160	/27
Subnet 6	192.168.187.192	/27
Subnet 7	192.168.187.224	/27

Hình 7.11. Hệ thống mạng không sử dụng VLSM để phân chia địa chỉ mạng



Với sơ đồ trên, nếu không sử dụng subnet đầu tiên người quản trị mạng sẽ có 7 subnet sử dụng được. Bây giờ, mỗi subnet được phân phối cho một mạng LAN trên các router Sydney, Brisbane, Perth và Melbourne, 3 subnet còn lại được phân phối cho 3 đường kết nối serial giữa các router. Như vậy, không còn subnet nào dự phòng cho sự mở rộng về sau. Trong khi đó, kết nối giữa 2 router là kết nối điểm - điểm nên chỉ cần 2 địa chỉ host là đủ. Như vậy phí mất 28 địa chỉ host trong mỗi subnet được phân phối cho liên kết WAN của router. Với cách chia đều, tất cả các subnet có chiều dài subnet bằng nhau như vậy, 1/3 không gian địa chỉ đã bị phí phạm.



Sử dụng VLSM trên đường kết nối điểm tới điểm  
chỉ sử dụng 2 địa chỉ cho host thay vì 30

Subnet Number	Subnet Address	
subnet 0	192.168.187.0	/27
subnet 1	192.168.187.32	/27
subnet 2	192.168.187.64	/27
subnet 3	192.168.187.96	/27
subnet 4	192.168.187.128	/27
subnet 5	192.168.187.160	/27
subnet 6	192.168.187.192	/27
subnet 7	192.168.187.224	/27
Subnet Number	Subnet Address	
sub-subnet 0	192.168.187.192	/30
sub-subnet 1	192.168.187.196	/30
sub-subnet 2	192.168.187.200	/30
sub-subnet 3	192.168.187.204	/30
sub-subnet 4	192.168.187.208	/30
sub-subnet 5	192.168.187.212	/30
sub-subnet 6	192.168.187.216	/30
sub-subnet 7	192.168.187.220	/30

Hình 7.12. Hệ thống được phân chia địa chỉ mạng theo VLSM

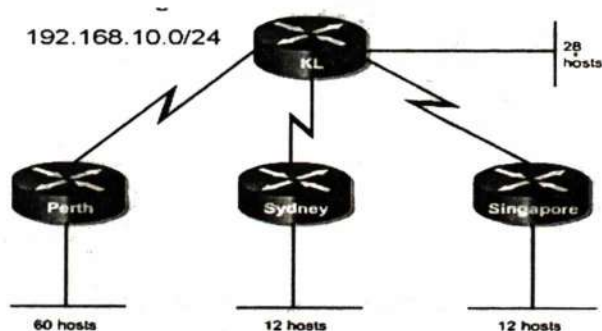
Cách phân phối địa chỉ như trên chỉ phù hợp với mạng nhỏ.

Vậy khi nào sử dụng VLSM?

Thiết kế sơ đồ địa chỉ IP sao cho đáp ứng được sự mở rộng sau này và không phí phạm địa chỉ là một việc hết sức quan trọng. Cũng với ví dụ trên, lần này người quản trị sử dụng VLSM để chia địa chỉ mạng lớp C 192.168.187.0 thành nhiều subnet khác nhau.

Xét một số ví dụ sau:

*Ví dụ 1:* Giả sử có địa chỉ mạng lớp C 192.168.10.0/24 để phân phối cho hệ thống mạng này:



**Hình 7.13.** Hệ thống mạng với địa chỉ mạng 192.168.10.0/24

Đầu tiên xét mạng LAN có nhiều user nhất trong hệ thống. Hệ thống trên là mạng LAN có 60 user. Nếu như chia subnet theo cách cũ, chúng ta chỉ mượn 2 bit để chia subnet, còn lại 6 bit dành cho host ( $2^6 - 2 = 62$  host) mới đủ đáp ứng cho mạng LAN 60 host. Như vậy chúng ta chỉ tạo được  $2^2 = 4$  subnet, trong đó sử dụng được tối đa 3 subnet, không đủ đáp ứng cho toàn bộ hệ thống mạng.

Chúng ta phải sử dụng VLSM như sau:

- Bước đầu tiên chúng ta cũng xét mạng LAN lớn nhất trong hệ thống là mạng LAN có 60 host ở Perth. Để đáp ứng cho mạng LAN này, chúng ta mượn 2 bit để chia subnet cho địa chỉ 192.168.10.0/24. Chúng ta sẽ được 4 subnet/26 như bảng 7.8.

**Bảng 7.8.** Bảng kết quả chia mạng con theo VLSM lần 1

#	Subnet – works	Dải địa chỉ host	Địa chỉ quảng bá
0	192.168.10.0/26	192.168.10.1 – 192.168.10.62	192.168.10.63
1	192.168.10.64/26	192.168.10.65 – 192.168.10.126	192.168.10.127
2	192.168.10.128/36	192.168.10.129 – 192.168.10.190	192.168.10.191
3	192.168.10.192/26	192.168.10.193 – 192.168.10.254	192.168.10.255

Chúng ta lấy subnet đầu tiên 192.168.10.0/26 phân phối cho mạng LAN 60 host của Perth.

- Bước thứ 2, chúng ta xét tới mạng LAN 28 host. Để đáp ứng cho mạng LAN này, chúng ta lấy subnet tiếp theo là 192.168.10.64/26, mượn tiếp 1 bit nữa để tách thành 2 subnet (/27) nhỏ hơn như bảng 7.9.

**Bảng 7.9.** Bảng kết quả chia mạng con theo VLSM lần 2

#	Subnet – works	Dải địa chỉ host	Địa chỉ quảng bá
0	192.168.10.64/27	192.168.10.65/27 – 192.168.10.94/27	192.168.10.95/27
1	192.168.10.96/27	192.168.10.97/27 – 192.168.126.10/27	192.168.10.127/27

Mỗi subnet/27 có 5 bit dành cho địa chỉ host nên đáp ứng được tối đa  $2^5 - 2 = 30$  host. Do đó ta lấy subnet 192.168.10.64/27 để phân phối cho mạng LAN 28 host.

- Bước thứ 3, chúng ta xét tiếp đến các mạng LAN nhỏ hơn tiếp theo. Chúng ta còn lại 2 mạng LAN ở Sydney và Singapore, mỗi mạng 12 host. Để đáp ứng cho 2 mạng này chúng ta lấy subnet 192.168.10.96/27 ở trên, mượn tiếp 1 bit nữa để tách thành 2 subnet/28 như bảng 7.10.

**Bảng 7.10.** Bảng kết quả chia mạng con theo VLSM lần 3

#	Subnet – works	Dải địa chỉ host	Địa chỉ quảng bá
0	192.168.96/28	192.168.10.96/28–192.168.10.110/28	192.168.10.111/28
1	192.168.10.112/28	192.168.10.113/28–192.168.10.126	192.168.10.127/28

Mỗi subnet/28 còn 4 bit dành cho phần host nên đáp ứng được tối đa  $2^4 - 2 = 14$  host. Chúng ta lấy 2 subnet/28 trong bảng trên phân phối cho hai mạng LAN ở Sydney và Singapore.

• Bước cuối cùng, bây giờ chỉ còn lại 3 đường liên kết WAN giữa các router, mỗi đường liên kết cần 2 địa chỉ host. Chúng ta thấy từ đầu đến giờ chúng ta đã sử dụng hết dải địa chỉ 192.168.10.0 –192.168.10.127. Bây giờ chúng ta lấy tiếp subnet 192.168.10.128/26 đã tạo ở bước 1, mượn tiếp 4 bit để tạo thành 16 subnet/30 như bảng 7.11.

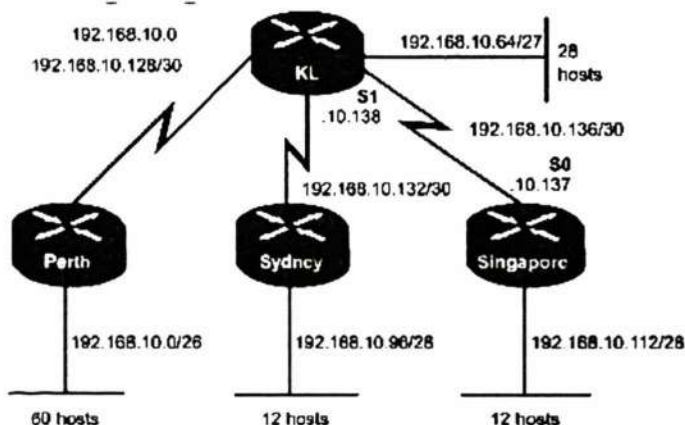
**Bảng 7.11.** Bảng kết quả chia mạng con theo VLSM lần 4

#	Subnet – works	Dải địa chỉ host	Địa chỉ quảng bá
0	192.168.10.128/30	192.168.10.129 – 192.168.10.130	192.168.10.131/30
1	192.168.10.132/30	192.168.10.133 – 192.168.10.134	192.168.10.135/30
2	192.168.10.136/30	192.168.10.137 – 192.168.10.138	192.168.10.139
3	192.168.10.140	192.168.10.142 – 192.168.10.142	192.168.10.143

Ta lấy 3 subnet đầu tiên trong bảng trên để phân phối cho các liên kết WAN giữa các router.

Kết quả chúng ta được sơ đồ phân phối địa chỉ như hình 7.14.





Hình 7.14. Hệ thống mạng phân chia địa chỉ mạng theo VLSM

### 7.3. GIAO THỨC PHÂN GIẢI ĐỊA CHỈ (Address Resolution Protocol)

Giao thức TCP/IP sử dụng ARP để tìm địa chỉ vật lý của trạm đích. Ví dụ khi cần gửi một gói dữ liệu IP cho một hệ thống khác trên cùng một mạng vật lý Ethernet, hệ thống gửi cần biết địa chỉ Ethernet của hệ thống đích để tăng liên kết dữ liệu xây dựng khung gói dữ liệu. Thông thường, mỗi hệ thống lưu giữ và cập nhật bảng thích ứng địa chỉ IP-MAC tại chỗ còn được gọi là bảng ARP Cache). Bảng thích ứng địa chỉ được cập nhật bởi người quản trị hệ thống hoặc tự động bởi giao thức ARP sau mỗi lần ánh xạ được một địa chỉ tương ứng mới.

Trước khi trao đổi thông tin với nhau, node nguồn cần phải xác định địa chỉ vật lý MAC của node đích bằng cách tìm kiếm trong bảng địa chỉ IP. Nếu không tìm thấy, node nguồn gửi quảng bá (Broadcast) một gói yêu cầu ARP (*ARP Request*) có chứa địa chỉ IP nguồn, địa chỉ IP đích cho tất cả các máy trên mạng. Các máy nhận, đọc, phân tích và so sánh địa chỉ IP của nó với địa chỉ IP của gói.

Nếu cùng địa chỉ IP, nghĩa là node đích tìm trong bảng thích ứng địa chỉ IP-MAC của nó và trả lời bằng một gói ARP Reply có chứa địa chỉ MAC cho node nguồn.

Nếu không cùng địa chỉ IP, nó chuyển tiếp gói yêu cầu nhận được dưới dạng quảng bá cho tất cả các trạm trên mạng.

Tóm lại tiến trình của ARP được mô tả như sau:

- IP yêu cầu địa chỉ MAC.
- Tìm kiếm trong bảng ARP.
- Nếu tìm thấy sẽ trả lại địa chỉ MAC.
- Nếu không tìm thấy, tạo gói ARP yêu cầu và gửi tới tất cả các trạm.
- Tuỳ theo gói tin trả lời, ARP cập nhật vào bảng ARP và gửi địa chỉ MAC cho IP.

**Bảng 7.12.** Bảng các trường của tiến trình ARP

Tổng quát	Các trường	Kích thước (byte)	Các giá trị
Ethernet Header	Ethernet Destination Address	6	Địa chỉ máy nhận, trong trường hợp này là một địa chỉ quảng bá.
	Ethernet Source Address	6	Địa chỉ của máy gửi thông điệp.
	Frame Type	2	Kiểu khung, có giá trị là 0x0806 khi ARP yêu cầu và 0x0835 khi ARP trả lời
ARP request/reply	Hardware Type	2	Giá trị là 1 cho mạng Ethernet
	Protocol Address	2	Có giá trị là 0x0800 cho địa chỉ IP
	Hardware Address	1	Chiều dài của địa chỉ vật lý, có giá trị là 6

*Giao thức phân giải địa chỉ ngược RARP (Reverse Address Resolution Protocol) RARP là giao thức phân giải địa chỉ ngược.*

Quá trình này ngược lại với quá trình ARP ở trên, nghĩa là cho trước địa chỉ mức liên kết, tìm địa chỉ IP tương ứng. Như vậy RARP được sử dụng để phát hiện địa chỉ IP, khi biết địa chỉ vật lý MAC. Và cũng được sử dụng

trong trường hợp trạm làm việc không có đĩa khuôn dạng gói tin RARP tương tự như khuôn dạng gói ARP đã trình bày, chỉ khác là trường Opcode có giá trị  $0 \times 0003$  cho mã lệnh yêu cầu (*RARP Request*) và có giá trị  $0 \times 0004$  cho mã lệnh trả lời (*RARP Reply*).

Nguyên tắc hoạt động của RARP ngược với ARP, nghĩa là máy đã biết trước địa chỉ vật lý MAC tìm địa chỉ IP tương ứng của nó. Máy A cần biết địa chỉ IP của nó, nó gửi gói tin RARP Request chứa địa chỉ MAC cho tất cả các máy trong mạng LAN. Mọi máy trong mạng đều có thể nhận gói tin này nhưng chỉ có Server mới trả lại RARP Reply chứa địa chỉ IP của nó.

## 7.4. GIAO THỨC THÔNG ĐIỆP ĐIỀU KHIỂN INTERNET ICMP

Giao thức IP không có cơ chế kiểm soát lỗi và kiểm soát luồng dữ liệu. Các nút mạng cần biết tình trạng các nút khác, các gói dữ liệu phát đi có tới đích hay không...

### 7.4.1. Các chức năng chính

ICMP là giao thức điều khiển của tầng IP, sử dụng để trao đổi các thông tin điều khiển dòng dữ liệu, thông báo lỗi và các thông tin trạng thái khác của bộ giao thức TCP/IP.

- Điều khiển lưu lượng (*Flow Control*): Khi các gói dữ liệu đến quá nhanh, thiết bị đích hoặc thiết bị định tuyến ở giữa sẽ gửi một thông điệp ICMP trở lại thiết bị gửi, yêu cầu thiết bị gửi tạm thời ngừng việc gửi dữ liệu.

- Thông báo lỗi: Trong trường hợp không tới được địa chỉ đích thì hệ thống sẽ gửi một thông báo lỗi "*Destination Unreachable*".

- Định hướng lại các tuyến (*Redirect Router*): Một Router gửi một thông điệp ICMP cho một trạm thông báo nên sử dụng Router khác. Thông điệp này có thể chỉ được dùng khi trạm nguồn ở trên cùng một mạng với hai thiết bị định tuyến.

- Kiểm tra các trạm ở xa: Một trạm có thể gửi một thông điệp ICMP "*Echo*" để kiểm tra trạm có hoạt động hay không.

**Bảng 7.13.** Bảng kiểm thông điệp của ICMP

Nhóm	Loại bản tin	Type
Thông điệp truy vấn (ICMP Queries)	Hỏi và phúc đáp Echo (Echo RequEst và Echo Reply)	8/0
	Hỏi và phúc đáp nhãn thời gian (Timestamp Request và Timestamp Reply)	13/14
	Yêu cầu và phúc đáp mặt nạ địa chỉ (Address mask request và address mask Reply)	17/18
	Yêu cầu quảng bá bộ định tuyến (Router soliciation và router advertisement)	10/9
Thông điệp báo lỗi (ICMP Error Reports)	Không thể đạt tới đích (Destination Urneachable)	3
	Yêu cầu ngừng hoặc giảm tốc độ phát (Source Quench)	4
	Định hướng lại (Redirection)	5
	Vượt ngưỡng thời gian (Time Exceeded)	11

## 7.4.2. Các loại thông điệp ICMP

Các thông điệp ICMP được chia thành hai nhóm: Các thông điệp truy vấn và các thông điệp thông báo lỗi. Các thông điệp truy vấn giúp cho người quản trị mạng nhận các thông tin xác định từ một node mạng khác. Các thông điệp thông báo lỗi liên quan đến các vấn đề mà bộ định tuyến hay trạm phát hiện ra khi xử lý gói IP. ICMP sử dụng địa chỉ IP nguồn để gửi thông điệp thông báo lỗi cho node nguồn của gói IP.

## 7.5. GIAO THỨC ĐỊNH TUYẾN

### 7.5.1. Khái quát về định tuyến

#### 7.5.1.1. Khái niệm

Định tuyến là một quá trình chọn lựa các đường đi trên một mạng máy tính để gửi dữ liệu qua đó.

Định tuyến chỉ ra hướng và đường đi tốt nhất từ nguồn đến đích của các gói tin (packer) thông qua các node trung gian là router.



### 7.5.1.2. Nguyên tắc định tuyến

Trong hoạt động định tuyến, người ta chia làm hai loại là định tuyến trực tiếp và định tuyến gián tiếp. Định tuyến trực tiếp là định tuyến giữa hai máy tính nối với nhau vào một mạng vật lý. Định tuyến gián tiếp là định tuyến giữa hai máy tính ở xa các mạng vật lý khác nhau nên chúng phải thực hiện thông qua các Gateway.

Để kiểm tra xem máy đích có nằm trên cùng một mạng vật lý với máy nguồn hay không thì người gửi phải tách lấy địa chỉ mạng của máy đích trong phần tiêu đề của gói dữ liệu và so sánh với phần địa chỉ mạng trong phần địa chỉ IP của nó. Nếu trùng thì gói tin sẽ được truyền trực tiếp, nếu không trùng cần phải xác định Gateway để truyền các gói này thông qua nó để ra mạng ngoài thích hợp.

### 7.5.2. Bảng định tuyến

Bảng định tuyến hay còn gọi là bảng chọn đường (*Routing table*). Các host và các router trên mạng Internet đều chứa một bảng định tuyến để tính toán các chặng tiếp theo cho gói tin. Bảng định tuyến này gán tương ứng mỗi địa chỉ đích với một địa chỉ Router cần đến ở chặng tiếp theo. Địa chỉ đích trong bảng định tuyến có thể bao gồm cả địa chỉ mạng, mạng con và hệ thống độc lập. Trong bảng định tuyến có thể bao gồm một tuyến mặc định được biểu diễn bằng địa chỉ 0.0.0.0.

Bảng định tuyến có thể tạo ra bởi người quản trị mạng hoặc từ sự thay đổi thông tin định tuyến giữa các router bằng các giao thức định tuyến động. Bảng định tuyến có rất nhiều dạng nhưng đơn giản và phổ biến nhất có thể diễn đạt được bằng mô hình mạng bao gồm các thông tin sau:

- Địa chỉ đích của mạng, mạng con và hệ thống độc lập.
- Địa chỉ IP của giao diện router kế tiếp phải đến.
- Giao tiếp vật lý trên router phải sử dụng để đến chặng kế tiếp.
- Mặt nạ mạng của địa chỉ đích.
- Khoảng cách quản trị.
- Thời gian (tính theo giây) từ khi router cập nhật.

### 7.5.3. Định tuyến tĩnh và định tuyến động

Định tuyến là quá trình mà router thực hiện để chuyển gói dữ liệu tới mạng đích. Tất cả các router dọc theo đường đi đều dựa vào địa chỉ IP đích của gói dữ liệu để chuyển gói theo đúng hướng đến đích cuối cùng. Để thực hiện được điều này, router phải học thông tin về đường đi tới các mạng khác. Nếu router chạy định tuyến động thì router tự động học những thông tin này từ các router khác. Còn nếu router chạy định tuyến tĩnh thì người quản trị mạng phải cấu hình các thông tin đến các mạng khác cho router.

Đối với định tuyến tĩnh, các thông tin về đường đi phải do người quản trị mạng nhập cho router. Khi cấu trúc mạng có bất kỳ thay đổi nào thì chính người quản trị mạng phải xoá hoặc thêm các thông tin về đường đi cho router. Những loại đường đi như vậy gọi là đường đi cố định. Đối với hệ thống mạng lớn thì công việc bảo trì mạng định tuyến cho router như trên tốn rất nhiều thời gian. Còn đối với hệ thống mạng nhỏ, ít có thay đổi thì công việc này đỡ mất công hơn. Chính vì định tuyến tĩnh đòi hỏi người quản trị mạng phải cấu hình mọi thông tin về đường đi cho router nên nó không có được tính linh hoạt như định tuyến động. Trong những hệ thống mạng lớn, định tuyến tĩnh thường được sử dụng kết hợp với giao thức định tuyến động cho một mục đích đặc biệt.

#### 7.5.3.1. Định tuyến tĩnh

##### • Hoạt động định tuyến tĩnh:

Hoạt động của định tuyến tĩnh có thể chia ra làm 3 bước như sau:

- Đầu tiên, người quản trị mạng cấu hình các đường cố định cho router.
- Router cài đặt các đường đi này vào bảng định tuyến.
- Gói dữ liệu được định tuyến theo các đường cố định này. Người quản trị mạng cấu hình đường cố định cho router bằng lệnh ip router.
- Cấu hình đường cố định.

##### • Sau đây là các bước để cấu hình đường cố định:

(1) Xác định tất cả các mạng đích cần cấu hình, subnet mask tương ứng và gateway tương ứng. Gateway có thể là cổng giao tiếp trên router hoặc là địa chỉ của trạm kế tiếp để đến được mạng đích.

(2) Vào chế độ cấu hình toàn cục của router.

(3) Nhập lệnh ip router với địa chỉ mạng đích, subnet mask tương ứng và gateway tương ứng mà bạn đã xác định ở bước 1. Nếu cần thì thêm thông số về chi số tin cậy.

(4) Lập lại bước 3 cho những mạng đích khác.

(5) Thoát khỏi chế độ cấu hình toàn cục.

(6) Lưu tập tin cấu hình đang hoạt động thành tập tin cấu hình khởi động bằng lệnh copy running - config statup - config.

(7) Kiểm tra cấu hình đường cố định.

Sau khi cấu hình đường cố định phải kiểm tra xem bảng định tuyến đã có đường, cố định cấu hình hay chưa, hoạt động định tuyến có đúng hay không. Bạn dùng lệnh show running - config để kiểm tra nội dung tập tin cấu hình đang chạy trên RAM xem câu lệnh cấu hình đường cố định đã được nhập vào đúng chưa. Sau đó dùng lệnh show ip router để xem có đường cố định trong bảng định tuyến hay không.

Sau đây là các bước kiểm tra cấu hình đường cố định:

– Ở chế độ đặc quyền, bạn nhập lệnh show running - config để xem tập tin cấu hình đang hoạt động.

– Kiểm tra xem câu lệnh cấu hình đường cố định có đúng không. Nếu không đúng thì phải vào lại chế độ cấu hình toàn cục, xoá câu lệnh sai đi và nhập lại câu lệnh mới.

– Nhập lệnh show ip router.

– Kiểm tra xem đường cố định mà bạn đã cấu hình có trong bảng định tuyến hay không.

### 7.5.3.2. Định tuyến động

Giao thức định tuyến khác với giao thức được định tuyến cả về chức năng và nhiệm vụ. Giao thức định tuyến được sử dụng để giao tiếp giữa các router với nhau. Giao thức định tuyến cho phép router này chia sẻ các thông tin định tuyến mà nó biết cho các router khác. Từ đó, các router có thể xây dựng và bảo trì bảng định tuyến của nó.

Sau đây là một số giao thức định tuyến:

• **RIP (Router Informatin Protocol)**

Sau đây là các đặc điểm chính của RIP:

- Là giao thức định tuyến theo vectơ khoảng cách.
- Sử dụng số lượng hop để làm thông số chọn đường đi.
- Nếu số lượng hop để tới đích lớn hơn 15 thì gói dữ liệu sẽ bị hủy bỏ.
- Cập nhật theo định kỳ mặc định là 30 giây.

• **IGRP (Interior Gateway Routing Protocol)** là giao thức được phát triển độc quyền bởi Cisco.

Sau đây là một số đặc điểm mạnh của IGRP:

- Là giao thức định tuyến theo vectơ khoảng cách.
- Sử dụng băng thông, tải, độ trễ và độ tin cậy của đường truyền làm thông số lựa chọn đường đi.

– Cập nhật theo định kỳ mặc định là 90 giây.

• **EIGRP (Enhanced Interior Gateway Routing Protocol)** là giao thức định tuyến nâng cao theo vectơ khoảng cách và là giao thức độc quyền của Cisco.

Sau đây là các đặc điểm chính của EIGRP:

- Là giao thức định tuyến nâng cao theo vectơ khoảng cách.
- Có chia tải.
- Có các ưu điểm của định tuyến theo vectơ khoảng cách và định tuyến theo trạng thái đường liên kết.
- Sử dụng thuật toán DUAL (*Diffused Update Algorithm*) để tính toán chọn đường tốt nhất. Cập nhật theo định kỳ mặc định là 90 giây hoặc cập nhật khi có thay đổi về cấu trúc mạng.

• **OSPF (Open Shortest Path First)** là giao thức định tuyến theo trạng thái đường liên kết.

Sau đây là các đặc điểm chính của OSPF:

- Là giao thức định tuyến theo trạng thái đường liên kết.



- Được định nghĩa trong RFC 2328.
- Sử dụng thuật toán SPF để tính toán chọn đường đi tốt nhất.
- Chỉ cập nhật khi cấu trúc mạng có sự thay đổi.
- **BGP** (*Border Gateway Protocol*) là giao thức định tuyến ngoại theo vectơ khoảng cách.

Sau đây là các đặc điểm chính của BGP:

- Được sử dụng để định tuyến giữa các ISP hoặc giữa ISP và khách hàng.
- Được sử dụng để định tuyến lưu lượng Internet giữa các hệ tự quản (AS). Còn giao thức được định tuyến thì được sử dụng để định hướng cho dữ liệu của người dùng. Một giao thức được định tuyến sẽ cung cấp đầy đủ thông tin về địa chỉ lớp mạng để gói dữ liệu có thể truyền đi từ host này đến host khác dựa trên cấu trúc địa chỉ đó.

## 7.5.4. Một số thuật toán định tuyến

### 7.5.4.1. Định tuyến theo vectơ khoảng cách

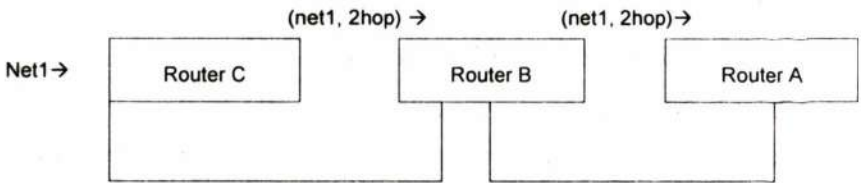
Giải thuật vectơ khoảng cách dựa trên thuật toán Bellman Ford là một phương pháp định tuyến đơn giản, hiệu quả và được sử dụng trong nhiều giao thức định tuyến như giao thức định tuyến Rip, OSPF.

Vectơ khoảng cách được thiết kế để giảm tối đa sự liên lạc giữa các router cũng như lượng dữ liệu trong bảng định tuyến. Bản chất của định tuyến vectơ khoảng cách là một router không cần biết tất cả các đường đi đến phân đoạn mạng, nó chỉ cần biết phải truyền một datagram được gán địa chỉ đến một phân đoạn mạng đi theo hướng nào. Khoảng cách giữa các phân đoạn mạng này đến phân đoạn mạng kia. Router sử dụng thuật toán vectơ khoảng cách để tối ưu hoá đường đi bằng cách giảm tối đa số lượng router mà datagram đi qua. Tham số khoảng cách này chính là số chặng phải qua (hop count).

Định tuyến vectơ khoảng cách dựa trên quan niệm rằng một router sẽ thông báo cho các router lân cận nó về tất cả các mạng nó biết và khoảng cách đến mỗi mạng này. Một router chạy giao thức định tuyến vectơ khoảng cách sẽ thông báo đến các router kế cận được kết nối trực tiếp với nó một

hoặc nhiều hơn các vectơ khoảng cách. Một vectơ khoảng cách bao gồm một bộ (network, cost) với network là mạng đích và cost là một giá trị có liên quan nó biểu diễn số các router hoặc link trong đường dẫn giữa các router thông báo và mạng đích. Do đó, cơ sở dữ liệu định tuyến bao gồm một số vectơ khoảng cách hoặc cost đến tất cả các mạng từ router đó.

Khi một router thu được bản tin cập nhật vectơ khoảng cách từ router kế cận nó thì nó bổ sung giá trị cost của chính nó (thường bằng 1) vào giá trị cost thu được trong bản tin cập nhật. Sau đó router so sánh giá trị cost tính được này với thông tin thu được trong bản tin cập nhật trước đó. Nếu cost nhỏ hơn thì router cập nhật cơ sở dữ liệu định tuyến với các cost mới, tính toán một bảng định tuyến mới, nó bao gồm các router kế cận vừa thông báo thông tin vectơ khoảng cách mới next hop.



**Hình 7.15.** Định tuyến vectơ khoảng cách

Router C thông báo một vectơ khoảng cách (net1, 1hop) cho mạng đích net1 được nối trực tiếp với nó. Router B thu được vectơ khoảng cách này thực hiện bổ sung cost của nó (1hop) và thông báo cho router A(net1, 2hop). Nhờ đó router A biết rằng nó có thể đạt tới net1 với 2hop và qua router B.

#### 7.5.4.2. Định tuyến theo trạng thái liên kết

Định tuyến vectơ khoảng cách sẽ không còn phù hợp đối với các mạng lớn gồm rất nhiều router. Khi đó, mỗi router phải duy trì một mục trong bảng định tuyến cho mỗi đích, và các mục này chỉ đơn thuần chứa các giá trị vectơ và hop count. Router cũng không thể tiết kiệm năng lực của mình khi đã biết nhiều về cấu trúc mạng. Hơn nữa, toàn bộ bảng giá trị khoảng cách và hop count phải được truyền giữa các router cho dù hầu hết các thông tin này không thực sự cần thiết trao đổi giữa các router. Định tuyến trạng thái liên kết ra đời đã khắc phục được các nhược điểm của định tuyến vectơ khoảng cách.

Bản chất của định tuyến trạng thái liên kết là mỗi router xây dựng bên trong nó một sơ đồ cấu trúc mạng. Định kỳ, mỗi router cũng gửi ra mạng những thông điệp trạng thái. Những thông điệp này liệt kê những router khác trên mạng kết nối trực tiếp với router đang xét và trạng thái liên kết. Các router sử dụng bản tin trạng thái nhận được từ các router khác để xây dựng sơ đồ mạng. Khi một router chuyển tiếp dữ liệu, nó sẽ chọn đường đi đến đích tốt nhất dựa trên những điều kiện hiện tại.

Giao thức trạng thái liên kết đòi hỏi nhiều thời gian xử lý trên mỗi router nhưng giảm được sự tiêu thụ băng thông vì mỗi router không cần gửi toàn bộ bảng định tuyến của mình. Hơn nữa, router cũng không dễ dàng theo dõi lỗi trên mạng vì bản tin trạng thái từ một router không thay đổi khi lan truyền trên mạng (ngược lại với phương pháp vectơ khoảng cách, giá trị hop count tăng lên mỗi khi thông tin định tuyến đi qua một router khác).

Định tuyến trạng thái liên kết làm việc trên quan điểm rằng một router, có thể thông báo với mọi router khác trong mạng trạng thái của các tuyến được kết nối đến nó, cost của các tuyến đó và xác định bất kỳ một router kế cận nào được nối với các tuyến này. Các router chạy một giao thức định tuyến trạng thái đường sẽ truyền bá các gói trạng thái LSP (*Link State Paket*) khắp mạng. Một LSP nói chung chứa một xác định nguồn, xác định kế cận và cost của tuyến giữa chúng. Các LSP được thu bởi tất cả các router được sử dụng để tạo nên một cơ sở dữ liệu cấu hình của toàn bộ mạng. Bảng định tuyến sau đó được tính toán dựa trên nội dung của cơ sở dữ liệu cấu hình. Tất cả các router trong mạng chứa một sơ đồ của cấu hình mạng và từ đó chúng tính toán đường ngắn nhất (*Least-cost path*) từ nguồn bất kỳ đến đích bất kỳ. Giá trị gắn với các link giữa các router là cost của link đó. Các router truyền bá các LSP đến tất cả các router khác trong mạng, nó được sử dụng để xây dựng cơ sở dữ liệu trạng thái đường. Tiếp theo, mỗi router trong mạng tính toán một cây bắt nguồn từ chính nó và phân nhánh đến tất cả các router khác dựa trên tiêu chí đường ngắn nhất hay có chi phí ít nhất.

### 7.5.5. So sánh các thuật toán định tuyến

Các giao thức định tuyến với thuật toán vectơ tỏ ra đơn giản và hiệu quả trong các mạng nhỏ và đòi hỏi ít (nếu có) sự giám sát. Tuy nhiên, chúng



không làm việc tốt, và có tài nguyên tập hợp ít ỏi, dẫn đến sự phát triển của các thuật toán trạng thái kết nối tuy phức tạp hơn nhưng tốt hơn dùng trong các mạng lớn. Giao thức vectơ kém hơn với rắc rối về đếm đến vô tận.

Ưu điểm chính của định tuyến bằng trạng thái kết nối là phản ứng nhanh nhạy hơn và trong một khoảng thời gian có hạn, đối với sự thay đổi kết nối.

Ngoài ra, những gói được gửi qua mạng trong định tuyến bằng trạng thái kết nối thì nhỏ hơn những gói dùng trong định tuyến bằng vectơ. Định tuyến bằng vectơ đòi hỏi bảng định tuyến đầy đủ phải được truyền đi, trong khi định tuyến bằng trạng thái kết nối thì chỉ có thông tin về "hàng xóm" của node được truyền đi.

Vì vậy, các gói tin này dùng tài nguyên mạng ở mức độ không đáng kể. Khuyết điểm chính của định tuyến bằng trạng thái kết nối là nó đòi hỏi nhiều sự lưu trữ và tính toán để chạy hơn định tuyến bằng vectơ.

## 7.5.6. Một số giao thức định tuyến ứng dụng thuật toán

### 7.5.6.1. Giao thức định tuyến RIP

#### a. Khái niệm

RIP (*Router Information Protocol – Giao thức thông tin định tuyến*) là một giao thức định tuyến miền trong được sử dụng bên trong hệ thống tự trị. Đây là một giao thức rất đơn giản dựa trên định tuyến vectơ khoảng cách, sử dụng giải thuật Bellman – Ford để tính toán bảng định tuyến.

Khi được sử dụng trong những mạng cùng loại nhỏ, RIP là một giao thức hiệu quả và sự vận hành của nó là khá đơn giản. RIP duy trì tất cả bảng định tuyến trong một mạng được cập nhật bởi truyền những bản tin cập nhật bảng định tuyến sau mỗi 30s. Sau một thiết bị RIP nhận một cập nhật, nó so sánh thông tin hiện tại của nó với những thông tin được chứa trong thông tin cập nhật RIP có hai phiên bản:

RIP phiên bản 1 RIPv1 (RIP version 1): RIPv1 là giao thức định tuyến phân lớp, không có thông tin về mặt nạ mạng con và không hỗ trợ định tuyến liên vùng không phân lớp CIDR (*Classless Interdomain routing*).

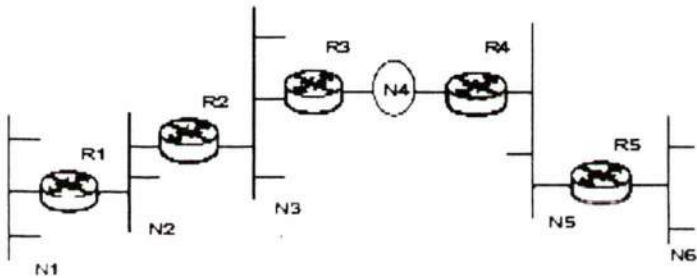


chiều dài biến của mặt nạ mạng con VLSM (*Variable-length Subnet Mask*). RIPv1 sử dụng địa chỉ quảng bá. RIPv1 được xác định trong RFC 1058 "*Routing Information Protocol*" năm 1988.

RIP phiên bản 2 RIPv1 (RIP version 2): RIPv2 là giao thức định tuyến không phân lớp, có thông tin về mặt nạ mạng con và hỗ trợ cho CIDR, VLSM. RIPv2 sử dụng địa chỉ đa hướng. RIPv2 được xác định đầu tiên trong các RFC sau: RFC1387 "*RIP Version 2 Protocol Analysis*" năm 1993, RFC1388 "*RIP Version 2 Carrying Additional Information*" năm 1993 và RFC1389 "*RIP Version 2 MIB Extensions*" năm 1993.

**b. Thuật toán và ví dụ minh họa**

Mỗi router có một bảng định tuyến trong đó chứa các mục tương ứng cho mỗi mạng đích mà router biết. Mục này gồm địa chỉ IP của mạng đích, khoảng cách ngắn nhất để tới đích (tính theo số bước nhảy) và bước nhảy tiếp theo (router tiếp theo). Bước nhảy tiếp theo là nơi cần gửi gói dữ liệu đến để có thể tới được đích cuối cùng. Số bước nhảy là số mạng mà một gói dữ liệu phải đi qua để tới được mạng đích.



Hình 7.16. Định tuyến trong mạng sử dụng RIP

Bảng 7.14. Định tuyến RIP được cập nhật khi router nhận được các thông báo RIP

Bảng của R2			Bảng của R3			Bảng của R4		
Đích	Node sau	Hop	Đích	Node sau	Hop	Đích	Node sau	Hop
N1	R1	2	N1	R2	3	N1	R3	4

Bảng của R2			Bảng của R3			Bảng của R4		
Đích	Node sau	Hop	Đích	Node sau	Hop	Đích	Node sau	Hop
N2	Trực tiếp	1	N2	R2	2	N2	R3	3
N3	Trực tiếp	1	N3	Trực tiếp	1	N3	R3	2
N4	R3	2	N4	Trực tiếp	1	N4	Trực tiếp	1
N5	R3	3	N5	R4	2	N5	Trực tiếp	1
N6	4	N6	R4	R4	3	N6	R5	2

Dưới đây chỉ ra giải thuật cập nhật định tuyến được RIP sử dụng.

Nhận một thông báo RIP trả lời 1.

(1) Cộng 1 vào số bước nhảy tiếp theo cho mỗi đích được quảng cáo.

(2) Lập lại các bước tiếp theo cho mỗi đích được quảng cáo:

(2.1). Nếu đích không có trong bảng định tuyến

Thêm thông tin được quảng cáo vào bảng định tuyến

(2.2). Trái lại

(2.2.1) Nếu bước nhảy tiếp theo giống nhau

Thay thế mục trong bảng bằng mục được quảng cáo.

(2.2.2) Trái lại

(2.2.2.1)

– Nếu số bước nhảy được quảng cáo nhỏ hơn số bước nhảy trong bảng.

– Thay thế mục trong bảng bằng mục được quảng cáo

(2.2.2.2) Trái lại

– Không làm gì cả.

(3) Kết thúc

### c. Chi tiết về giao thức định tuyến RIP

- RIP phiên bản 1 (RIPv1)

- Đặc điểm:

RIPv1 là một giao thức định tuyến theo vectơ khoảng cách nên nó quang bá (theo địa chỉ 255.255.255.255) toàn bộ bảng định tuyến của nó cho các bộ định tuyến lân cận theo định kỳ. Chu kỳ cập nhật của RIP là 30 giây. Thông số định tuyến của RIP là số lượng hop, giá trị tối đa là 15 hop nếu lớn hơn thì gói dữ liệu đó sẽ bị huy bỏ. Thời gian giữ chậm cho một tuyến là 180 giây, nếu lớn hơn thì tuyến này coi như là hết hạn.

RIPv1 là giao thức định tuyến được sử dụng phổ biến vì mọi bộ định tuyến IP đều có hỗ trợ giao thức này. RIPv1 được phổ biến vì tính đơn giản và tính tương thích toàn cầu của nó. RIPv1 có thể chia tài ra tối đa là 6 đường có chi phí bằng nhau (mặc định là 4 đường).

RIPv1 là giao thức định tuyến theo lớp địa chỉ. Khi RIP bộ định tuyến nhận thông tin về một mạng nào đó từ một công, trong thông tin định tuyến này không có thông tin về mặt nạ mạng con đi kèm. Do đó bộ định tuyến sẽ lấy mặt nạ mạng con của công để áp dụng cho địa chỉ mạng mà nó nhận được từ công này. Nếu mặt nạ mạng con này không phù hợp thì nó sẽ lấy mặt nạ mạng con mặc định theo địa chỉ áp dụng cho địa chỉ mạng mà nó nhận được:

- Địa chỉ lớp A có mặt nạ mạng con mặc định là 255.0.0.0.
- Địa chỉ lớp B có mặt nạ mạng con mặc định là 255.255.0.0.
- Địa chỉ lớp C có mặt nạ mạng con mặc định là 255.255.255.0.

Do RIPv1 là một giao thức định tuyến theo vectơ khoảng cách nên nó sử dụng cơ chế đường cắt ngang để chống lặp vòng.

Cấu trúc ban tin:

1-octet command field	1-octet version number field	2-octet zero field	2-octet AFI field	2-octet zero field	4 octet IP Address field	4-octet zero field	4-octet zero field	4 octet metric field
-----------------------------	---------------------------------------	--------------------------	-------------------------	--------------------------	--------------------------------	--------------------------	--------------------------	----------------------------

- Các trường chức năng trong gói tin IP RIP:

+ Command: Cho ta biết gói tin là gói tin yêu cầu (Request) hay gói tin trả lời (Response). Gói tin Request sẽ đưa ra yêu cầu cho một bảng định

tuyến gửi tất cả hay 1 phần bảng định tuyến của nó. Gói tin Response được đưa ra khi 1 bộ định tuyến nhận được gói tin Request. Nhiều gói tin RIP có thể được sử dụng để vận chuyển cho một bảng định tuyến lớn.

+ Version number: Chỉ ra phiên bản RIP đang sử dụng. Trường này dùng các ký hiệu khác nhau để chỉ ra các phiên bản khác nhau đang được sử dụng trong mạng.

+ Zero: Trường này thực tế không sử dụng, nó được thêm vào để cung cấp tính tương thích về sau cho các chuẩn của RIP. Trường này có thể được thiết lập mặc định giá trị 0.

+ AFI (*Address-Family Identifier*): Chỉ ra kiểu địa chỉ được sử dụng để cấu hình mạng. Do RIP được thiết kế để mang thông tin định tuyến cho nhiều các giao thức khác nhau nên mỗi loại sẽ có 1 nhận dạng riêng cho ta biết kiểu địa chỉ mà giao thức đang sử dụng. Giá trị AFI cho IP là 2.

+ Address: Chỉ ra địa chỉ IP của các bộ định tuyến.

+ Metric: Cho ta biết có bao nhiêu bước liên mạng (Internetwork hop) đã đi qua trong hành trình đến đích. Giá trị này sẽ nằm trong khoảng 1 đến 15 cho các đường đi còn hiệu lực và 16 cho các đường đi không thể thực hiện được bởi RIP.

– Các bộ định thời của RIPv1:

Để hỗ trợ cho hoạt động của hệ thống, RIP sử dụng 3 bộ định thời bao gồm: Bộ định thời định kỳ (*periodic*) điều khiển việc gửi thông báo, bộ định thời hết hạn (*expiration*) quản lý tính hợp lệ của một tuyến và bộ định thời thu lượm rác (*garbage collection*) quảng bá lỗi của một tuyến:

+ Bộ định thời định kỳ: Bộ định thời này điều khiển việc quảng bá đều đặn các thông báo cập nhật. Mặc dù giao thức đã chỉ rõ mỗi bộ định thời này phải được đặt là 30 giây, nhưng các mô hình đang hoạt động hiện nay thường sử dụng một số ngẫu nhiên trong khoảng từ 25 đến 35 giây với mục đích để tránh tình trạng quá tải trên một liên kết mạng khi tất cả các bộ định tuyến gửi cập nhật cùng lúc. Bộ định thời này được đếm lùi. Khi đạt đến giá trị 0, thông báo cập nhật sẽ được gửi và bộ định thời lại được thiết lập lại.

+ Bộ định thời hết hạn: Bộ định thời này quản lý tính hợp lệ của một tuyến. Khi bộ định tuyến nhận được thông tin cập nhật về một tuyến, bộ



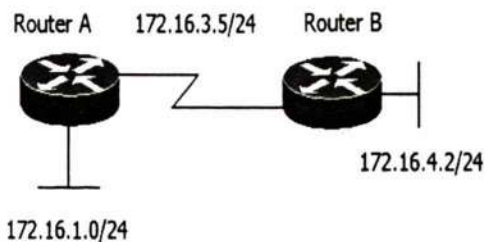
định thời hết hạn cho tuyến này sẽ được thiết lập là 180 giây. Mỗi lần có một cập nhật mới về tuyến này bộ định thời được đặt lại. Trong trường hợp bình thường thì cứ 30 giây điều này xảy ra một lần. Tuy nhiên, nếu có trục trặc trên liên mạng và bộ định tuyến không nhận được cập nhật về tuyến này trong khoảng thời gian 180 giây, tuyến này được xem như là hết hạn và giá trị trường bước nhảy của nó được đặt là 16, nghĩa là không thể đến đích. Mỗi tuyến đều có bộ định thời hết hạn của riêng mình.

+ Bộ định thời thu lượm rác: Khi một tuyến hết hạn, bộ định tuyến không loại bỏ ngay tuyến này ra khỏi bảng định tuyến. Thay vào đó, nó tiếp tục quảng bá tuyến này với giá trị đo lường là 16. Cùng lúc đó, bộ định thời thu lượm rác được đặt là 120 giây cho tuyến này. Khi giá trị của bộ định thời này đạt tới 0, tuyến bị loại khỏi bảng định tuyến. Bộ định thời này cho phép các lần cận biết về sự không hợp lệ của một tuyến trước khi loại tuyến ra khỏi bảng định tuyến.

– Thiết kế RIPv1:

Một số điều cần nhớ trong thiết kế mạng với RIPv1 là nó không hỗ trợ VLSM hoặc CIDR. Lược đồ địa chỉ IP với RIPv1 yêu cầu mặt nạ mạng con giống nhau cho mỗi thực thể mạng IP, 1 mạng IP bằng phẳng. Giới hạn số hop trong RIPv1 là 15. Vì vậy kích thước mạng không thể vượt quá số giới hạn đó. RIPv1 cũng quảng bá bảng định tuyến của nó 30 giây một lần. RIPv1 thường có giới hạn khi truy nhập vào mạng nơi mà giao thức này có thể hoạt động liên kết với các máy chủ được thực hiện định tuyến.

Như trong hình 7.17, khi sử dụng RIPv1, tất cả các địa chỉ trong mạng phải có cùng mặt nạ mạng con.



Hình 7.17. Các địa chỉ phải có cùng mặt nạ mạng con

- RIP phiên bản 2 (RIPv2):

- Đặc điểm:

RIPv2 là bản được phát triển từ RIPv1 nên nó có các đặc điểm như RIPv1:

- + Là một giao thức định tuyến theo vectơ khoảng cách, sử dụng số lượng bước nhảy làm thông số định tuyến.

- + Giá trị hop tối đa là 15.

- + Thời gian giữ chậm cũng là 180 giây.

- + Sử dụng cơ chế chia rẽ tầng để chống lặp vòng.

- + RIPv2 đã khắc phục được những điểm giới hạn của RIPv1.

- + RIPv2 có gửi mật nạ mạng con đi kèm với các địa chỉ mạng trong thông tin định tuyến. Nhờ đó mà RIPv2 có thể hỗ trợ VLSM và CIDR.

- + RIPv2 có hỗ trợ việc xác minh thông tin định tuyến.

- + RIPv2 gửi thông tin định tuyến theo địa chỉ đa hướng 244.0.0.9.

- Cấu trúc bản tin:

1-octet command field	1-octet version number field	2-octet unused field	2-octet AFI field	2-octet route tag field	4-octet network address field	4-octet subnet mask field	4-octet next hop field	4-octet metric field
-----------------------------	---------------------------------------	----------------------------	-------------------------	----------------------------------	--	------------------------------------	---------------------------------	----------------------------

Bản tin IP RIPv2 cho phép mang nhiều thông tin hơn ngoài các thông tin như trong bản tin IP RIP nó còn cung cấp một cơ chế xác thực không được hỗ trợ bởi RIP.

- Một số đặc tính sau đây là những dấu hiệu lớn nhất được bổ sung vào RIPv2:

- + Sự nhận thực của dòng tin truyền dẫn.

- + Hỗ trợ mật nạ con.

- + Địa chỉ IP bước kế tiếp.

- + Bản tin đa phương RIP-2.

Một số hỗ trợ khác gồm có sự gia tăng khối thông tin quản lý và hỗ trợ cho các thẻ của bộ định tuyến ngoài mạng.

- Các trường chức năng trong định dạng bản tin IP RIPv2:
  - + Command, Version number, AFI, Address, Metric: Chức năng của chúng cũng giống như trong bản tin IP RIP.
  - + Unused: Có giá trị được thiết lập mặc định là 0.
  - + Route tag (Nhãn đường đi): Cung cấp một phương thức phân biệt giữa bộ định tuyến nội bộ (sử dụng giao thức RIP) và các bộ định tuyến ngoài (sử dụng các giao thức định tuyến khác).
  - + Subnet mask: Chứa đựng mặt nạ mạng con cho các bộ định tuyến.
  - + Next hop: Cho biết địa chỉ IP của bước đi tiếp mà gói tin có thể chuyển tiếp.

Trong RIP phiên bản 2, kiểu bản tin xác thực được thêm vào để bảo vệ bản tin thông báo. Tuy nhiên, không cần thêm các trường mới vào thông báo. Mục đầu tiên của thông báo sẽ chứa thông tin xác thực.

Để chỉ rõ một mục chứa thông báo xác thực chứ không phải là thông tin định tuyến, giá trị hexa FFFF được đặt trong trường AFI. Trường tiếp theo trong thông báo xác thực đó là loại xác thực, dùng để định nghĩa phương pháp sử dụng để xác thực. Trường cuối cùng trong thông báo xác thực là để chứa dữ liệu xác thực. Định dạng của bản tin xác thực như bảng 7.15.

**Bảng 7.15.** Định dạng bản tin

1 - octet command field	1 - octet version number field	2 - octet unused field	2 - octet AFI field	2 - octet Authentication type field	16 - octet Data field
2 - octet AFI = 0xFFFF File		2 - octet Authentication		16 - octet Password field	

Thông tin xác thực được thêm trường AFI. Ngoài ra RIP phiên bản 2 còn hỗ trợ phát đa hướng (Multicast) so với phiên bản 1. RIP phiên bản 1 sử dụng phát quảng bá để gửi các thông báo RIP tới tất cả các bộ định tuyến lân cận. Do đó, không chỉ các bộ định tuyến trên mạng nhận được thông báo mà mọi trạm trong mạng đều có thể nhận được. Trong khi đó, RIP phiên bản 2 sử dụng địa chỉ đa hướng 224.0.0.9 để phát đa hướng các thông báo RIP tới chỉ các bộ định tuyến sử dụng giao thức RIP trên một mạng mà thôi.

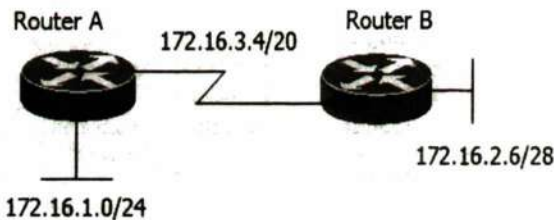
– Các bộ định thời:

Những bộ định thời của RIPv2 cũng giống như RIPv1. Để hỗ trợ cho hoạt động của hệ thống, RIP sử dụng 3 bộ định thời bao gồm: Bộ định thời định kỳ (periodic) điều khiển việc gửi thông báo, bộ định thời hết hạn (expiration) quản lý tính hợp lệ của một tuyến và bộ định thời thu lượm rác (garbage collection) quảng bá lỗi của một tuyến.

– Thiết kế RIPv2:

Một số điều cần ghi nhớ trong việc thiết kế mạng với RIPv2 là nó hỗ trợ VLSM bên trong mạng và CIDR để tóm tắt những mạng gần kề ở bên kia. RIPv2 cho phép tóm tắt các lộ trình trong cùng 1 mạng. RIPv2 vẫn có giới hạn số hop là 16. Vì thế kích thước mạng không thể vượt quá giới hạn này. RIPv2 gửi bảng định tuyến 30s mỗi lần đến các máy để gửi địa chỉ IP là 224.0.0.9. RIPv2 thường có giới hạn khi truy nhập vào mạng nơi mà giao thức này có thể hoạt động liên kết với các máy chủ được thực hiện định tuyến. RIPv2 cũng cung cấp sự xác nhận lộ trình.

Như trong hình 7.18, khi sử dụng RIPv2, tất cả các địa chỉ trong mạng có thể có những mặt nạ mạng con khác nhau.



**Hình 7.18.** Các địa chỉ có thể có mặt nạ mạng con khác nhau

#### • So sánh giao thức RIP 1 và RIP2

– Những điểm giống nhau:

+ Là giao thức định tuyến theo vector khoảng cách.

+ Sử dụng số hop làm thông số định tuyến và chu kỳ cập nhật mặc định là 30 giây.



- + Sử dụng thời gian giữ chậm để chống lặp vòng, thời gian này mặc định là 180 giây.
- + Sử dụng cơ chế cắt ngang để chống lặp vòng.
- + Nếu gói dữ liệu đến mạng đích có số lượng hop lớn hơn 15 thì gói dữ liệu đó sẽ bị huỷ bỏ.
- + Cùng giữ những thông tin sau về mỗi đích:
  - ✓ IP address: địa chỉ của máy đích hoặc mạng.
  - ✓ Gateway: Cổng vào/ra đầu tiên mà đường dẫn tiến về đích.
  - ✓ Interface: Phần mạng vật lý mà sử dụng để đến cổng ra đầu tiên của đường dẫn về đích.
  - ✓ Metric: Là số cho biết số hop đến đích.
  - ✓ Timer: Là lượng thời gian kể từ khi bộ định tuyến cập nhật lần cuối cùng.
- Những điểm khác nhau:

**Bảng 7.16.** Bảng so sánh những điểm khác nhau giữa RIPv1 và RIPv2

<i>RIP version 1 – RIPv1</i>	<i>RIP version 2 – RIPv2</i>
Định tuyến theo lớp địa chỉ.	Định tuyến không theo lớp địa chỉ.
Không gửi thông tin về mặt nạ mạng con trong thông tin định tuyến.	Có gửi thông tin về mặt nạ mạng con trong thông tin định tuyến.
Không hỗ trợ VLSM. Do đó tất cả các mạng trong hệ thống RIPv1 phải có cùng mặt nạ mạng con.	Có hỗ trợ VLSM. Do vậy các mạng trong hệ thống RIPv2 có thể có chiều dài mặt nạ mạng con khác nhau.
Không hỗ trợ CIDR.	Có hỗ trợ CIDR.
Không có cơ chế xác minh thông tin định tuyến.	Có cơ chế xác minh thông tin định tuyến.
Gửi quảng bá thông tin định tuyến theo địa chỉ: 255.255.255.255	Gửi thông tin định tuyến theo địa đa hướng 224.0.0.9 nên hiệu quả hơn.
Cùng giữ những thông tin giống nhau về đích nhưng RIPv1 không giữ được thông tin về mặt nạ mạng con còn RIPv2 giữ được thông tin về mặt nạ mạng con.	

### • Ứng dụng

– Giới hạn chung cho cả 2 RIPv1 và RIPv2:

+ Một điều bất lợi của các bộ định tuyến khi dùng RIP là chúng kết nối liên tục với các bộ định tuyến lân cận để cập nhật các bảng định tuyến của chúng, do đó tạo ra một lượng tải lớn trên mạng.

+ Đặc tính của RIP là các gói giới hạn dưới 15 hop và bảng định tuyến được trao đổi với các bộ định tuyến khác khoảng 30 giây/lần. Nếu một bộ định tuyến không thông báo trong vòng 180 giây, đường đi qua bộ định tuyến này được xem như không dùng được. Các vấn đề có thể xảy ra trong lúc tạo lại bảng định tuyến nếu bộ định tuyến này được kết nối với một mạng diện rộng chạy chậm. Hơn nữa, trao đổi các bảng làm mạng thường xuyên quá tải, gây tắc nghẽn và các trì hoãn khác.

+ Do RIP là giao thức định tuyến theo vectơ khoảng cách nên mỗi bộ định tuyến nhận được bảng định tuyến của những bộ định tuyến lân cận kết nối trực tiếp với nó, do vậy bộ định tuyến sẽ không biết được chính xác cấu trúc của toàn bộ hệ thống mạng.

+ RIP sử dụng thuật toán định tuyến theo vectơ khoảng cách. Nếu có nhiều đường đến cùng một lúc tới đích thì RIP sẽ chọn đường có số hop ít nhất. Chính vì vậy, dựa vào số lượng hop để chọn đường nên đôi khi con đường mà RIP chọn không phải là đường ngắn nhất và nhanh nhất tới đích.

+ Không dùng cho những mạng lớn hay phức tạp.

– Những giới hạn riêng của RIPv1 và RIPv2.

+ Những giới hạn của RIPv1:

- ✓ Không gửi thông tin mật nạ mạng con trong thông tin định tuyến.
- ✓ Không hỗ trợ xác minh thông tin nhận được.
- ✓ Gửi quảng bá thông tin định tuyến theo địa chỉ 255.255.255.255 nên nó sẽ gửi tới tất cả các bộ định tuyến lân cận (tức là không chỉ có bộ định tuyến trên mạng nhận được mà mọi trạm trong mạng để có thể nhận được).

- ✓ Không hỗ trợ VLSM (Variable Length Subnet Masking – Subnet Mask có chiều dài khác nhau) và CIDR (*Classless Interdomain Routing – Tuyến liên vùng không phân lớp*).
- + Sang đến phiên bản 2 thì RIPv2 đã giải quyết các giới hạn của RIPv1:
  - ✓ Có gửi thông tin mật nạ mạng con đi kèm với các địa chỉ mạng trong thông tin định tuyến.
  - ✓ Có hỗ trợ việc xác minh thông tin định tuyến.
  - ✓ Sử dụng địa chỉ đa hướng 224.0.0.9 để phát đa hướng các thông báo RIP tới chỉ các bộ định tuyến sử dụng giao thức RIP trên một mạng.
  - ✓ Nhờ có gửi thông tin mật nạ mạng con trên nên RIPv2 có thể hỗ trợ VLSM và CIDR.

#### • Bảo mật

Do có thể khai thác những điểm yếu trong bảo mật của RIP, do vậy chúng ta đề xuất một cách bảo vệ mới để giao thức định tuyến theo khoảng cách vectơ mang tên là SRIP (*A Secure Distance Vector Routing Protocol*).

- Những mục tiêu của S-RIP bao gồm:
  - Ngăn chặn những bộ định tuyến giả mạo.
  - Ngăn chặn sự giả mạo quyền hạn.
  - Ngăn chặn gian lận khoảng cách (cả ngắn và dài). Sự gian lận có thể thực hiện bởi những nút riêng lẻ hoặc những nút đã bị giả mạo.

#### • Ứng dụng:

– RIP được thiết kế như là một giao thức IGP (Interior Gateway Protocol là giao thức định tuyến nội miền) dùng cho các hệ thống tự trị AS (*AS Autonomously system*) có kích thước nhỏ, RIP chỉ áp dụng cho những mạng nhỏ, không sử dụng cho hệ thống mạng lớn và phức tạp. Bởi vì:

+ RIP sử dụng giao thức định tuyến theo vectơ khoảng cách thường tốn ít tài nguyên hệ thống nhưng tốc độ đồng bộ giữa các bộ định tuyến lại chậm mà đối với 1 mạng lớn hay phức tạp thì lại gồm nhiều bộ định tuyến nên RIP không phù hợp với những hệ thống mạng lớn và phức tạp.

+ RIP giới hạn số hop tối đa là 15 (bất kỳ mạng đích nào mà có số hop lớn hơn 15 thì xem như mạng đó không đến được). Số lượng 15 hop sẽ không đủ khi muốn xây dựng một mạng lớn.

+ Khi cấu trúc mạng thay đổi thì thông tin cập nhật phải được xử lý trong toàn bộ hệ thống, nên điều này sẽ thực hiện rất khó đối với mạng lớn vì sẽ rất dễ gây ra hiện tượng tắc nghẽn trong mạng.

+ Do sử dụng thuật toán định tuyến theo vectơ khoảng cách nên có tốc độ hội tụ chậm (Trạng thái hội tụ là tất cả các bộ định tuyến trong hệ thống mạng đều có thông tin định tuyến về hệ thống mạng và chính xác), do vậy đối với mạng lớn hay phức tạp thì sẽ mất rất lâu mới hội tụ được.

### 7.5.6.2. Giao thức OSPF

#### a. Khái niệm

Giao thức ưu tiên đường đi ngắn nhất (*OSPF – Open Shortest Path First*) là một giao thức định tuyến công trong khác được sử dụng rất rộng rãi. Phạm vi hoạt động của nó cũng là một hệ thống tự trị (AS). Các router đặc biệt được gọi là router biên AS có trách nhiệm ngăn thông tin về các hệ thống tự trị – AS khác vào trong hệ thống hiện tại. Để định tuyến hiệu quả OSPF chia hệ thống tự trị thành các vùng nhỏ.

Giao thức OSPF là giao thức định tuyến trạng thái liên kết, được thiết kế cho các mạng lớn hoặc các mạng liên hợp và phức tạp. Các giải thuật định tuyến trạng thái sử dụng các giải thuật SPE (*Shortest Path First*) cùng với một cơ sở dữ liệu phức tạp về cấu hình của mạng. Cơ sở dữ liệu về cấu hình mạng về cơ bản bao gồm tất cả các dữ liệu có liên kết đến bộ định tuyến chứa cơ sở dữ liệu.

#### b. Thuật toán và ví dụ minh họa

OSPF là một giao thức dựa theo trạng thái liên kết. Giống như các giao thức trạng thái liên kết khác, mỗi bộ định tuyến OSPF đều thực hiện thuật toán SPF – Shortest Path First để xử lý các thông tin chứa trong cơ sở dữ liệu trạng thái liên kết. Thuật toán tạo ra một cây đường đi ngắn nhất mô tả cụ thể các tuyến đường nên chọn dẫn tới mạng đích. Cơ sở dữ liệu về cấu

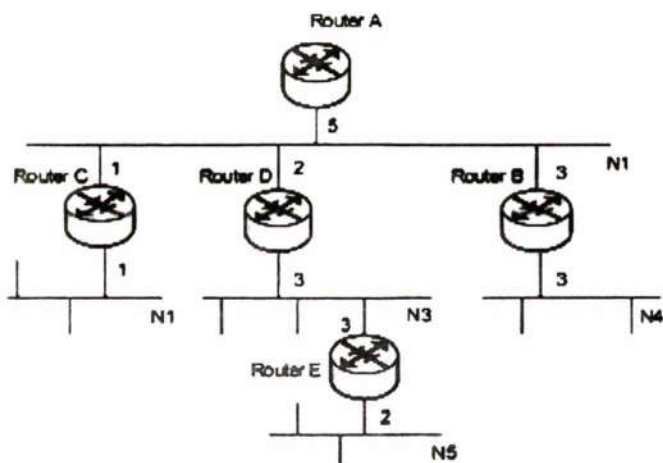


hình mạng về cơ bản bao gồm tất cả các dữ liệu về mạng có liên kết tới bộ định tuyến chứa cơ sở dữ liệu.

Mỗi router sử dụng cây đường đi ngắn nhất để xây dựng bảng định tuyến của mình. Bảng định tuyến chỉ ra giá để mỗi mạng trong khu vực. Để tìm giá tới mạng bên ngoài khu vực, các router sử dụng các quảng cáo liên kết tóm tắt tới mạng, liên kết tóm tắt tới router biên AS và liên kết ngoài. Để thực hiện định tuyến hiệu quả, OSPF chia hệ thống tự trị ra thành nhiều khu vực nhỏ. Mỗi AS có thể được chia ra thành nhiều khu vực khác nhau. Khu vực là tập hợp các mạng, trạm và router nằm trong cùng một hệ thống tự trị.

Tất cả các mạng trong một khu vực phải được kết nối với nhau. Tại biên của khu vực, các router biên khu vực tóm tắt thông tin về khu vực của mình và gửi các thông tin này tới các khu vực khác.

Trong số các khu vực bên trong AS, có một khu vực đặc biệt được gọi là đường trục; tất cả các khu vực trong một AS phải được nối tới đường trục. Hay nói cách khác là đường trục được coi như là khu vực sơ cấp còn các khu vực còn lại đều được coi như là các khu vực thứ cấp.



Hình 7.19. Định tuyến sử dụng cây đường đi ngắn nhất

Các router bên trong khu vực đường trục được gọi là các router đường trục, các router đường trục cũng có thể là một router biên khu vực. Nếu vì

một lý do nào đó mà kết nối giữa một khu vực và đường trục bị hỏng thì người quản trị mạng phải tạo một liên kết ảo (virtual link) giữa các router để cho phép đường trục tiếp tục hoạt động như một hoạt động sơ cấp như hình 7.19.

**Bảng 7.17.** Bảng sử dụng các router trong mạng

Mạng	Giá	Router tiếp theo	Thông tin khác
N1	5		
N2	7	C	
N3	10	D	
N4	11	B	
N5	15	D	

Giải thuật chọn đường dẫn ngắn nhất SPF là cơ sở cho hệ thống OSPF. Khi 1 bộ định tuyến sử dụng SPF được khởi động, bộ định tuyến sẽ khởi tạo cấu trúc cơ sở dữ liệu của giao thức định tuyến và sau đó đợi chỉ báo từ các giao thức tầng thấp hơn dưới dạng các hàm.

Bộ định tuyến sẽ sử dụng các gói tin OSPF Hello để thu nhận các bộ định tuyến lân cận của mình. Bộ định tuyến gửi gói tin Hello đến các lân cận và nhận các bản tin Hello từ các bộ định tuyến lân cận. Ngoài việc sử dụng gói tin Hello để thu nhận các lân cận, bản tin Hello còn được sử dụng để xác nhận việc mình vẫn đang hoạt động đến các bộ định tuyến khác.

Mỗi bộ định tuyến định kỳ gửi các gói thông báo về trạng thái liên kết (LSA) để cung cấp thông tin cho các bộ định tuyến lân cận hoặc cho các bộ định tuyến khác khi một bộ định tuyến thay đổi trạng thái. Bằng việc so sánh trạng thái liên kết của các bộ định tuyến liền kề để tồn tại trong cơ sở dữ liệu, các bộ định tuyến bị lỗi sẽ bị phát hiện ra nhanh chóng và cấu hình mạng sẽ được biến đổi thích hợp. Từ cấu trúc dữ liệu được sinh ra do việc cập nhật liên tục các gói LSA, mỗi bộ định tuyến sẽ tính toán cây đường đi ngắn nhất của mình và tự mình sẽ làm gốc của cây. Sau đó từ cây đường đi ngắn nhất sẽ sinh ra bảng định tuyến.

Một khi có sự thay đổi topo mạng hoặc lưu lượng các node mạng phải khởi tạo và tính toán lại tuyến đường đi ngắn nhất, tùy theo giao thức được sử dụng trong mạng.

Ưu điểm và nhược điểm:

- Cân bằng tải giữa các tuyến cùng giá: Việc sử dụng cùng lúc nhiều tuyến cho phép tận dụng có hiệu quả tài nguyên mạng.

- Phân chia mạng một cách logic: Điều này làm giảm bớt các thông tin phát ra trong những điều kiện bất lợi. Nó cũng giúp kết hợp các thông báo về định tuyến, hạn chế việc phát đi những thông tin không cần thiết về mạng.

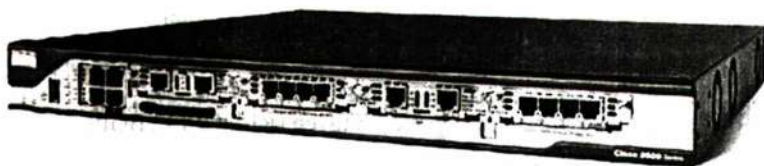
- Hỗ trợ nhận thực: OSPF hỗ trợ nhận thực cho tất cả các node phát thông tin quảng cáo định tuyến. Điều này hạn chế được nguy cơ thay đổi bảng định tuyến với mục đích xấu.

- Thời gian hội tụ nhanh hơn: OSPF cho phép truyền các thông tin về thay đổi tuyến một cách tức thì. Điều đó giúp rút ngắn thời gian hội tụ cần thiết để cập nhật thông tin cấu hình mạng.

- Hỗ trợ CIDR và VLSM: Điều này cho phép nhà quản trị mạng có thể phân phối nguồn địa chỉ IP một cách có hiệu quả hơn.

## 7.6. ROUTER

### 7.6.1. Giới thiệu chung về router



*Hình 7.20. Hình ảnh thực tế của router (router hãng Cisco)*

#### 7.6.1.1. Nhiệm vụ

Router là thiết bị mạng hoạt động ở tầng thứ 3 của mô hình OSI – tầng network. Router được chế tạo với hai mục đích chính:

– Phân cách các mạng máy tính thành các múi (*segment*) riêng biệt để giảm hiện tượng ùn đống hay thực hiện chức năng bảo mật.

– Kết nối các mạng máy tính hay kết nối các user với mạng máy tính ở các khoảng cách xa nhau thông qua các đường truyền thông.

– Cùng với sự phát triển của switch, chức năng đầu tiên của router ngày nay đã được switch đảm nhận một cách hiệu quả. Router chỉ còn phải đảm nhận việc thực hiện các kết nối truy cập từ xa (*remote access*) hay các kết nối WAN cho hệ thống mạng LAN.

– Do hoạt động ở tầng thứ 3 của mô hình OSI, router sẽ hiểu được các giao thức (*protocol*) và quyết định phương thức truyền dữ liệu. Các địa chỉ mà router hiểu là các địa chỉ "giả" được quy định bởi các giao thức (*protocol*). Ví dụ như địa chỉ IP đối với protocol TCP/IP, địa chỉ IPX đối với protocol IPX...

– Do đó tùy theo cấu hình, router quyết định phương thức và đích đến của việc chuyển các gói (*packet*) từ nơi này sang nơi khác. Một cách tổng quát router sẽ chuyển packet theo các bước sau:

- + Đọc packet.
- + Gỡ bỏ dạng format quy định bởi protocol của nơi gửi.
- + Thay thế phần gỡ bỏ đó bằng dạng format của protocol của đích đến.
- + Cập nhật thông tin về việc chuyển dữ liệu: địa chỉ, trạng thái của nơi gửi, nơi nhận.
- + Gửi packet đến nơi nhận qua đường truyền tối ưu nhất.

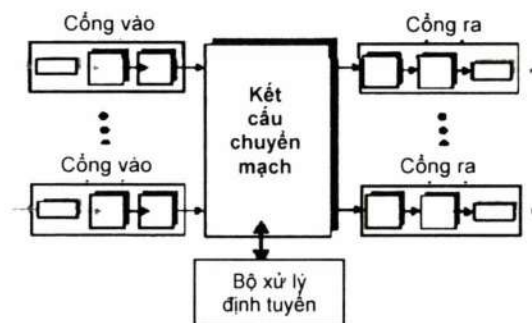
### **7.6.1.2. Chức năng**

- Chạy các thuật toán, giao thức định tuyến (RIP, OSPF, BGP).
- Chuyển các gói dữ liệu dựa vào địa chỉ luận lý (IP).
- Các router hoạt động như một PC: Có hệ điều hành và các thành phần cơ bản tương đương PC.

– Router thực hiện các chức năng cơ bản của nó một phần nhờ vào bảng định tuyến (Routing table): Chứa các thông tin về đường đi đến các mạng mà nó biết.



– Routing table được xây dựng dựa vào các giao thức định tuyến mà router đó được cài đặt.



Hình 7.21. Mô tả quá trình chuyển các gói dữ liệu của router

### 7.6.1.3. Phân loại

Router có nhiều cách phân loại khác nhau. Tuy nhiên, người ta thường có hai cách phân loại chủ yếu sau:

- Dựa theo công dụng của router: theo cách phân loại này người ta chia router thành remote access router, ISDN router, Serial router, router/hub...
- Dựa theo cấu trúc của router: router cấu hình cố định (*Fixed configuration router*), modular router. Tuy nhiên, không có sự phân loại rõ ràng router: mỗi một hãng sản xuất có thể có các tên gọi khác nhau, cách phân loại khác nhau.

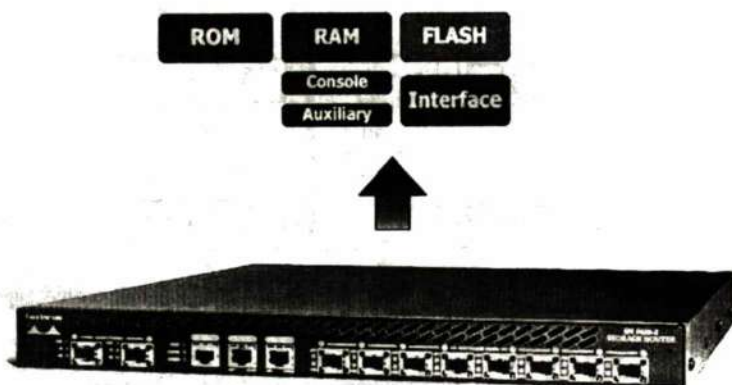
### 7.6.1.4. Cấu tạo của router và các kết nối

#### a. Các thành phần bên trong của router

Cấu trúc chính xác của router rất khác nhau tùy theo từng phiên bản router. Trong phần này chỉ giới thiệu về các thành phần cơ bản của router.

- CPU – Đơn vị xử lý trung tâm

Thực thi các câu lệnh của hệ điều hành để thực hiện các nhiệm vụ sau: khởi động hệ thống, định tuyến, điều khiển các cổng giao tiếp mạng. CPU là một bộ giao tiếp mạng. CPU là một bộ vi xử lý. Trong các router lớn có thể có nhiều CPU.

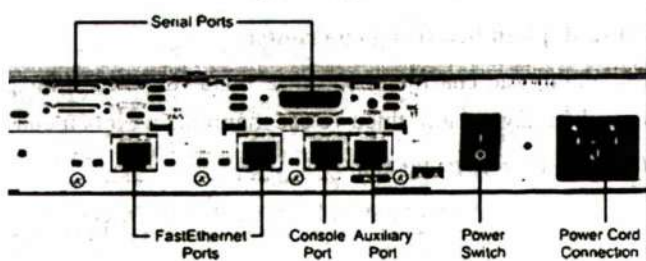


Hình 7.22. Mô tả cấu trúc bên trong của router (Hãng Cisco)

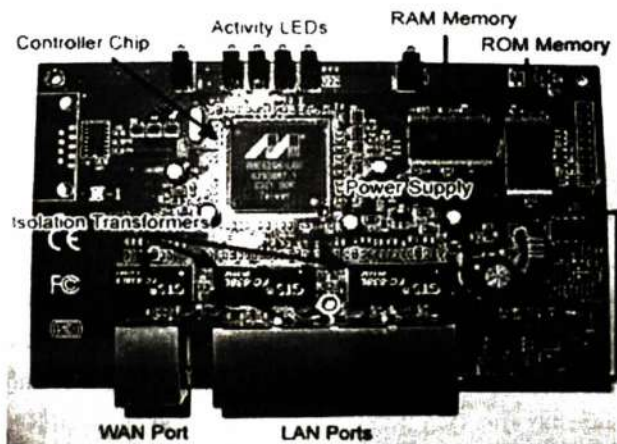


- o RAM - Random Access Memory
- o NVRAM - NonVolatile RAM
- o FLASH
- o ROM - Read Only Memory
- o CONSOLE
- o INTERFACES
- o CPU,
- o AC Power Supply

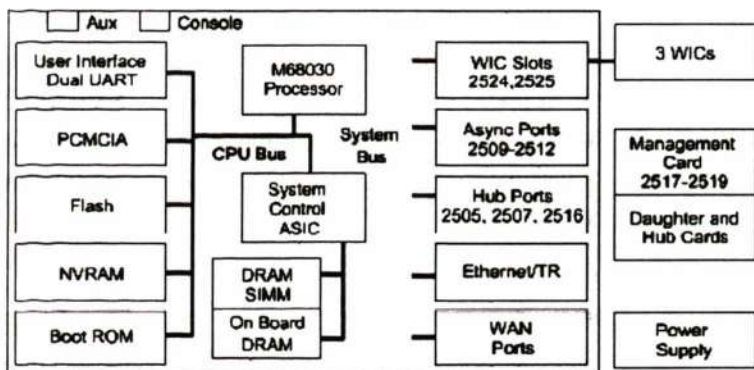
Hình 7.23. Router và các thành phần của router



Hình 7.24. Hình ảnh phía sau của router



Hình 7.25. Cấu tạo bên trong của rotuer (Router hãng Cisco)



Hình 7.26. Sơ đồ khối mô tả cấu tạo bên trong của router

• RAM

Được sử dụng để lưu bảng định tuyến, cung cấp bộ nhớ cho chuyển mạch nhanh, chạy tập tin cấu hình và cung cấp hàng đợi cho các gói dữ liệu. Trong đa số router, hệ điều hành Cisco IOS (*Internetwork Operating System*) chạy trên RAM. RAM thường được chia thành hai phần: phần bộ nhớ xử lý chính và phần bộ nhớ chia sẻ xuất/nhập. Phần bộ nhớ chia sẻ

xuất/nhập được chia cho các cổng giao tiếp làm nơi lưu trữ tạm các gói dữ liệu. Toàn bộ nội dung trên RAM sẽ bị xoá khi tắt điện. Thông thường, RAM trên router là loại RAM động (*DRAM – Dynamic RAM*) và có thể nâng thêm RAM bằng cách gắn thêm DIMM (*Dual In-Line Memory Module*).

- *Flash*

Bộ nhớ Flash được sử dụng để lưu toàn bộ phần mềm hệ điều hành Cisco IOS. Mặc định là router tìm IOS của nó trong flash. Bạn có thể nâng cấp hệ điều hành bằng cách chép phiên bản mới hơn vào flash. Phần mềm IOS có thể ở dưới dạng nén hoặc không nén. Đối với hầu hết các router, IOS được chép lên RAM trong quá trình khởi động router. Còn có một số router thì IOS có thể chạy trực tiếp trên flash mà không cần chép lên RAM. Bạn có thể gắn thêm hoặc thay thế các thanh SIMM hay card PCMCIA để nâng dung lượng flash.

- *NVRAM (Non-volatile Random-access Memory)*

Là bộ nhớ RAM không bị mất thông tin, được sử dụng để lưu tập tin cấu hình. Trong một số thiết bị có NVRAM và flash riêng, NVRAM được thực thi nhờ flash. Trong một số thiết bị, flash và NVRAM là cùng một bộ nhớ. Trong cả hai trường hợp, nội dung của NVRAM vẫn được lưu giữ khi tắt điện.

- *ROM (Read Only Memory)*

Là nơi lưu đoạn mã của chương trình kiểm tra khi khởi động. Nhiệm vụ chính của ROM là kiểm tra phần cứng của router khi khởi động, sau đó chép phần mềm Cisco IOS từ flash vào RAM. Một số router có thể có phiên bản IOS cũ dùng làm nguồn khởi động dự phòng. Nội dung trong ROM không thể xoá được. Ta chỉ có thể nâng cấp ROM bằng cách thay chip ROM mới.

- *Các cổng giao tiếp*

Là nơi router kết nối với bên ngoài. Router có 3 loại cổng: LAN, WAN và console/AUX. Cổng giao tiếp LAN có thể gắn cố định trên router hoặc dưới dạng card rời. Cổng giao tiếp WAN có thể là cổng Serial, ISDN, cổng tích hợp đơn vị dịch vụ kênh CSU (*Chanel Service Unit*).



Tương tự như cổng giao tiếp LAN, các cổng giao tiếp WAN cũng có chip điều khiển đặc biệt. Cổng giao tiếp WAN có thể định trên router hoặc ở dạng card rời. Cổng console/AUX là cổng nối tiếp, chủ yếu được sử dụng để cấu hình router. Hai cổng này không phải là loại cổng để kết nối mạng mà kết nối vào máy tính thông qua Modem hoặc thông qua cổng COM trên máy tính để từ máy tính thực hiện cấu hình router.

- *Nguồn điện*

Cung cấp điện cho các thành phần của router, một số router lớn có thể sử dụng nhiều bộ nguồn hoặc nhiều card nguồn. Còn ở một số router nhỏ, nguồn điện có thể là bộ phận nằm ngoài router.

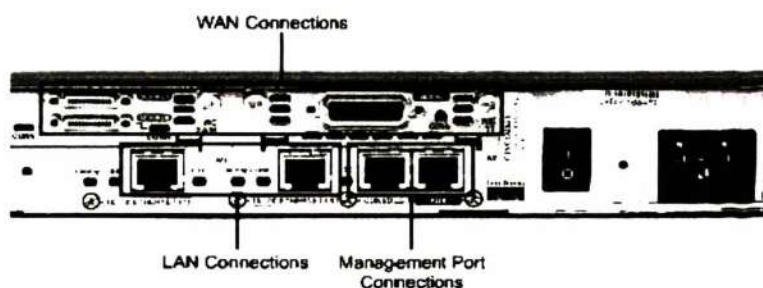
### **b. Các loại kết nối của router**

Router có ba loại kết nối cơ bản là: cổng LAN, WAN và cổng quản lý router.

- Cổng giao tiếp LAN cho phép router kết nối vào môi trường mạng cục bộ LAN. Thông thường, cổng giao tiếp LAN là cổng Ethernet. Ngoài ra cũng có cổng Token Ring và ATM (Asynchronous Transfer Mode).

- Kết nối mạng WAN cung cấp kết nối thông qua các nhà cung cấp dịch vụ đến các chi nhánh ở xa hoặc kết nối vào Internet. Loại kết nối này có thể là nối tiếp hay bất kỳ loại giao tiếp WAN, bạn cần phải có thêm một thiết bị ngoại vi như CSU chẳng hạn để nối router đến nhà cung cấp dịch vụ. Đối với một số loại giao tiếp WAN khác thì bạn có thể kết nối trực tiếp router của mình đến nhà cung cấp dịch vụ.

- Chức năng của port quản lý hoàn toàn khác với hai loại trên kết nối LAN, WAN để kết nối router và mạng để router nhận và phát các gói dữ liệu. Trong khi đó, port quản lý cung cấp cho bạn một kết nối dạng văn bản để bạn có thể cấu hình hoặc xử lý trên router. Cổng quản lý thường là cổng console hoặc cổng AUX (Auxilliary). Đây là loại cổng nối tiếp bất đồng bộ EIA-232. Các cổng này kết nối vào cổng COM trên máy tính. Trên máy tính, chúng ta sử dụng chương trình mô phỏng thiết bị đầu - cuối để thiết lập phiên kết nối dạng văn bản vào router. Thông qua kiểu kết nối này, người quản trị mạng có thể quản lý thiết bị của mình.



Hình 7.27. Mặt sau của router

## 7.6.2. Nguyên tắc hoạt động của router

Như ta đã biết tại tầng network của mô hình OSI, thường sử dụng các loại địa chỉ mang tính chất quy ước như IP, IPX... Các địa chỉ này là các địa chỉ có hướng, nghĩa là chúng được phân thành hai phần riêng biệt là phần địa chỉ network và phần địa chỉ host. Cách đánh số địa chỉ như vậy nhằm giúp cho việc tìm ra các đường kết nối từ hệ thống mạng này sang hệ thống mạng khác được dễ dàng hơn. Các địa chỉ này có thể được thay đổi theo tùy ý người sử dụng. Trên thực tế, các card mạng chỉ có thể kết nối với nhau theo địa chỉ MAC, địa chỉ cố định và duy nhất của phần cứng. Do vậy ta phải có một phương pháp để chuyển đổi các dạng địa chỉ này qua lại với nhau. Từ đó có giao thức phân giải địa chỉ: Address Resolution Protocol (ARP).

ARP là một protocol dựa trên nguyên tắc: khi một thiết bị mạng muốn biết địa chỉ MAC của một thiết bị mạng nào đó mà nó đã biết địa chỉ ở tầng network (IP, IPX...) nó sẽ gửi một ARP request bao gồm địa chỉ MAC address của nó và địa chỉ IP của thiết bị mà nó cần biết MAC address trên toàn bộ một miền broadcast. Mỗi một thiết bị nhận được request này sẽ so sánh địa chỉ IP trong request với địa chỉ tầng network của mình. Nếu trùng địa chỉ thì thiết bị đó phải gửi ngược lại cho thiết bị gửi ARP request một packet (trong đó có chứa địa chỉ MAC của mình).

Trong một hệ thống mạng đơn giản, ví dụ như máy A muốn gửi packet đến máy B và nó chỉ biết được địa chỉ IP của máy B. Khi đó máy A sẽ phải

gửi một ARP broadcast cho toàn mạng để hỏi xem "địa chỉ MAC của máy có địa chỉ IP này là gì?" Khi máy B nhận được broadcast này, nó sẽ so sánh địa chỉ IP trong packet này với địa chỉ IP của nó. Nhận thấy địa chỉ đó là địa chỉ của mình, máy B sẽ gửi lại một packet cho máy A trong đó có chứa địa chỉ MAC của B. Sau đó máy A mới bắt đầu truyền packet cho B.

Trong một môi trường phức tạp hơn: hai hệ thống mạng gắn với nhau thông qua một router C. Máy A thuộc mạng A muốn gửi packet đến máy B thuộc mạng B. Do các broadcast không thể truyền qua router nên khi đó máy A sẽ xem router C như một cầu nối để truyền dữ liệu. Trước đó, máy A sẽ biết được địa chỉ IP của router C (port X) và biết được rằng để truyền packet tới B phải đi qua C. Tất cả các thông tin như vậy sẽ được chứa trong một bảng gọi là bảng routing (routing table). Bảng routing table theo cơ chế này được lưu giữ trong mỗi máy. Routing table chứa thông tin về các gateway để truy cập vào một hệ thống mạng nào đó. Ví dụ, trong trường hợp trên trong bảng sẽ chỉ ra rằng để đi tới LAN B phải qua port X của router C. Routing table có chứa địa chỉ IP của port X. Quá trình truyền dữ liệu theo từng bước sau:

- Máy A gửi một ARP request (broadcast) để tìm địa chỉ MAC của port X.
- Router C trả lời, cung cấp cho máy A địa chỉ MAC của port X.
- Máy A truyền packet đến port X của router.
- Router nhận được packet từ máy A, chuyển packet ra port Y của router. Trong packet có chứa địa chỉ IP của máy B.
- Router sẽ gửi ARP request để tìm địa chỉ MAC của máy B.
- Máy B sẽ trả lời cho router biết địa chỉ MAC của mình.
- Sau khi nhận được địa chỉ MAC của máy B, router C gửi packet của A đến B.

## CÂU HỎI VÀ BÀI TẬP

1. Cho một mạng vật lý có địa chỉ mạng là 121.0.0.0.
  - a) Chia mạng thành 8 mạng con. Xác định submask và địa chỉ mạng của các mạng con, số lượng máy tối đa trong một mạng con.
  - b) Cho một máy có địa chỉ IP: 121.181.185.135. Cho biết máy thuộc mạng con nào? Địa chỉ mạng con và địa chỉ máy trong mạng con? Địa chỉ broadcast của mạng con.
2. Cho 4 phòng máy A, B, C, D
  - Phòng A, B mỗi phòng gồm 5 máy;
  - Phòng C gồm 6 máy;
  - Phòng D gồm 1 server.
  - a) Vẽ sơ đồ cấu hình mạng (giả sử dùng HUB 8 cổng, phòng A, B cách phòng D 50m, phòng C cách phòng D 500m).
  - b) Giả sử mạng được cung cấp dải địa chỉ lớp C là 192.168.5.0. Yêu cầu gán địa chỉ cho các máy trong các phòng.
3. Giả thiết khi kết nối mạng, trường ĐH Sư Phạm Kỹ Thuật Hưng Yên được nhà cung cấp phân cho một dải địa chỉ mạng lớp C là 192.10.10.0/24. Bạn hãy phân phối một cách hợp lý địa chỉ mạng con (subnet) và mặt nạ mạng con (subnet mask) cho các khoa và phòng ban trong trường hợp cụ thể như sau:
  - Khoa CNTT có 50 máy;
  - Khoa Điện 30 máy;
  - Khoa Máy có 28 máy;
  - Phòng ban có 12 máy;
  - Khoa Sư phạm có 5 máy.Cho biết dải địa chỉ có thể sử dụng được của từng khoa/phòng ban.
4. Với địa chỉ IP 178.125.0.0 kết nối vào mạng Internet được cấp cho mạng của mình, một công ty dự định thiết kế 4 mạng con có tối đa 60 máy tính/mạng. Giả thiết các mạng con có địa chỉ 178.125.27.0, 178.125.31.64, 178.125.35.128, 178.125.41.192.



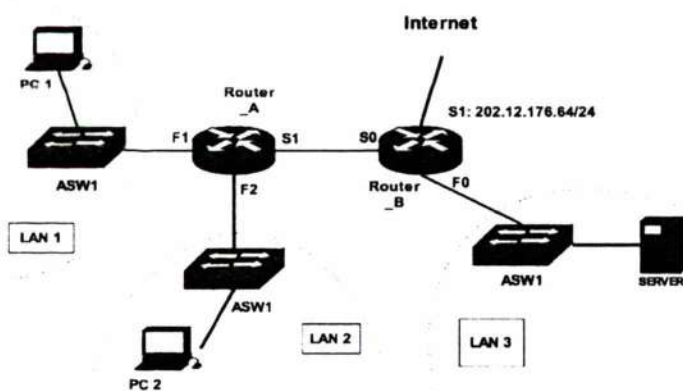
Xác định mặt nạ, dải địa chỉ IP và địa chỉ broadcast của mỗi mạng con. Hãy phân tích và đưa ra cấu hình thiết kế nối mạng Internet cho công ty với các mạng con đã cho, biết rằng công ty sử dụng hai router với 2 giao diện mạng (2 interface) và một router với 3 giao diện mạng. Nếu mỗi mạng con có 2 máy tính kết nối vào, hãy định địa chỉ IP phù hợp cho mỗi máy tính đó.

5. Giả thiết nhà cung cấp dịch vụ (ISP) cho phép bạn sử dụng dải địa chỉ IP với mặt nạ mạng là:

IP address: 199.141.27.0

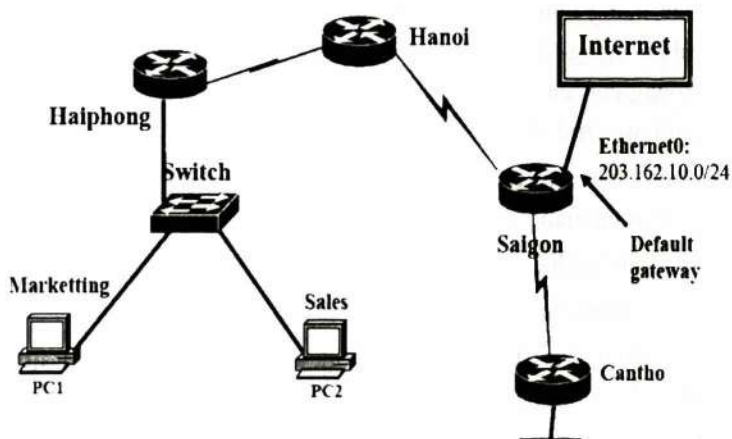
Subnet mask: 255.255.255.240

- Xác định địa chỉ mạng (network address) và địa chỉ quảng bá (broadcast address).
  - Xác định số lượng mạng mà ISP có thể cung cấp.
  - Xác định dải địa chỉ mạng trên.
6. Giả thiết một công ty có sơ đồ logic mạng như hình sau. Quản trị mạng sử dụng dải địa chỉ IP là 200.168.105.0/24 để gán cho các máy của công ty. Biết rằng trong mạng LAN 1 và LAN 2, mỗi mạng quản lý 50 máy tính, mạng LAN 3 quản lý SERVER.



Yêu cầu: Đánh địa chỉ IP và xác định mặt nạ mạng cho các PC1, PC2, SERVER, các interface S1 của router\_A và S0 của router\_B.

7. Giả thiết mạng công ty như hình sau. Biết rằng công ty được cấp 1 dải địa chỉ IP lớp C là 215.112.13.0/24



Yêu cầu: Gán và xác định địa chỉ IP của các máy tính và các interface của các router.

## **Chương 8**

# **MẠNG CỤC BỘ – MẠNG LAN**

### **MỤC TIÊU**

*Sau khi học xong chương này người học phải có khả năng:*

- ✓ Trình bày được lớp liên kết dữ liệu và các mô hình dịch vụ, giao thức đa truy nhập, công nghệ mạng cục bộ – LAN, các thiết bị mạng LAN, công nghệ mạng diện rộng WAN.
- ✓ Phân biệt được các thiết bị của mạng LAN.
- ✓ So sánh được công nghệ mạng cục bộ LAN và công nghệ mạng diện rộng WAN.
- ✓ Phát triển khả năng tư duy, sáng tạo, tự giác học tập...

### **NỘI DUNG**

- 8.1. Đặc trưng của mạng cục bộ.
  - 8.2. Kiến trúc mạng cục bộ
  - 8.3. Các phương pháp truy cập đường truyền
  - 8.4. Công nghệ mạng cục bộ – LAN.
  - 8.5. Các thiết bị mạng LAN.
  - 8.6. Công nghệ mạng diện rộng WAN.
- Câu hỏi và bài tập.

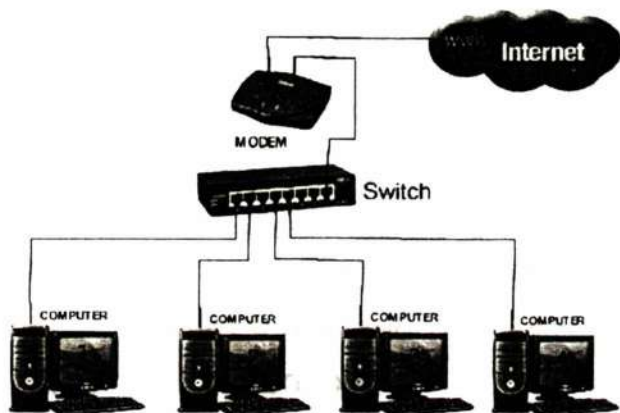
## **8.1. ĐẶC TRƯNG CỦA MẠNG CỤC BỘ**

LAN là viết tắt của *Local Area Network* (Mạng cục bộ).

Do nhu cầu thực tế của các cơ quan, trường học, doanh nghiệp, tổ chức cần kết nối các máy tính đơn lẻ thành một mạng nội bộ để tạo khả năng trao đổi thông tin, sử dụng chung tài nguyên (phần cứng, phần mềm). Ví dụ

trong một văn phòng có một máy in, để tất cả mọi người có thể sử dụng chung máy in đó thì giải pháp nối mạng có thể khắc phục được hạn chế này.

Mục đích của việc sử dụng mạng ngày nay có nhiều thay đổi so với trước kia. Mặc dù mạng máy tính phát sinh từ nhu cầu chia sẻ và dùng chung tài nguyên, nhưng chủ yếu vẫn là sử dụng chung tài nguyên phần cứng. Ngày nay mục đích chính của mạng là trao đổi thông tin và CSDL dùng chung trong công nghệ mạng cục bộ phát triển vô cùng nhanh chóng.



*Hình 8.1. Mạng LAN kết nối nhiều thiết bị*

Để phân biệt mạng LAN với các loại mạng khác người ta căn cứ theo các đặc trưng sau:

- *Đặc trưng địa lý*: cài đặt trong phạm vi nhỏ (toà nhà, một căn cứ quân sự...) có đường kính từ vài chục mét đến vài chục km trong có ý nghĩa tương đối.
- *Đặc trưng về tốc độ truyền*: cao hơn mạng diện rộng, khoảng 100 Mb/s, có thể đến 1000 Mbps với công nghệ Gigabit.
- *Đặc trưng độ tin cậy*: tỷ suất lỗi thấp, có thể đạt  $10^{-8}$  đến  $10^{-11}$ .
- *Đặc trưng quản lý*: thường là sở hữu riêng của một tổ chức trong việc quản lý khai thác tập trung, thống nhất.

Tuy nhiên, sự phân biệt mạng LAN theo các đặc trưng trên chỉ mang tính tương đối, cùng với công nghệ ngày càng cao thì ranh giới giữa LAN MAN, WAN ngày càng mờ đi.



## 8.2. KIẾN TRÚC MẠNG CỤC BỘ

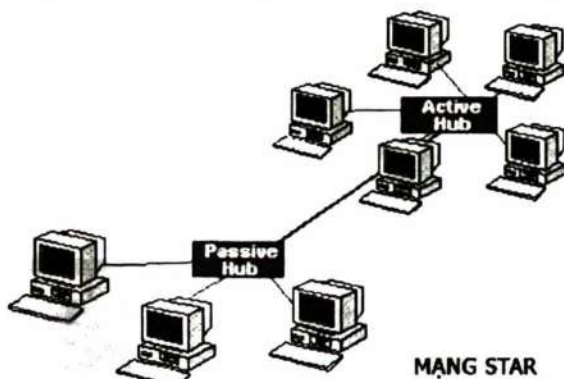
### 8.2.1 Cấu trúc liên kết trong mạng LAN (Topology)

Về nguyên tắc mọi topology của mạng máy tính nói chung đều có thể dùng cho mạng cục bộ. Song do đặc thù của mạng cục bộ nên chỉ có 3 topology thường được sử dụng: hình sao (star), hình vòng (ring), tuyến tính (bus).

#### 8.2.1.1. Hình sao (star)

Tất cả các trạm được nối vào một thiết bị trung tâm có nhiệm vụ nhận tín hiệu từ các trạm và chuyển đến trạm đích của tín hiệu. Thiết bị trung tâm có thể là Hub, Switch, router.

Vai trò của thiết bị trung tâm là thực hiện việc "bắt tay" giữa các trạm cần trao đổi thông tin với nhau, thiết lập các liên kết điểm – điểm giữa chúng.



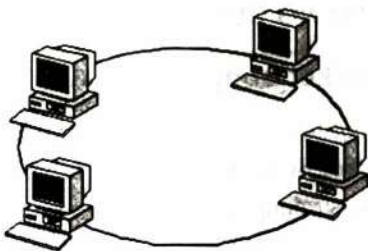
Hình 8.2. Sơ đồ kết nối hình sao với HUB ở trung tâm

#### 8.2.1.2. Hình vòng (ring)

Tín hiệu được lưu chuyển theo một chiều duy nhất.

Mỗi trạm làm việc được nối với vòng qua một bộ chuyển tiếp (repeater), có nhiệm vụ nhận tín hiệu rồi chuyển đến trạm kế tiếp trên vòng.

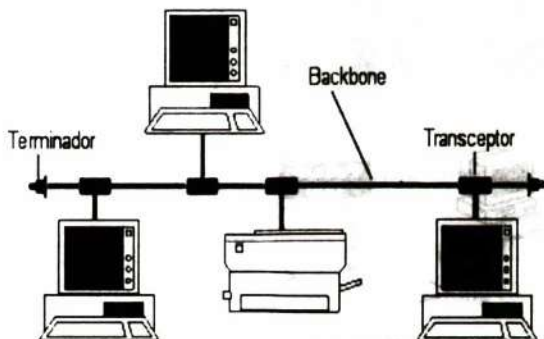
Để tăng độ tin cậy của mạng, phải lắp vòng dự phòng, khi đường truyền trên vòng chính bị sự cố thì vòng phụ được sử dụng với chiều đi của tín hiệu ngược với chiều đi của mạng chính.



Dạng vòng

*Hình 8.3. Sơ đồ kết nối hình vòng (ring)***8.2.1.3. Dạng đường thẳng (Bus)**

- Tất cả các trạm đều dùng chung một đường truyền chính (Bus) được giới hạn bởi hai đầu nối (terminator).
- Mỗi trạm được nối vào Bus qua một đầu nối chữ T (T – connector).
- Khi một trạm truyền dữ liệu thì tín hiệu được quảng bá trên 2 chiều của Bus (tất cả các trạm khác đều có thể nhận tín hiệu).

*Hình 8.4. Sơ đồ kết nối đường thẳng (bus)***8.2.1.4. So sánh giữa các cách kết nối và ưu nhược điểm của chúng**

Kiểu hình sao là kết nối điểm – điểm trực tiếp giữa hai máy tính thông qua một thiết bị trung tâm. Kiểu vòng thì tín hiệu lưu chuyển trên vòng là một chuỗi các kết nối điểm – điểm. Kiểu tuyến tính thì dữ liệu truyền dựa trên điểm – nhiều điểm hoặc quảng bá.

Ưu điểm: Cả ba cách kết nối đều đơn giản, dễ lắp đặt, dễ thay đổi cấu hình.

- *Hình sao:*

- Ưu điểm: Dễ kiểm soát. Do sử dụng liên kết điểm – điểm nên tận dụng được tối đa tốc độ của đường truyền vật lý.

- Nhược điểm: Độ dài đường truyền nối một trạm với thiết bị trung tâm bị hạn chế (trong vòng 100m với công nghệ hiện tại).

- *Dạng vòng:*

Nhược điểm: Nếu xảy ra sự cố trên đường truyền, tất cả các máy trong mạng không thể giao tiếp với nhau. Đòi hỏi giao thức truy nhập đường truyền khá phức tạp. Tuy nhiên toàn bộ công việc này được hệ phần mềm giải quyết.

- *Dạng đường thẳng:*

Nhược điểm: Nếu xảy ra sự cố trên đường truyền, toàn bộ các máy trong mạng không thể giao tiếp với nhau được nữa. Giao thức quản lý truy nhập đường truyền phức tạp.

Do ưu, nhược điểm của từng loại mà trong thực tế người ta thường chọn kiểu kết nối lại là tổ hợp của các kiểu kết nối trên.

### 8.2.2. Đường truyền vật lý

Mạng cục bộ thường sử dụng 3 loại đường truyền vật lý và cáp đôi xoắn, cáp đồng trục, và cáp sợi quang. Ngoài ra gần đây người ta cũng đã bắt đầu sử dụng nhiều các mạng cục bộ không dây nhờ radio hoặc viba.

Cáp đồng trục đường sử dụng nhiều trong các mạng dạng tuyến tính, hoạt động truyền dẫn theo dải cơ sở (baseband) hoặc dải rộng (broadband). Với dải cơ sở, toàn bộ khả năng của đường truyền được dành cho một kênh truyền thông duy nhất, trong khi đó với dải rộng thì hai hoặc nhiều kênh truyền thông cùng phân chia dải thông của kênh truyền. Hầu hết các mạng cục bộ đều sử dụng phương thức dải rộng. Với phương thức này tín hiệu có thể truyền đi dưới cả hai dạng: tương tự (analog) và số (digital) không cần điều chế.

Cáp đồng trục có hai loại là cáp gầy (thin cable) và cáp béo (thick cable). Cả hai loại cáp này đều có tốc độ làm việc 10Mb/s nhưng cáp gầy có

độ suy hao tín hiệu lớn hơn, có độ dài cáp tối đa cho phép giữa hai repeater nhỏ hơn cáp béo. Cáp gầy thường dùng để nối các trạm trong cùng một văn phòng, phòng thí nghiệm, còn cáp béo dùng để nối dọc theo hành lang, lên các tầng lầu...

Phương thức truyền thông theo dải rộng có thể dùng cả cáp đôi xoắn, nhưng cáp đôi xoắn chỉ thích hợp với mạng nhỏ hiệu năng thấp và chi phí đầu tư ít. Phương thức truyền theo dải rộng chia dải thông (tần số) của đường truyền thành nhiều dải tần con (kênh), mỗi dải tần con đó cung cấp một kênh truyền dữ liệu tách biệt nhờ sử dụng một cặp Modem đặc biệt. Phương thức này vốn là một phương tiện truyền một chiều: các tín hiệu đưa vào đường truyền chỉ có thể truyền đi theo một hướng không cài đặt được các bộ khuếch đại để chuyển tín hiệu của một tần số theo cả hai chiều. Vì thế xảy ra tình trạng chỉ có trạm nằm dưới trạm truyền là có thể nhận được tín hiệu. Vậy làm thế nào để có hai đường dẫn dữ liệu trên mạng. Điểm gặp nhau của hai đường dẫn đó gọi là điểm đầu - cuối.

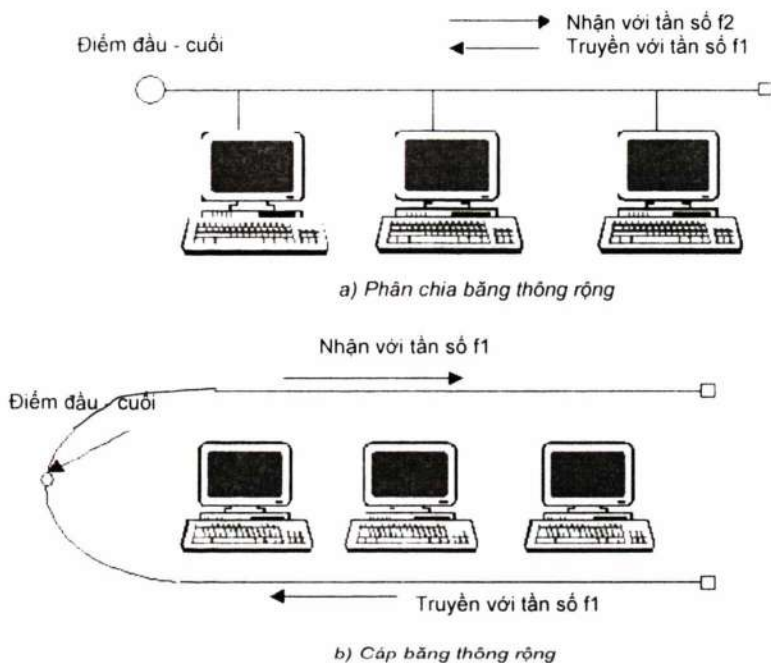
*Ví dụ:* Trong topo dạng bus thì điểm đầu - cuối đơn giản chính là đầu mút của bus (*terminator*), còn với topo dạng cây (*tree*) thì chính là gốc của cây (*root*). Các trạm khi truyền đều truyền về hướng điểm đầu - cuối (gọi là đường dẫn về), sau đó các tín hiệu nhận được ở điểm đầu - cuối sẽ truyền theo đường dẫn thứ hai xuất phát từ điểm đầu - cuối (gọi là đường dẫn đi). Tất cả các trạm đều nhận dữ liệu trên đường dẫn đi.

Trong cấu hình cáp đôi (*dual cable*), các đường dẫn về và đi chạy trên các cáp riêng biệt và điểm đầu - cuối đơn giản chỉ là một đầu nối thụ động của chúng. Trạm gửi và nhận cùng một tần số. Trong cấu hình tách (*split*), cả hai đường dẫn đều ở trên cùng một cáp nhưng tần số khác nhau: đường dẫn về có tần số thấp và đường dẫn đi có tần số cao hơn. Điểm đầu - cuối là bộ chuyển đổi tần số.

*Chú ý:* Việc lựa chọn đường truyền và thiết kế sơ đồ đi cáp (trọng trường hợp hữu tuyến) là một trong những công việc quan trọng nhất khi thiết kế và cài đặt một mạng máy tính nói chung và mạng cục bộ nói riêng. Giải pháp lựa chọn đáp ứng được nhu cầu sử dụng mạng thực tế không chỉ cho hiện tại mà cho cả tương lai.



*Ví dụ:* Muốn truyền dữ liệu đa phương tiện thì không thể chọn loại cáp cho phép thông lượng tối đa là vài Mb/s, mà phải nghĩ đến loại cáp cho phép thông lượng trên 100 Mb/s. Việc lắp đặt hệ thống cáp trong nhiều trường hợp (toà nhà nhiều tầng) là tốn rất nhiều công nên phải lựa chọn cẩn thận, không thể để xảy ra trường hợp sau 1 – 2 năm gỡ bỏ, lắp hệ thống mới.



Hình 8.5. Cấu hình vật lý cho Broadband

### 8.3. CÁC PHƯƠNG PHÁP TRUY CẬP ĐƯỜNG TRUYỀN

Như đã trình bày ở trên, chương này trình bày về mạng LAN – mạng dạng truyền quảng bá và các giao thức truyền quảng bá của nó. Trong bất kỳ mạng dạng quảng bá nào, vấn đề then chốt luôn là cách thức quyết định ai có quyền truy cập kênh truyền tại một thời điểm.

Để làm rõ vấn đề hơn, hãy xem xét ví dụ sau: Có sáu người đang họp thông qua hệ thống điện thoại, mọi người đều được nối kết để có thể nghe

và nói với những người khác. Khi một người ngừng nói mà có hai người hoặc nhiều hơn cùng phát biểu tiếp sẽ tạo ra tình trạng lộn xộn. Trong các cuộc họp dạng gặp mặt trực tiếp, tình trạng lộn xộn này có thể được giải quyết bằng cách đưa tay xin phát biểu. Nhưng trong hệ thống hội thảo thông qua điện thoại này, khi mà đường truyền rảnh, việc quyết định ai sẽ nói tiếp có vẻ khó làm hơn. Đã có nhiều giao thức dùng giải quyết vấn đề trên. Và chúng chính là nội dung trình bày của phần này.

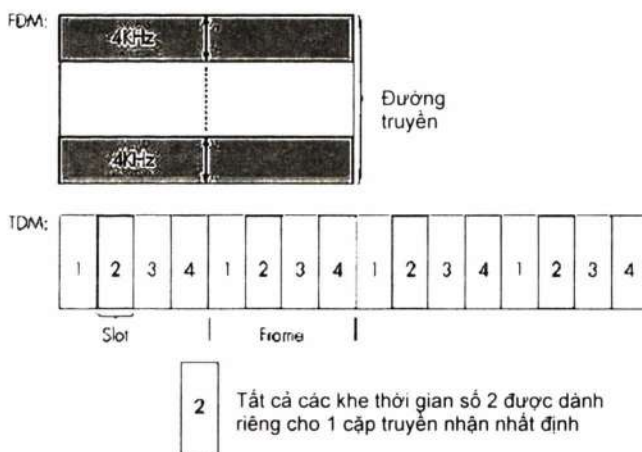
Nói một cách khác, các kênh truyền dạng quảng bá thính thoảng còn được gọi là các kênh đa truy cập (*multiaccess channels*) hay là các kênh truy cập ngẫu nhiên (*random access channels*). Các giao thức được sử dụng để quyết định ai có quyền truy cập đường truyền quảng bá trước được gom vào trong một lớp con của tầng liên kết dữ liệu gọi là lớp con MAC. Lớp con MAC là đặc biệt quan trọng trong mạng LAN, do nhiều mạng LAN sử dụng đường truyền dạng quảng bá như là phương tiện truyền thông nền tảng. Các mạng WAN, theo xu hướng ngược lại, dùng các nối kết dạng điểm – điểm (ngoại trừ các mạng dùng vệ tinh). Về cơ bản, có ba phương pháp điều khiển truy cập đường truyền: Chia kênh, truy cập ngẫu nhiên (*Random Access*) và phân lượt ("*Taking-turns*"). Giải thích cụ thể về ba phương pháp điều khiển truy cập đường truyền trên sẽ được trình bày dưới đây.

### 8.3.1. Phương pháp chia kênh

Ý tưởng chung của phương pháp này là: đường truyền sẽ được chia thành nhiều kênh truyền, mỗi kênh truyền được cấp phát riêng cho một trạm. Có ba phương pháp chia kênh chính: FDMA, TDMA, CDMA.

#### 8.3.1.1. Truy nhập kênh truyền phân chia theo tần số (*FDMA – Frequency Division Multiple Access*)

Một phương thức truyền thống để chia sẻ một kênh truyền đơn cho nhiều người dùng cạnh tranh là chia tần số (FDMA). Phổ của kênh truyền được chia thành nhiều băng tần (frequency bands) khác nhau. Mỗi trạm được gán cho một băng tần cố định. Những trạm nào được cấp băng tần mà không có dữ liệu để truyền thì ở trong trạng thái nhàn rỗi (idle).



Hình 8.6. FDMA và TDMA

*Nhận xét:*

– Do mỗi người dùng được cấp một băng tần riêng, nên không có sự đụng độ xảy ra. Khi chỉ có số lượng người dùng nhỏ và ổn định, mỗi người dùng cần giao tiếp nhiều thì FDMA chính là cơ chế điều khiển truy cập đường truyền hiệu quả.

– Tuy nhiên, khi lượng người gửi dữ liệu lớn và liên tục thay đổi hoặc đường truyền vượt quá khả năng phục vụ thì FDMA bộc lộ một số vấn đề. Nếu phổ đường truyền được chia làm N vùng và có ít hơn N người dùng cần truy cập đường truyền, thì một phần lớn phổ đường truyền bị lãng phí. Ngược lại, có nhiều hơn N người dùng có nhu cầu truyền dữ liệu thì một số người dùng sẽ phải bị từ chối không có truy cập đường truyền vì thiếu băng thông. Tuy nhiên, nếu lại giả sử rằng số lượng người dùng bằng cách nào đó luôn được giữ ổn định ở con số N, thì việc chia kênh truyền thành những kênh truyền con như thế tự thân là không hiệu quả. Lý do cơ bản ở đây là: nếu có vài người dùng rồi, không truyền dữ liệu thì những kênh truyền con cấp cho những người dùng này bị lãng phí.

Có thể dễ dàng thấy được hiệu năng nghèo nàn của FDMA từ một phép tính theo lý thuyết xếp hàng đơn giản. Bắt đầu là thời gian trì hoãn trung bình T trong một kênh truyền có dung lượng C bps, với tỷ lệ đến trung bình

là  $\lambda$  khung/giây, mỗi khung có chiều dài được chỉ ra từ hàm phân phối mũ với giá trị trung bình là  $1/\mu$  bit/khung. Với các tham số trên ta có được tỷ lệ phục vụ là  $\mu C$  khung/giây. Từ lý thuyết xếp hàng ta có:

$$T = \frac{1}{\mu C - \lambda}$$

Ví dụ: Nếu  $C = 100$  Mbps,  $1/\mu = 10000$  bit và  $\lambda = 5000$  khung/giây thì  $T = 200 \mu s$ . Bây giờ nếu ta chia kênh lớn này thành  $N$  kênh truyền nhỏ độc lập, mỗi kênh truyền nhỏ có dung lượng  $C/N$  bps. Tỷ lệ trung bình các khung đến các kênh truyền nhỏ là  $\lambda/N$ . Tính toán lại  $T$  ta có:

$$T_{\text{FDMA}} = \frac{1}{\mu \left( \frac{C}{N} \right) - \left( \frac{\lambda}{N} \right)} = \frac{N}{\mu C - \lambda} = NT$$

Thời gian chờ đợi trung bình trong các kênh truyền con sử dụng FDMA là xấu hơn gấp  $N$  lần so với trường hợp ta sắp xếp cho các khung được truyền tuần tự trong một kênh lớn.

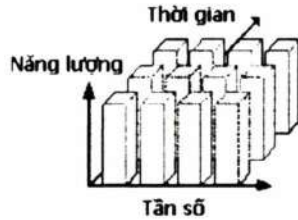
### 8.3.1.2. Truy nhập kênh truyền phân chia theo thời gian (TDM – Time Division Multiple Access)

Trong phương pháp này, các trạm sẽ xoay vòng (round) để truy cập đường truyền. Vòng ở đây có thể hiểu là vòng thời gian. Một vòng thời gian là khoảng thời gian đủ để cho tất cả các trạm trong LAN đều lược quyền truyền dữ liệu. Quy tắc xoay vòng như sau: một vòng thời gian sẽ được chia đều thành các khe (slot) thời gian bằng nhau, mỗi trạm sẽ được cấp một khe thời gian – đủ để nó có thể truyền hết một gói tin. Những trạm nào tới lượt được cấp cho khe thời gian của mình mà không có dữ liệu để truyền thì vẫn chiếm lấy khe thời gian đó, và khoảng thời gian bị chiếm này lược gọi là thời gian nhàn rỗi (idle time). Tập hợp tất cả các khe thời gian trong một vòng được gọi lại là khung (frame). Chúng ta cũng áp dụng cùng một nhận xét về mạng TDMA như mạng FDMA. Mỗi người dùng được cấp phát một khe thời gian. Và nếu người dùng không sử dụng khe thời gian này để truyền dữ liệu thì thời gian sẽ bị lãng phí. *Kết hợp giữa TDMA và FDMA.*

Trong thực tế, hai kỹ thuật TDMA và FDMA thường được kết hợp sử dụng với nhau, ví dụ như trong các mạng điện thoại di động. Các điện thoại



đi động TDMA sử dụng các kênh 30 KHz, mỗi kênh lại được chia thành ba khe thời gian. Một thiết bị cầm tay sử dụng một khe thời gian cho việc gửi và một khe khác cho việc nhận dữ liệu. Chẳng hạn như các hệ thống: Cingular (Nokia 8265, TDMA 800/1900 MHz, AMPS 800 mHz), AT&T Wireless.

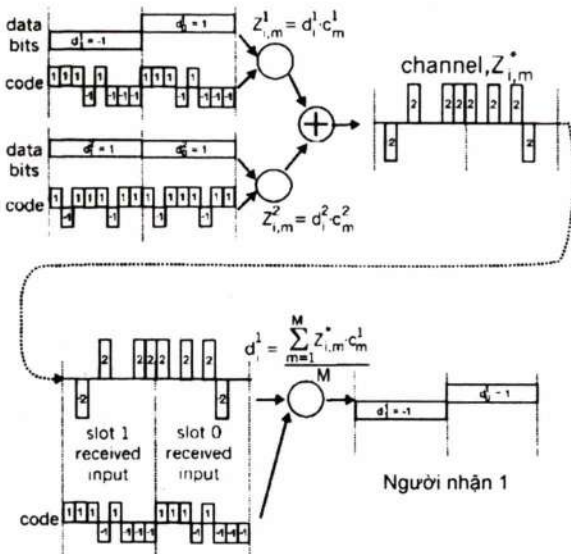


Hình 8.7. Phân chia kênh theo tần số (FDM)

Hệ thống GSM sử dụng các kênh 200 KHz được chia thành 8 khe thời gian. Một thiết bị cầm tay sẽ sử dụng một khe thời gian trong hai kênh khác nhau để gửi và nhận thông tin. Các hệ thống Cingular, T-Mobile, AT&T đang chuyển sang dùng kỹ thuật này.

**8.3.1.3. Truy nhập kênh truyền phân chia theo mã (CDMA – Code Division Multiple Access)**

Bên truyền



Hình 8.8. Mã hoá và giải mã tín hiệu CDMA ở phía phát và phía thu

CDMA hoàn toàn khác với FDMA và TDMA. Thay vì chia một dãy tần số thành nhiều kênh truyền băng thông hẹp, CDMA cho phép mỗi trạm có quyền phát dữ liệu lên toàn bộ phổ tần của đường truyền lớn tại mọi thời điểm. Các cuộc truy cập đường truyền xảy ra đồng thời sẽ được tách biệt với nhau bởi kỹ thuật mã hoá. CDMA cũng xoá tan lo lắng cho rằng những khung dữ liệu bị đụng độ trên đường truyền sẽ bị biến dạng. Thay vào đó CDMA chỉ ra rằng nhiều tín hiệu đồng thời sẽ được cộng lại một cách tuyến tính. Kỹ thuật CDMA thường được sử dụng trong các kênh truyền quảng bá không dây (mạng điện thoại di động, vệ tinh...).

*Nhận xét:*

– Đầu tiên, chúng ta phải giả sử rằng tất cả các dãy chip được đồng bộ hoá để được gửi nhận cùng thời điểm. Nhưng trong thực tế, kiểu đồng bộ hoá như vậy là không thể có được. Những gì người ta có thể làm được để đồng bộ hoá là: người gửi và người nhận đồng bộ hoá với nhau bằng cách cho người gửi gửi một dãy chip được định nghĩa trước, dãy này phải đủ dài để cho bên nhận có thể theo kịp bên gửi. Tất cả các cuộc truyền nhận khác được xem như là nhiễu ngẫu nhiên. Người ta chứng minh được rằng, chuỗi chip càng dài thì xác suất phát hiện ra chuỗi này một cách chính xác là càng cao với sự hiện diện của nhiễu.

– Cũng cần phải giả thiết rằng: Bên nhận biết chính xác bên gửi là ai. Tuy trong thực tế, cần phải nói rằng: đặt giả thiết thì dễ hơn là làm. Nhưng hãy tin tưởng là CDMA có nhiều chi tiết phức tạp hơn và thông minh hơn để làm được chuyện đó.

### 8.3.2. Phương pháp truy nhập ngẫu nhiên

Trong phương pháp này, người ta để cho các trạm tự do tranh chấp đường truyền chung để truyền từng khung dữ liệu một. Nếu một trạm cần gửi một khung, nó sẽ gửi khung đó trên toàn bộ dải thông của kênh truyền. Sẽ không có sự phối hợp trình tự giữa các trạm. Nếu có hơn hai trạm phát cùng một lúc, "đụng độ" (collision) sẽ xảy ra, các khung bị đụng độ sẽ bị hư hại. Giao thức truy cập đường truyền ngẫu nhiên được dùng để xác định:

- Làm thế nào để phát hiện đụng độ.
- Làm thế nào để phục hồi sau đụng độ.

Ví dụ về các giao thức truy cập ngẫu nhiên: slotted ALOHA và pure ALOHA, CSMA và CSMA/CD, CSMA/CA.

### 8.3.2.1. ALOHA

Vào những năm 1970, Norman Abramson cùng các đồng sự tại Đại học Hawaii đã phát minh ra một phương pháp mới dùng để giải quyết bài toán về cấp phát kênh truyền. Sau đó công việc của họ tiếp tục được mở rộng bởi nhiều nhà nghiên cứu khác. Mặc dù công trình của Abramson, được gọi là hệ thống ALOHA, sử dụng hệ thống truyền quảng bá trên sóng radio mặt đất, nhưng ý tưởng cơ sở của nó có thể áp dụng cho bất kỳ hệ thống nào trong đó những người dùng không phối hợp với nhau sẽ tranh chấp sử dụng đường truyền chung duy nhất.

Ở đây, chúng ta sẽ tìm hiểu về hai phiên bản của ALOHA: pure (nguyên bản) và slotted (theo khe).

#### • Slotted ALOHA

Thời gian được chia thành nhiều slot có kích cỡ bằng nhau (bằng thời gian truyền một khung). Một trạm muốn truyền một khung thì phải đợi đến đầu slot thời gian kế tiếp mới được truyền. Dĩ nhiên là sẽ xảy ra đụng độ và khung bị đụng độ sẽ bị hại. Tuy nhiên, dựa trên tính phản hồi của việc truyền quảng bá, trạm phát luôn có thể theo dõi xem khung của nó phát đi có bị huỷ hoại hay không bằng cách lắng nghe kênh truyền. Những trạm khác cũng làm theo cách tương tự. Trong trường hợp vì lý do nào đó mà trạm không thể dùng cơ chế lắng nghe đường truyền, hệ thống cần yêu cầu bên nhận trả lời một khung báo nhận (acknowledgement) cho bên phát. Nếu phát sinh đụng độ, trạm phát sẽ gửi lại khung tại đầu slot kế tiếp với xác suất  $p$  cho đến khi thành công.

*Ví dụ:* Có 3 trạm đều muốn truyền một khung thông tin.

Do sẽ có đụng độ mà mất khung thông tin, một câu hỏi đặt ra là: đâu là tỷ suất truyền khung thành công của các trạm trong mạng?

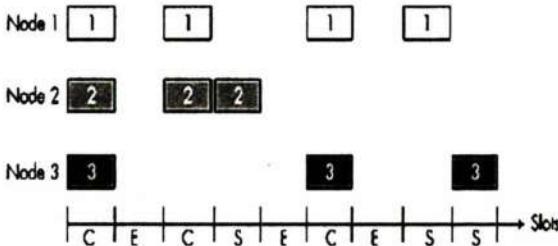
Giả sử có  $N$  trạm muốn truyền dữ liệu, mỗi trạm truyền khung thông tin của mình trong một slot với xác suất  $p$ .

Xác suất 1 nút truyền thành công =  $p(1 - p)^{N - 1}$ .

Xác suất một nút trong N nút truyền thành công =  $Np(1 - p)^{N - 1}$ .

Hiệu suất sử dụng kênh truyền là:

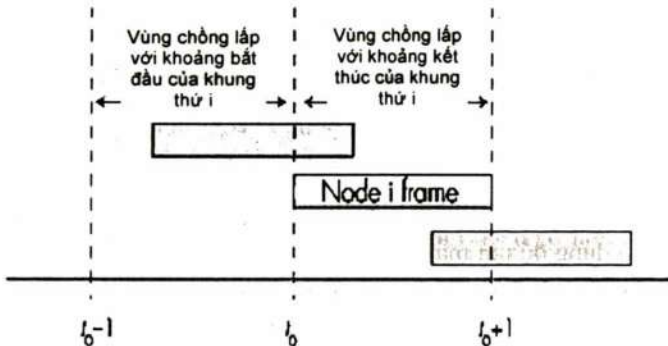
$$\frac{1}{e} = 0,37.$$



Hình 8.9. Minh hoạ giao thức Slotted ALOHA

#### • Pure ALOHA

Kỹ thuật Pure ALOHA đơn giản hơn Slotted ALOHA do không có sự đồng bộ hoá giữa các trạm. Mỗi khi muốn truyền một khung thông tin, trạm sẽ truyền nó ngay mà không cần đợi đến đầu của slot thời gian kế tiếp. Vì thế xác suất bị đụng độ tăng thêm! Nghĩa là khung thông tin được gửi tại thời điểm  $t_0$  sẽ đụng độ với những khung được gửi trong khoảng thời gian  $[t_0 - 1, t_0 + 1]$ .



Hình 8.10. Minh hoạ giao thức Pure ALOHA



Gọi  $P$  là xác suất của một sự kiện nào đó, ta có những phân tích sau:

- $P$  (một nút truyền thành công) =  $P$  (nút truyền)  $\times$ .
- $P$  (không có nút nào truyền  $t_0 - 1$  tới  $t_0$ )  $\times$ .
- $P$  (không có nút nào truyền  $t_0$  tới  $t_0 + 1$ ) =  $p(1 - p)2(N - 1)$ .
- $P$  (bất kỳ nút nào trong  $N$  nút truyền thành công) =  $NP(1 - p)2(N - 1)$ .
- Hiệu suất sử dụng tối đa kênh truyền thấp:  $1/(2e)$ .

Nhận xét chung về ALOHA:

Hiệu năng thấp do không có thăm dò đường truyền trước khi gửi khung, dẫn đến việc mất nhiều thời gian cho việc phát hiện đụng độ và phục hồi sau đụng độ. Hoạt động theo kiểu ALOHA có khả năng dẫn đến việc hệ thống bị "chết đứng" do mọi nỗ lực gửi gói tin của tất cả các trạm đều bị đụng độ.

### 8.3.2.2. CSMA (*Carrier Sense Multiple Access*)

Giao thức ALOHA mặc dù đã chạy được, nhưng một điều đáng ngạc nhiên là người ta lại để cho các trạm làm việc tự do gửi thông tin lên đường truyền mà chẳng cần quan tâm đến việc tìm hiểu xem những trạm khác đang làm gì. Và điều đó dẫn đến rất nhiều vụ đụng độ tín hiệu. Tuy nhiên, trong mạng LAN, người ta có thể thiết kế các trạm làm việc sao cho chúng có thể điều tra xem các trạm khác đang làm gì và tự điều chỉnh hành vi của mình một cách tương ứng. Làm như vậy sẽ giúp cho hiệu năng mạng đạt được cao hơn. CSMA là một giao thức như vậy!

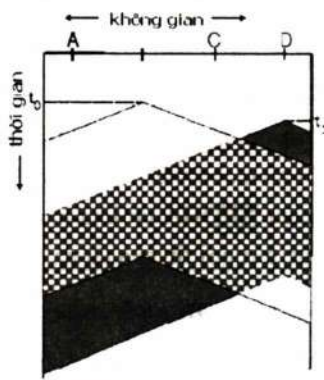
Các giao thức mà trong đó các trạm làm việc lắng nghe đường truyền trước khi đưa ra quyết định mình phải làm gì tương ứng với trạng thái đường truyền đó được gọi là các giao thức có "cảm nhận" đường truyền (carrier sense protocol). Cách thức hoạt động của CSMA như sau: lắng nghe kênh truyền, nếu thấy kênh truyền rỗi thì bắt đầu truyền khung, nếu thấy đường truyền bận thì trì hoãn lại việc gửi khung.

Thế nhưng trì hoãn việc gửi khung cho đến khi nào? Có ba giải pháp:

- Theo dõi không kiên trì (Non-persistent CSMA): Nếu đường truyền bận, đợi trong một khoảng thời gian ngẫu nhiên rồi tiếp tục nghe lại đường truyền.

• Theo dõi kiên trì (persistent CSMA): Nếu đường truyền bận, tiếp tục nghe đến khi đường truyền rỗi rồi thì truyền gói tin với xác suất bằng 1.

Theo dõi kiên trì với xác suất  $p$  ( $P$  – persistent CSMA): Nếu đường truyền bận, tiếp tục nghe đến khi đường truyền rỗi rồi thì truyền gói tin với xác suất bằng  $p$ .



**Hình 8.11.** Mô tả không gian và thời gian đựng độ

Để thấy rằng giao thức CSMA cho dù là theo dõi đường truyền kiên trì hay không kiên trì thì khả năng tránh đựng độ vẫn tốt hơn là ALOHA. Tuy thế, đựng độ vẫn có thể xảy ra trong CSMA! Tình huống phát sinh như sau: khi một trạm vừa phát xong thì một trạm khác cũng phát sinh yêu cầu phát khung và bắt đầu nghe đường truyền. Nếu tín hiệu của trạm thứ nhất chưa đến trạm thứ hai, trạm thứ hai sẽ cho rằng đường truyền đang rảnh và bắt đầu phát khung. Như vậy đựng độ sẽ xảy ra:

• Hậu quả của đựng độ là: khung bị mất và toàn bộ thời gian từ lúc đựng độ xảy ra cho đến khi phát xong khung là lãng phí!

• Bây giờ phát sinh vấn đề mới: các trạm có quan tâm theo dõi xem có đựng độ xảy ra không và khi đựng độ xảy ra thì các trạm sẽ làm gì?

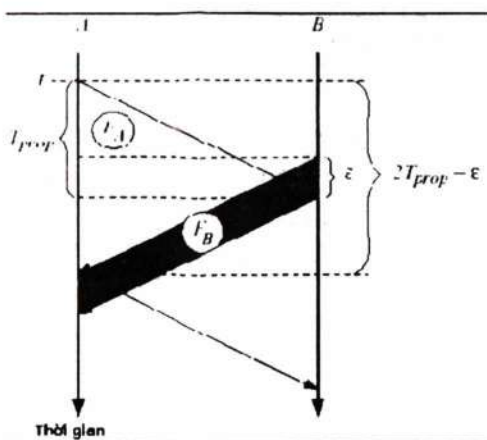
• CSMA với cơ chế theo dõi đựng độ (CSMA/CD – CSMA with Collision Detection)

• CSMA/CD về cơ bản là giống như CSMA: lắng nghe trước khi truyền. Tuy nhiên CSMA/CD có hai cải tiến quan trọng là: phát hiện đựng

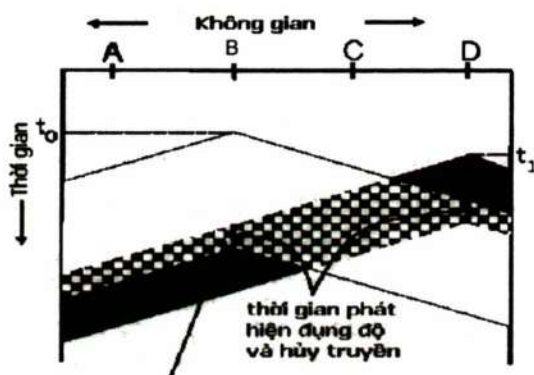
độ và làm lại sau đụng độ. Phát hiện đụng độ: Trạm vừa truyền vừa tiếp tục dò xét đường truyền.

Ngay sau khi đụng độ được phát hiện thì trạm ngưng truyền, phát thêm một dãy nhồi (dãy nhồi này có tác dụng làm tăng cường thêm sự va chạm tín hiệu, giúp cho tất cả các trạm khác trong mạng thấy được sự đụng độ), và bắt đầu làm lại sau đụng độ.

CSMA/CD, cũng giống như các giao thức trong LAN khác, sử dụng mô hình quan niệm như trong hình sau: Tại thời điểm  $t_0$ , một trạm đã phát xong khung của nó. Bất kỳ trạm nào khác có khung cần truyền bây giờ có thể cố truyền thử. Nếu hai hoặc nhiều hơn các trạm làm như vậy cùng một lúc thì sẽ xảy ra đụng độ. Đụng độ có thể được phát hiện bằng cách theo dõi năng lượng hay độ rộng của xung của tín hiệu nhận được và đem so sánh với độ rộng của xung vừa truyền đi. Bây giờ ta đặt ra câu hỏi: Sau khi truyền xong khung (hết giai đoạn truyền), trạm sẽ bỏ ra thời gian tối đa là bao lâu để biết được là khung của nó đã bị đụng độ hoặc nó đã truyền thành công? Gọi thời gian này là "cửa sổ va chạm" và ký hiệu nó là  $T_w$ . Phân tích sau đây sẽ cho ra câu trả lời. Hình sau sẽ mô phỏng chi tiết về thời gian phát khung giữa hai trạm A và B ở hai đầu mút xa nhất trên đường truyền tải.



Hình 8.12. Thời gian cần thiết để truyền một khung



thay vì lãng phí thời gian để truyền hết khung bị đụng độ, hủy bỏ việc truyền ngay sau khi đụng độ xảy ra

Hình 8.13. Xử lý khu đụng độ

Đặt  $T_{prop}$  là thời gian lan truyền tín hiệu giữa hai đầu mút xa nhau nhất trên đường truyền tải.

- Tại thời điểm  $t$ , A bắt đầu phát đi khung dữ liệu của nó.
- Tại  $t + T_{prop} - \epsilon$ , B phát hiện kênh truyền rảnh và phát đi khung dữ liệu của nó.
- Tại  $t + T_{prop}$ , B phát hiện sự đụng độ.



- Tại  $t + 2T_{prop} - \epsilon$ , A phát hiện sự đụng độ.
- Theo phân tích trên, thì  $T_w = 2T_{prop}$ .

Làm lại sau khi đụng độ: Sau khi bị đụng độ, trạm sẽ chạy một thuật toán gọi là back-off dùng để tính toán lại lượng thời gian nó phải chờ trước khi gửi lại khung. Lượng thời gian này phải là ngẫu nhiên để các trạm sau khi quay lại không bị đụng độ với nhau nữa. Thuật toán back-off hoạt động như sau:

- Rút ngẫu nhiên ra một số nguyên  $M$  thoả mãn:  $0 \leq M \leq 2k$ . Trong đó,  $k = \min(n, 10)$  với  $n$  là tổng số lần đụng độ mà trạm đã gánh chịu.

- Kỳ hạn mà trạm phải chờ trước khi thử lại một lần truyền mới là  $M * T_w$ .

- Khi mà  $n$  đạt đến giá trị 16 thì huỷ bỏ việc truyền khung. (Trạm đã chịu đựng quá nhiều vụ đụng độ rồi, và không thể chịu đựng hơn được nữa!)

- Đánh giá hiệu suất của giao thức CSMA/CD:

- Gọi:  $P$  là kích thước của khung, ví dụ như 1000 bit.

- $C$  là dung lượng của đường truyền, ví dụ như 10 Mbps.

Ta có thời gian phát hết một khung thông tin là  $P/C$  giây.

Trung bình, chúng ta sẽ thử  $e$  lần trước khi truyền thành công một khung. Vì vậy, với mỗi lần phát thành công một khung (tốn  $P/C$  giây), ta đã mất tổng cộng  $2eT_{prop}$  ( $\approx 5T_{prop}$ ) vì đụng độ.

Do đó hiệu năng của giao thức (tỷ lệ giữa thời gian hoạt động hữu ích trên tổng thời gian hoạt động) là:

$$\frac{\frac{P}{C}}{\frac{P}{C} + 5T_{prop}} = \frac{1}{1 + \frac{5T_{prop}}{\frac{P}{C}}} = \frac{1}{1 + 5a}$$

với  $a = \frac{T_{prop} C}{P}$ .

Giá trị của  $a$  đóng vai trò rất quan trọng đến hiệu suất hoạt động của mạng kiểu CSMA/CD.

### 8.3.3. Phương pháp phân lượt truy nhập đường truyền

Bây giờ thử nhìn lại hai phương pháp điều khiển truy cập đường truyền "chia kênh" và "truy cập ngẫu nhiên", ta sẽ thấy chúng đều có những điểm hay và hạn chế:

Trong các giao thức dạng chia kênh, kênh truyền được phân chia một cách hiệu quả và công bằng khi tải trọng đường truyền là lớn. Tuy nhiên chúng không hiệu quả khi tải trọng của đường truyền là nhỏ: có độ trì hoãn khi truy cập kênh truyền, chỉ  $1/N$  băng thông được cấp cho người dùng ngay cả khi chỉ có duy nhất người dùng đó hiện diện trong hệ thống.

Các giao thức dạng truy cập ngẫu nhiên thì lại hoạt động hiệu quả khi tải trọng của đường truyền thấp. Nhưng khi tải trọng đường truyền cao thì phải tốn nhiều chi phí cho việc xử lý đụng độ. Các giao thức dạng "phân lượt" sẽ đề ý đến việc tận dụng những mặt mạnh của hai dạng nói trên. Ý tưởng chính của các giao thức dạng "phân lượt" là không để cho đụng độ xảy ra bằng cách cho các trạm truy cập đường truyền một cách tuần tự.

Về cơ bản, có hai cách thức để "phân lượt" sử dụng đường truyền:

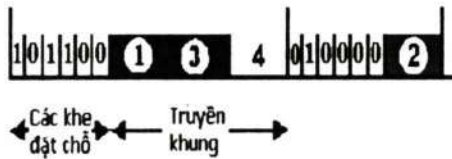
- Thăm dò (polling): Trạm chủ (master) sẽ mời các trạm tớ (slave) truyền khi đến lượt. Lượt truyền được cấp phát cho trạm tớ có thể bằng cách: trạm chủ dành phần cho trạm tớ hoặc trạm tớ yêu cầu và được trạm chủ đáp ứng. Tuy nhiên có thể thấy những vấn đề sẽ gặp phải của giải pháp này là: chi phí cho việc thăm dò, độ trễ do phải chờ được phân lượt truyền, hệ thống rối loạn khi trạm chủ gặp sự cố.

- Truyền thẻ bài (token passing): Thẻ bài điều khiển sẽ được chuyển lần lượt từ trạm này qua trạm kia. Trạm nào có trong tay thẻ bài sẽ được quyền truyền, truyền xong phải truyền thẻ bài qua trạm kế tiếp. Những vấn đề cần phải quan tâm: chi phí quản lý thẻ bài, độ trễ khi phải chờ thẻ bài, khó khăn khi thẻ bài bị mất.

#### 8.3.3.1. Thăm dò phân tán (Distributed Polling)

Thời gian được chia thành những "khe" (slot). Giả sử hệ thống hiện có  $N$  trạm làm việc. Một chu kỳ hoạt động của hệ thống bắt đầu bằng  $N$  khe

thời gian ngắn dùng để đặt chỗ (reservation slot). Khe thời gian dùng để đặt chỗ bằng với thời gian lan truyền tín hiệu giữa hai đầu nút xa nhất trên đường truyền. Tới khe đặt chỗ thứ  $i$ , trạm thứ  $i$  nếu muốn truyền dữ liệu sẽ phát tín hiệu đặt chỗ của mình lên kênh truyền, và tín hiệu này sẽ được nhìn thấy bởi tất cả các trạm khác trong mạng. Sau thời gian đặt chỗ, các trạm bắt đầu việc truyền dữ liệu của mình theo đúng trình tự đã đăng ký.



Hình 8.14. Mô tả các chu kỳ hoạt động của hệ thống thăm dò phân tán

### 8.3.3.2. Phương pháp chuyển thẻ bài: Token Ring

Giao thức này sử dụng mạng kiểu hình vòng, dùng thẻ bài để cấp quyền sử dụng đường truyền. Mạng token ring bao gồm một tập hợp các trạm được nối với nhau thành một vòng. Dữ liệu luôn chạy theo một hướng vòng quanh vòng. Mỗi trạm nhận khung từ trạm phía trên của nó và rồi chuyển khung đến trạm phía dưới. Thẻ bài là công cụ để quyết định ai có quyền truyền tại một thời điểm.

Cách thức hoạt động của mạng token ring như sau: một thẻ bài, thực chất chỉ là một dãy bit, sẽ chạy vòng quanh vòng; mỗi nút sẽ nhận thẻ bài rồi lại chuyển tiếp thẻ bài này đi. Khi một trạm có khung cần truyền và đúng lúc nó thấy có thẻ bài tới, nó liền lấy thẻ bài này ra khỏi vòng (nghĩa là không có chuyển tiếp chuỗi bit đặc biệt này lên vòng nữa), và thay vào đó, nó sẽ truyền khung dữ liệu của mình đi. Khi khung dữ liệu đi một vòng và quay lại, trạm phát sẽ rút khung của mình ra và chèn lại thẻ bài vào vòng. Hoạt động cứ xoay vòng như thế. Card mạng dùng cho token ring sẽ có trên đó một bộ nhận, một bộ phát và một bộ đệm dùng chứa dữ liệu. Khi không có trạm nào trong vòng có dữ liệu để truyền, thẻ bài sẽ lưu chuyển vòng quanh. Nếu một trạm có dữ liệu cần truyền và có thẻ bài, nó có quyền truyền một hoặc nhiều khung dữ liệu tùy theo quy định của hệ thống. Mỗi khung dữ liệu được phát đi sẽ có một phần thông tin chứa địa chỉ đích của trạm bên



nhận; ngoài ra nó còn có thể chứa địa chỉ multicast hoặc broadcast tùy theo việc bên gửi muốn gửi khung cho một nhóm người nhận hay tất cả mọi người trong vòng. Khi khung thông tin chạy qua mỗi trạm trong vòng, trạm này sẽ nhìn vào địa chỉ đích trong khung đó để biết xem có phải nó là đích đến của khung không. Nếu phải, trạm sẽ chép nội dung của khung vào trong bộ đệm của nó, chỉ chép thôi chứ không được xoá khung ra khỏi vòng. Một vấn đề cần phải quan tâm đến là một trạm đang giữ thẻ bài thì nó có quyền truyền bao nhiêu dữ liệu, hay nói cách khác là trạm được cho bao nhiêu thời gian để truyền dữ liệu? Chúng ta gọi thời gian này là thời gian giữ thẻ bài – THT (Token Holding Time). Trong trường hợp trong vòng chỉ có một trạm cần truyền dữ liệu và các trạm khác không có nhu cầu truyền, thì ta có thể cấp THT cho trạm có nhu cầu càng lâu càng tốt. Điều này sẽ làm tăng hiệu suất sử dụng hệ thống một cách đáng kể. Bởi vì sẽ thật là ngớ ngẩn nếu bắt trạm ngừng, chờ thẻ bài chạy hết một vòng, rồi lại truyền tiếp. Tuy nhiên, giải pháp trên sẽ không hoạt động tốt nếu có nhiều trạm trong vòng cần gửi dữ liệu. THT dài chỉ thích hợp cho những trạm cần truyền nhiều dữ liệu, nhưng lại không phù hợp với những trạm chỉ có ít thông điệp cần gửi đi ngay cả khi thông điệp này là tối quan trọng. Điều này cũng giống như tình huống mà bạn xếp hàng để sử dụng máy ATM ngay sau một anh chàng định rút ra 10 triệu đồng, trong khi bạn chỉ cần vào đây để kiểm tra tài khoản của mình còn bao nhiêu tiền! Trong các mạng 802.5, THT mặc định là 10 ms. Từ thời gian giữ thẻ bài, chúng ta lại nghĩ ra một số đo quan trọng khác: Thời gian xoay vòng của thẻ bài – TRT (*Token rotation time*), nghĩa là lượng thời gian bỏ ra để thẻ bài đi hết đúng một vòng. Để nhận thấy rằng:

$$TRT \leq \text{Số nút hoạt động} \times THT + \text{Độ trễ của vòng}$$

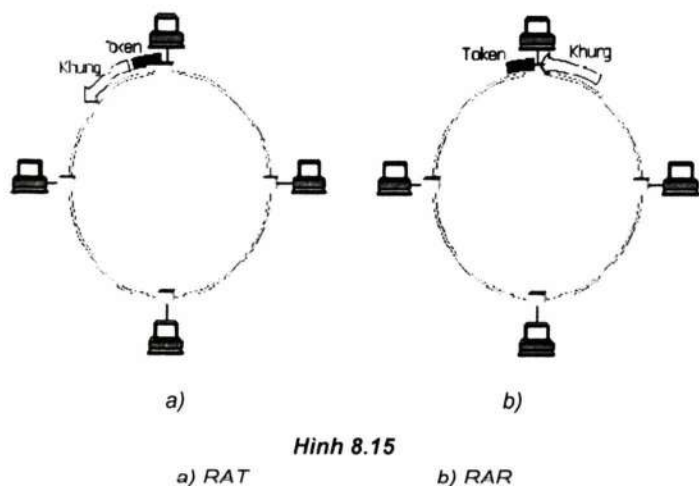
Với "Độ trễ của vòng" là tổng thời gian để thẻ bài đi hết một vòng khi trong vòng không có trạm nào cần truyền dữ liệu, "Số nút hoạt động" ám chỉ số trạm có dữ liệu cần truyền.

Giao thức 802.5 cung cấp một phương thức truyền dữ liệu tin cậy bằng cách sử dụng hai bit A và C ở đuôi của khung dữ liệu. Hai bit này ban đầu nhận giá trị 0. Khi một trạm nhận ra nó là đích đến của một khung dữ liệu, nó sẽ đặt bit A trong khung này lên. Khi trạm chép khung vào trong bộ nhớ



đệm của nó, nó sẽ đặt bit C lên. Khi trạm gửi thấy khung của nó quay lại với bit A vẫn là 0, nó biết là trạm đích bị hư hỏng hoặc không có mặt. Nếu bit A là 1, nhưng bit C là 0, điều này ám chỉ trạm đích có mặt nhưng vì lý do nào đó trạm đích không thể nhận khung (ví dụ như thiếu bộ đệm chẳng hạn). Vì thế khung này có thể sẽ được truyền lại sau đó với hy vọng là trạm đích có thể tiếp nhận nó. Chi tiết cuối cùng cần phải xem xét là: chính xác khi nào thì trạm sẽ nhả thẻ bài ra? Có hai đề nghị:

- Nhà thẻ bài ra ngay sau khi trạm vừa truyền khung xong (RAT);
- Nhà thẻ bài ra ngay sau khi trạm nhận lại khung vừa phát ra (RAR).



Hình 8.15

a) RAT

b) RAR

### 8.3.3.3. Quản lý hoạt động của mạng Token Ring

Cần thiết phải đề cử ra một trạm làm nhiệm vụ quản lý mạng token ring gọi là monitor. Công việc của monitor là đảm bảo sức khoẻ cho toàn bộ vòng. Bất kỳ trạm nào cũng có thể trở thành monitor. Thủ tục bầu chọn monitor diễn ra khi vòng vừa được tạo ra hoặc khi monitor của vòng bị sự cố. Một monitor mạnh khoẻ sẽ định kỳ thông báo sự hiện diện của nó cho toàn vòng bằng một thông điệp đặc biệt. Nếu một trạm không nhận được thông báo hiện diện của monitor trong một khoảng thời gian nào đó, nó sẽ coi như monitor bị hỏng và sẽ cố trở thành monitor mới.

Khi một trạm quyết định rằng cần phải có một monitor mới, nó sẽ gửi một thông điệp thỉnh cầu, thông báo ý định trở thành monitor của mình. Nếu thông điệp này chạy một vòng và về lại được trạm, trạm sẽ cho rằng mọi người đồng ý vị trí monitor của nó. Còn nếu đồng thời có nhiều trạm cùng gửi thông điệp thỉnh cầu, chúng sẽ phải áp dụng một luật lựa chọn nào đó, chẳng hạn như "ai có địa chỉ cao nhất sẽ thắng cử".

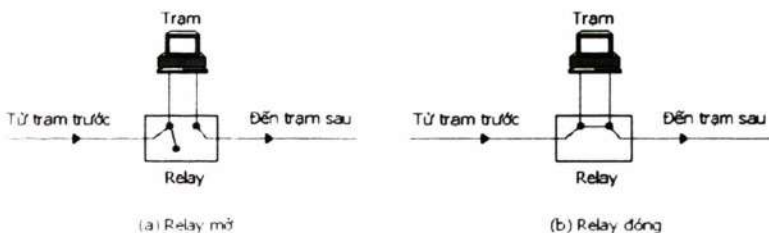
Nhiệm vụ đáng chú ý của monitor là phải đảm bảo rằng luôn luôn có sự hiện diện của thẻ bài ở đâu đó trên vòng, có thể là đang di chuyển hay đang bị giữ bởi một trạm nào đó. Rõ ràng là thẻ bài có thể bị biến mất vì lý do nào đó chẳng hạn như lỗi bit, trạm đang giữ nó bị hư hỏng. Để phát hiện ra việc thẻ bài bị mất, khi thẻ bài chạy ngang qua monitor, nó sẽ bật một bộ đếm thời gian để tính giờ. Bộ đếm này có giá trị tối đa là:

$$\text{Số lượng trạm} \times \text{THT} + \text{Độ trễ của vòng}$$

Trong đó "*Số lượng trạm*" là số các trạm làm việc đang hiện diện trên vòng, "*độ trễ của vòng*" là tổng thời gian lan truyền tín hiệu trên vòng. Nếu bộ đếm đạt đến giá trị tối đa mà monitor vẫn không thấy thẻ bài chạy qua nó nữa thì nó sẽ tạo ra thẻ bài mới.

Monitor cũng phải kiểm tra xem có khung nào bị hỏng hoặc vô thừa nhận hay không. Một khung nếu có lỗi checksum hoặc khuôn dạng không hợp lệ sẽ chạy một cách vô định trên vòng. Monitor sẽ thu khung này lại trước khi chèn lại thẻ bài vào vòng. Một khung vô thừa nhận là khung mà đã được chèn thành công vào vòng, nhưng cha của nó bị chết, nghĩa là trạm gửi nó chỉ gửi nó lên vòng, nhưng chưa kịp thu nó lại thì đã bị chết (down). Những khung như vậy sẽ bị phát hiện bằng cách thêm vào một bit điều khiển gọi là monitor bit. Khi được phát lần đầu tiên, monitor bit trên khung sẽ nhận giá trị 0. Khi khung đi ngang qua monitor, monitor sẽ đặt monitor bit lên 1. Nếu monitor thấy khung này lại chạy qua nó với monitor bit là 1, nó sẽ rút khung này ra khỏi vòng. Một chức năng quản lý vòng khác là phát hiện ra một trạm bị chết. Nếu một trạm trong vòng bị chết, nó sẽ làm đứt vòng. Để tránh tình trạng này người ta thêm vào trạm một role điện tử (relay). Khi trạm còn mạnh khỏe, role sẽ mở và trạm được nối với vành, khi

trạm bị chết và ngưng không cung cấp năng lượng cho role, role sẽ tự động đóng mạch và bỏ qua trạm này.



Hình 8.16. Sử dụng relay để tránh đứt vòng

Khi monitor nghi ngờ một trạm bị chết, nó sẽ gửi đến trạm đó một khung đặc biệt gọi là khung beacon. Nếu không nhận được trả lời thích đáng, monitor sẽ coi trạm đó đã chết.

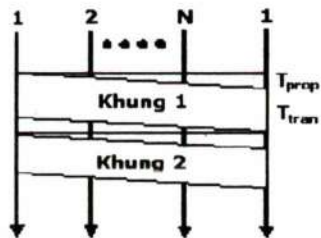
• Đánh giá về mạng Token Ring

Ta sẽ khảo sát hai kiểu chuyển thế bài: *Release After Reception (RAR)* và *Release After Transmissions (RAT)*.

- RAR: Nhà thế bài sau khi nhận lại dữ liệu.

Sau khi một trạm phát đi khung dữ liệu của nó, trạm sẽ chờ đến khi khung này quay trở lại mới truyền thế bài cho trạm kế tiếp. Mạng IEEE 802.5 Token Ring (16Mbps) sử dụng cơ chế này.

Ta gọi hiệu suất truyền khung là  $\eta_{RAR}$ . Mạng kiểu RAR sẽ đạt được hiệu suất tối đa nếu một trạm phát liên tục.



Hình 8.17. Mô phỏng cơ chế truyền thế bài trong RAR

Đặt:  $T_{prop}$  là thời gian lan truyền tín hiệu giữa hai đầu nút xa nhau nhất trên đường truyền tải.

$T_{tran}$  là thời gian để phát hết một khung dữ liệu lên đường truyền.

$P$  là kích thước của khung dữ liệu, ví dụ 1000 bit.

$C$  là dung lượng của đường truyền, ví dụ 10 Mbps.

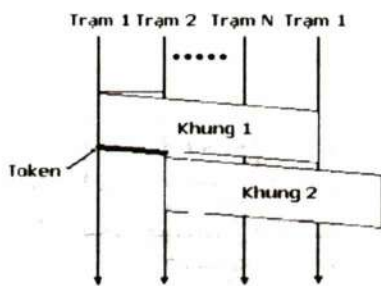
Sau đây là biểu đồ mô phỏng mối liên quan giữa thời gian phát khung và thời gian truyền tín hiệu:

Có thể thấy:

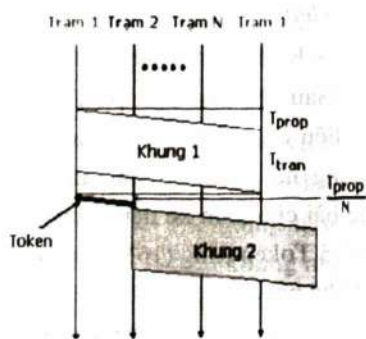
$$n_{RAR} = \frac{1}{1 + \alpha} \quad \text{với} \quad \alpha = \frac{T_{prop}}{T_{tran}}$$

Trong trường hợp một trạm luôn phải nhường token ngay sau khi token đi hết một vòng và quay trở lại nó thì hiện trạng phân bố thời gian được vẽ ra như hình 8.18, 8.19.

Giả sử token có kích thước quá nhỏ để coi như thời gian phát nó bằng không, mạng có  $N$  trạm làm việc và khoảng cách giữa các trạm là bằng nhau. Vì vậy, thời gian lan truyền tín hiệu từ một trạm đến một trạm liền kề nó là  $T_{prop}/N$ . Có thể nhận thấy thời gian để token chuyển từ một trạm sang một trạm kề nó là  $T_{prop}/N$ .



Hình 8.18. Thời gian lan truyền và phát khung với RAR



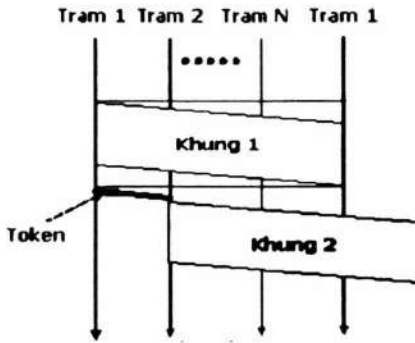
Hình 8.19. Các khoảng thời gian một trạm phải trải qua khi gửi dữ liệu và truyền thẻ bài

Vi vậy:

$$n_{RAR} = \frac{T_{tran}}{T_{prop} + T_{tran} + \dots + \frac{T_{prop}}{N}} = \frac{1}{1 + \left(\frac{N+1}{N}\right) \frac{T_{prop}}{T_{tran}}}$$



- RAT: Nhà thẻ bài ngay sau khi truyền dữ liệu:



Hình 8.20. Mô phỏng kỹ thuật RAT

Với kỹ thuật RAT, trạm sẽ truyền thẻ bài điều khiển cho trạm kế tiếp ngay sau khi nó vừa phát xong khung dữ liệu

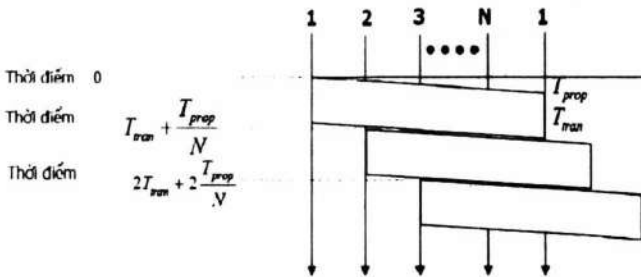
Vì dụ mạng FDDI (Fiber Distributed Data Interface – 100Mbps) sử dụng kỹ thuật này.

- Gọi hiệu suất truyền khung là  $\eta_{RAT}$ .
- Với kỹ thuật RAT, hiệu suất đạt tối đa khi một trạm liên tục truyền khung.

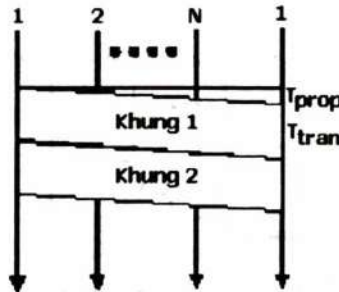
Sau đây là biểu đồ thời gian mô phỏng:

Có thể thấy khi một trạm phát khung liên tục thì  $\eta_{RAT} = 1$ .

Tuy nhiên khi một trạm buộc phải nhận token ngay sau khi nó vừa phát dữ liệu xong, thì biểu đồ thời gian có khác.



Hình 8.21. Thời gian phát và lan truyền



**Hình 8.21.** Khi một trạm phải nhà token ngay sau khi nó vừa phát dữ liệu

Ta tính lại hiệu suất như sau:

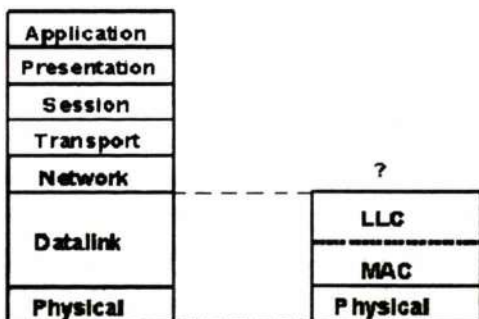
$$n_{RAT} = \frac{T_{tran}}{T_{tran} + \frac{T_{prop}}{N}} = \frac{1}{1 + \frac{1}{N} \frac{T_{prop}}{T_{tran}}} = \frac{1}{1 + \frac{\alpha}{N}}$$

## 8.4. CÔNG NGHỆ MẠNG CỤC BỘ – LAN

### 8.4.1. Chuẩn hoá mạng cục bộ

Ngoài mô hình OSI dùng cho việc chuẩn hoá các mạng nói chung, việc chuẩn hoá mạng cục bộ cũng đã được thực hiện trong một khoảng thời gian dài. Do đặc trưng riêng, việc chuẩn hoá mạng cục bộ chỉ được thực hiện trên hai tầng thấp nhất, tương ứng với tầng vật lý và liên kết dữ liệu trong mô hình OSI.

Trong LAN, tầng liên kết dữ liệu được chia làm hai tầng con: LLC (Logical Link Layer) và MAC. MAC quản lý việc truy cập đường truyền, trong khi LLC đảm bảo tính độc lập của việc quản lý các liên kết dữ liệu với đường truyền vật lý và phương pháp truy cập đường truyền MAC. IEEE (Institute of Electrical and Electronic Engineers) là tổ chức đi tiên phong trong lĩnh vực chuẩn hoá mạng cục bộ với dự án IEEE 802 nổi tiếng bắt đầu được triển khai từ năm 1980 và kết quả là hàng loạt chuẩn thuộc họ IEEE 802.x ra đời, tạo nền tảng quan trọng cho việc thiết kế và cài đặt mạng nội bộ trong thời gian qua. Vị trí của họ chuẩn này càng cao hơn khi ISO đã xem xét và tiếp nhận chúng thành chuẩn quốc tế mang tên 8802.x.



Mô hình tham khảo OSI      Mô hình tham khảo cho mạng LAN

*Hình 8.23. Mô hình phân tầng của mạng cục bộ*

Đến nay họ IEEE 802.x bao gồm các chuẩn sau:

- IEEE 802.1: High Level Interface.
- IEEE 802.2: Logical Link Control (LLC).
- IEEE 802.3: CSMA/CD.
- IEEE 802.4: Token bus.
- IEEE 802.5: Token ring.
- IEEE 802.6: MAN.
- IEEE 802.7: Broadband Technical Advisory Group.
- IEEE 802.8: Fiber Technical Advisory Group.
- IEEE 802.9: Intergrated Data and Voice Network.
- IEEE 802.10: Standard for Interoperable LAN security.
- IEEE 802.11: Wireless LAN.
- IEEE 802.12: 100VG – AnyLAN.
- ...

Trong đó:

- IEEE 802.1 là chuẩn đặc tả kiến trúc mạng, nối kết giữa các mạng và việc quản trị mạng đối với mạng cục bộ.
- IEEE 802.2 là chuẩn đặc tả tầng LLC (dịch vụ, giao thức) của mạng cục bộ. Có 3 kiểu giao thức LLC chính được định nghĩa:

- + LLC type 1: Là giao thức kiểu không liên kết, không báo nhận.
- + LLC type 2: Là giao thức kiểu có liên kết.
- + LLC type 3: Là giao thức dạng không liên kết, có báo nhận.

Các giao thức này được xây dựng dựa theo phương thức cân bằng của giao thức HDLC và có các khuôn dạng dữ liệu và các chức năng tương tự, đặc biệt là trong trường hợp LLC-type 2.

- IEEE 802.3: Là chuẩn đặc tả một mạng cục bộ dựa trên mạng Ethernet nổi tiếng do Digital, Intel và Xerox hợp tác phát triển từ năm 1990. IEEE 802.3 bao gồm cả tầng vật lý và tầng con MAC với các đặc tả sau:

- + Đặc tả dịch vụ MAC.
- + Giao thức MAC.
- + Đặc tả vật lý độc lập với đường truyền.
- + Đặc tả vật lý phụ thuộc vào đường truyền.

+ Phần cốt lõi của IEEE 802.3 là giao thức MAC dựa trên phương pháp CSMA/CD đã trình bày ở phần trước.

- IEEE 802.4 là chuẩn đặc tả mạng cục bộ với hình trạng bus sử dụng thẻ bài để điều khiển truy cập đường truyền. IEEE 802.4 cũng bao gồm cả tầng vật lý và tầng con MAC với các đặc tả sau:

- + Đặc tả dịch vụ MAC.
- + Giao thức MAC.
- + Đặc tả dịch vụ tầng vật lý.
- + Đặc tả thực thể tầng vật lý.
- + Đặc tả đường truyền.

- IEEE 802.5 là chuẩn đặc tả mạng cục bộ với hình trạng vòng sử dụng thẻ bài để điều khiển truy cập đường truyền. IEEE 802.5 cũng bao gồm cả tầng vật lý và tầng con MAC với các đặc tả sau:

- + Đặc tả dịch vụ MAC.
- + Giao thức MAC.
- + Đặc tả thực thể tầng vật lý.
- + Đặc tả nối trạm.



– IEEE 802.6: là chuẩn đặc tả một mạng tốc độ cao nối kết nhiều LAN thuộc các khu vực khác nhau của một đô thị. Mạng này sử dụng cáp quang với hình trạng dạng bus kép (dual-bus), vì thế còn được gọi là DQDB (Distributed Queue Dual Bus). Lưu thông trên mỗi bus là một chiều và khi cả cặp bus cùng hoạt động sẽ tạo thành một cấu hình chịu lỗi. Phương pháp điều khiển truy cập dựa theo một giải thuật xếp hàng phân tán có tên là QPDS (Queued-Packet, Distributed-Switch).

– IEEE 802.9: là chuẩn đặc tả một mạng tích hợp dữ liệu và tiếng nói bao gồm 1 kênh dự bộ 10 Mbps cùng với 95 kênh 64 Kbps. Dải thông tổng cộng 16 Mbps. Chuẩn này được thiết kế cho các môi trường có lưu lượng lưu thông lớn và cấp bách.

– IEEE 802.10: là chuẩn đặc tả về an toàn thông tin trong các mạng cục bộ có khả năng liên tác (interoperable).

– IEEE 802.11: là chuẩn đặc tả mạng LAN không dây (Wireless LAN). Xu hướng chọn phương pháp truy cập CSMA được khẳng định.

– IEEE 802.12: Là chuẩn đặc tả mạng cục bộ dựa trên công nghệ được đề xuất bởi AT&T, IBM và HP, gọi là 100 VG – AnyLAN. Mạng này sử dụng hình trạng mạng hình sao và một phương pháp truy cập đường truyền có điều khiển tranh chấp. Khi có nhu cầu truyền dữ liệu, trạm sẽ gửi yêu cầu đến hub và trạm chỉ có thể truyền dữ liệu khi được hub cho phép. Chuẩn này nhằm cung cấp một mạng tốc độ cao (100 Mbps và có thể lớn hơn) có thể hoạt động trong các môi trường hỗn hợp Ethernet và Token Ring, bởi thế nó chấp nhận cả hai dạng khung. 100 VG – AnyLAN là đối thủ cạnh tranh đáng gờm của 100BASE-T (Fast Ethernet) nhờ một số tính năng trội hơn, chẳng hạn về khoảng cách di cấp tối đa cho phép...

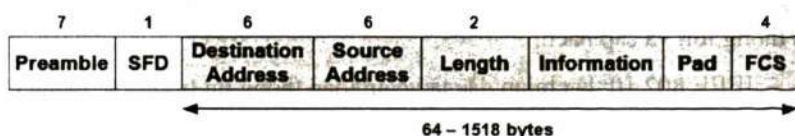
### 8.4.2. Ethernet 802.3

Ethernet là công nghệ của mạng LAN cho phép truyền tín hiệu giữa các máy tính với tốc độ 10Mb/s đến 10 Gigabit/s. Trong các kiểu Ethernet thì kiểu sử dụng cáp xoắn đôi là thông dụng nhất. Hiện nay có khoảng 85% mạng LAN sử dụng công nghệ Ethernet.

Năm 1980, Xerox, tập đoàn Intel và tập đoàn Digital Equipment đưa ra tiêu chuẩn Ethernet 10 Mbps (Tiêu chuẩn DIX). Ban đầu, một phân đoạn mạng của Ethernet (Ethernet segment) được cài đặt trên một sợi cable đồng trục dài tối đa 500 m. Các trạm nối vào Ethernet segment bằng cách "mắc dây" (tap) nối vào nó. Các điểm đầu nối phải cách nhau ít nhất 2,5 m. Transceiver, một thiết bị nhỏ được gắn trực tiếp vào điểm đầu nối, làm nhiệm vụ.

#### 8.4.2.1. Khuôn dạng gói tin Ethernet

##### IEEE 802.3 MAC Frame



Hình 8.24. Khuôn dạng gói tin Ethernet

– Preamble: dài 7 byte với mẫu 10101010 theo sau bởi 1 byte với mẫu 10101011, được sử dụng để đồng bộ hoá tốc độ đồng hồ giữa bên gửi và bên nhận.

– Start of Frame Delimiter (SFD): 10101011.

– Source and dest. addresses: Địa chỉ nguồn và đích, gồm 6 byte. Khung được nhận bởi tất cả các trạm trong LAN. Khung bị xoá nếu dest. Address không trùng với địa chỉ MAC của bất kỳ trạm nào hoặc không phải thuộc dạng multicast

– Length: Số lượng các byte trong trường thông tin.

– Max frame: 1518 ngoại trừ phần Preamble và FSD.

– Max information bytes: 1500 (Hexa: 05DC).

– Pad: Phần thêm vào để đảm bảo chiều dài tối thiểu là 64 byte.

– FCS: CCITT–32 CRC, tính toán đối với các trường address, length, information, và pad. NIC sẽ loại bỏ các khung dữ liệu nếu chiều dài không chính xác hoặc FCS sai DIX Ethernet frame format

– DIX: Digital, Intel, Xerox cùng nhau đưa ra khuôn dạng này.

- Type: Xác định giao thức tầng trên: IP, ARP...

0800: IP

0806: ARP

8137: Netware IPX

8160: NetBIOS

#### 8.4.2.2. Địa chỉ Ethernet

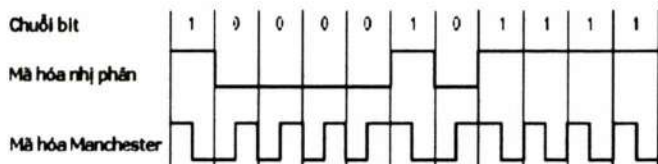
Mỗi host trong một mạng Ethernet (thật ra là tất cả các host trên thế giới) có một địa chỉ Ethernet duy nhất. Mô tả một cách kỹ thuật, địa chỉ được gắn vào card mạng chứ không phải máy tính; nó được ghi vào ROM trên card mạng. Các địa chỉ Ethernet thường được in theo thể thức mà con người có thể đọc được: một dãy gồm 6 byte được viết dưới dạng thập lục phân, cách nhau bởi dấu hai chấm. Ví dụ 8:0:2b:e4:b1:2 là cách biểu diễn dễ đọc của địa chỉ Ethernet sau 00001000 00000000 00101011 11100100 10110001 00000010

Để đảm bảo rằng mọi card mạng được gán một địa chỉ duy nhất, mỗi nhà sản xuất thiết bị Ethernet được cấp cho một phần đầu địa chỉ (prefix) khác nhau.

*Ví dụ:* Advanced Micro Devices đã được cấp phần đầu dài 24 bit x08002 (hay 8:0:2). Nhà sản xuất này sau đó phải đảm bảo phần đuôi (suffix) của các địa chỉ mà họ sản xuất ra là duy nhất. Mỗi khung được phát ra Ethernet sẽ được nhận bởi tất cả các card mạng có nối với đường truyền. Mỗi card mạng sẽ so sánh địa chỉ đích trong khung với địa chỉ của nó, và chỉ cho vào máy tính những khung nào trùng địa chỉ. Địa chỉ duy nhất như vậy gọi là địa chỉ unicast.

Ngoài ra còn có loại địa chỉ broadcast là loại địa chỉ quảng bá, có tất cả các bit đều mang giá trị 1. Mọi card mạng đều cho phép các khung thông tin có địa chỉ đích là broadcast đi đến host của nó. Cũng có một loại địa chỉ khác gọi là multicast, trong đó chỉ một vài bit đầu được đặt là 1. Một host có thể lập trình điều khiển card mạng của nó chấp nhận một số lớp địa chỉ multicast. Địa chỉ multicast được dùng để gửi thông điệp đến một tập con (subset) các host trong mạng Ethernet.

### 8.4.2.3. Cách thức mã hoá tín hiệu



Hình 8.25. Mã Manchester

Để chuyển đổi dữ liệu bit sang tín hiệu truyền trên đường truyền, Ethernet dùng kiểu mã hoá Manchester. Trong sơ đồ mã hoá Manchester, một bit sẽ được mã hoá bằng một sự thay đổi điện thế. Với bit "1", điện thế đổi từ 1 xuống 0. Còn với bit "0", điện thế đổi từ 0 lên 1.

### 8.4.2.4. Các thông số của Ethernet

#### • Ethernet ban đầu:

- Tốc độ truyền: 10 Mbps.
- Độ dài cực tiểu của khung: 512 bit = 64 byte.
- Thời gian của khe: 512 bit/10 Mbps = 51.2 $\mu$ s.
- Chiều dài cực đại: 2500m + 4 trạm lặp.

(Tốc độ bit tăng 10 lần thì khoảng cách phải giảm đi 10 lần)

#### • Ethernet theo chuẩn 802.3:

- Thời gian của khe: 512 bit/lần.
- Khoảng cách giữa các khung: 96 $\mu$ s.
- Giới hạn số lần thử: 16.
- Giới hạn số lần quay lui: 10.
- Kích thước ùn tắc: 32 bit.
- Độ dài cực đại của khung: 1518 byte.
- Độ dài cực tiểu của khung: 512 bit (64 byte).

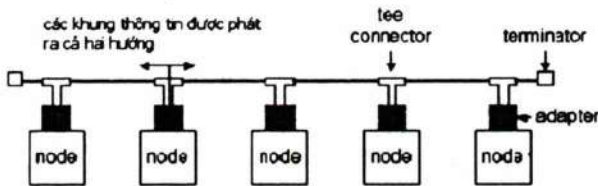
### 8.4.2.5. Các công nghệ Ethernet

#### a. Công nghệ 10BASE2

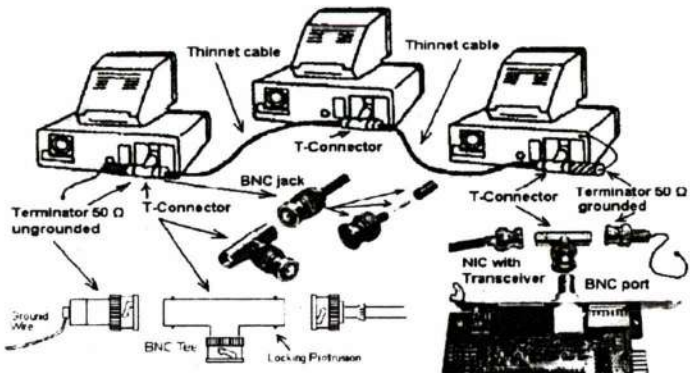
Mạng Ethernet 10BASE2 sử dụng cáp đồng trục gầy, hình thái bus. Trong trường hợp mạng có nhiều segments, các repeaters sẽ được sử dụng để nối kết các segments này lại.



- Đặc điểm:
  - Các thiết bị đầu - cuối phải có trở kháng 50 ohm.
  - Khoảng cách tối thiểu 0.5m.
  - Khoảng cách PC tới đầu nối 4cm.
  - Chiều dài tối đa 185m.
  - Dùng các bộ lặp nối rộng khoảng cách.
  - Nhỏ hơn, nhẹ hơn và mềm hơn.
  - Chi phí thấp, không yêu cầu bộ tập trung (Hub) và không cần phải cấu hình hệ thống.
  - Mã hoá Manchester,
  - Đầu nối BNC hình T.
  - Chiều dài tối đa 185m, tối thiểu là 0,5m,
  - Truyền bán song công, tốc độ truyền tối đa là 10Mbps.
  - Hỗ trợ tối đa 30 máy.



Hình 8.26. Mô hình mạng 10BASE2



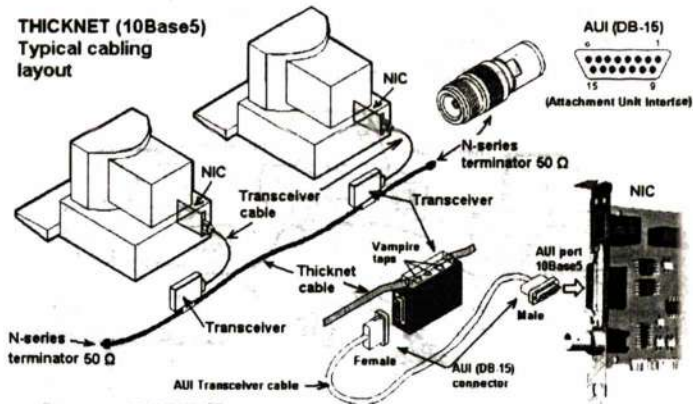
Hình 8.27. Mô hình kết nối các thiết bị trong mạng 10BASE2

### b. Công nghệ 10BASE5

- Đặc điểm:

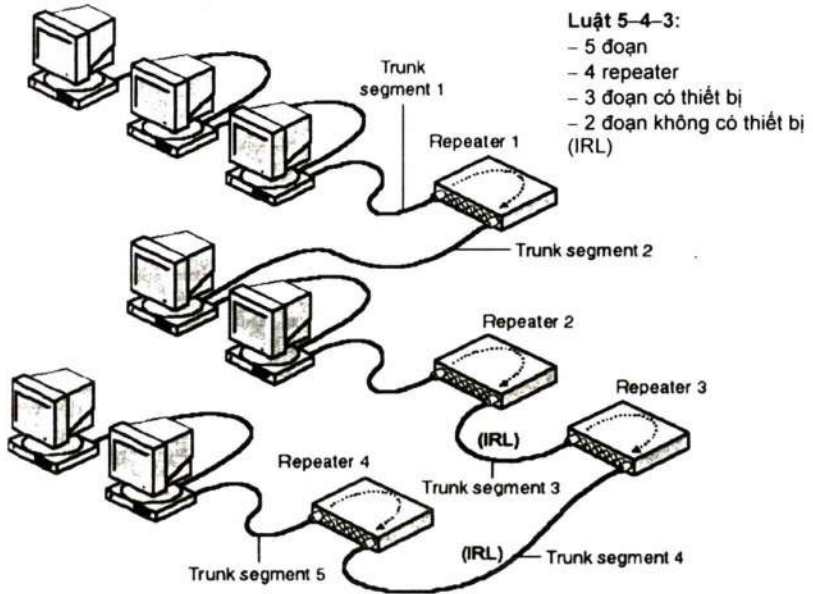
- Kết nối dạng BUS.
- Tốc độ truyền 10Mbps trên băng cơ bản – baseband (Không hỗ trợ truyền song công). Sử dụng mã hoá Manchester.
- Chiều dài tối đa: 500m.
- Hỗ trợ tối đa 101 đoạn mạng theo chuẩn Ethernet (Một đường trục và 100 nhánh mạng).
- Tối đa có 1024 trạm nối vào mạng 10BASE5.
- Tối đa 100 bộ thu phát nối vào một đoạn mạng có chiều dài 500m.
- Khoảng cách tối thiểu giữa hai bộ thu phát là 2,5m.
- Cuối mỗi đoạn cáp cần có bộ đầu - cuối (50 ohm) để dập tín hiệu.
- Bộ lặp được sử dụng để mở rộng chiều dài mạng. Sử dụng tối đa 2 bộ lặp giữa các thiết bị.

Thực tế chúng ta có thể sử dụng 4 repeater theo luật "5-4-3" do đó chiều dài tối đa có thể lên đến 2.5km.



**Hình 8.28.** Mô hình kết nối trong mạng 10BASE5

Luật 5-4-3: Giữa 2 nút trong một mạng, có tối đa 5 segment được kết nối thông qua 4 repeater và chỉ có 3 trong số 5 segment chứa các kết nối tới người dùng



Hình 8.29. Sơ đồ đi dây luật 5-4-3 dùng 4 repeater

**c. Công nghệ 10BASE – T**

• Đặc điểm:

- Sơ đồ hình sao, sử dụng bộ tập trung HUB, tốc độ 10Mbps, mã hoá Manchester.
- Chiều dài tối đa trên mỗi đoạn mạng: 100m.
- Tối đa 2 thiết bị trên một đoạn mạng: Một trạm và một hub.
- Tối đa có 2 đường nối các repeater (Inter-Repeater Links) giữa các thiết bị mà không sử dụng bridge hoặc Switch (hub là một repeater).
- Sử dụng cáp xoắn loại 2 (CAT 2) trở lên (Khuyến nghị sử dụng từ CAT 3).

Phân loại Cáp UTP (ScTP) theo chuẩn EIA/TIA 568A:

**Bảng 8.1. Chuẩn 10BASE-T**

Loại	Tốc độ tối đa	Ứng dụng
1	1 Mbps	1 cặp dây truyền tải tín hiệu thoại (không truyền dữ liệu)
2	4 Mbps	4 cặp dây xoắn đôi, truyền tín hiệu thoại và dữ liệu cho mạng IDSN, Tokenring và ARCNET
3	10 Mbps	4 cặp dây xoắn, truyền tín hiệu thoại và dữ liệu, sử dụng cho mạng 10Base-T
5	155 Mbps	4 cặp dây xoắn, tốc độ truyền cao hơn (100MHz) và khả năng nhiễu tốt hơn, sử dụng cho mạng 10Base-T, mạng 100Base-TX, mạng ARCNET tốc độ cao 160 Mbps, 16Mbps, mạng Tokenring và ATM 135Mbps
5e	1000 Mbps	Nâng cấp của cat5, được kiểm tra trên cả 4 đầu dây về ảnh hưởng trong quá trình truyền, loại bỏ được nhiễu xuyên âm. Loại này sử dụng trong mạng Gigabit Ethernet.
6	1000 Mbps	Tiêu chuẩn được đề xuất cho loại cáp có thể truyền với tần số 200MHz.
7	1000 Mbps	Tiêu chuẩn được đề xuất cho loại cáp có thể truyền với tần số 600MHz. sử dụng cáp có vỏ bảo vệ.

Chuẩn IEEE 802.3 cho cáp xoắn kép kết nối ethernet topo dạng star:

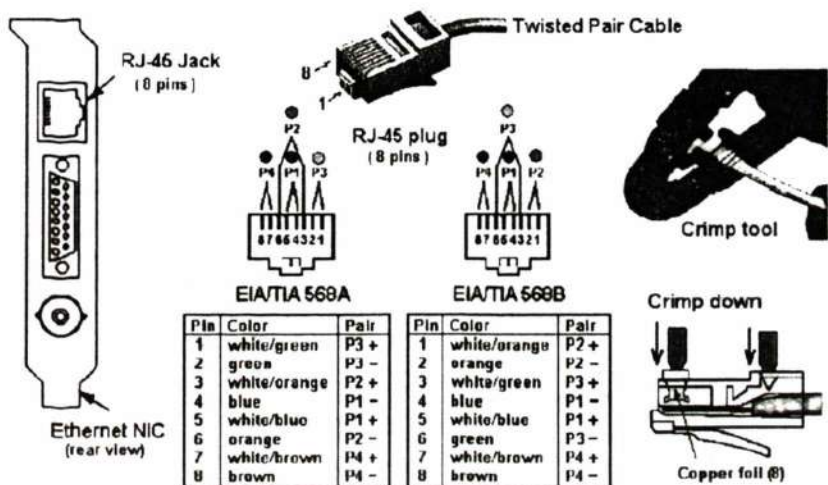
**Bảng 8.2. Chuẩn IEEE 802.3**

Chuẩn	Tốc độ ký tự	Mã hoá	Môi trường truyền	Truyền song công
10Base-T	10Mbd		2 cặp dây UTP 100Ω, loại cat3 trở lên	Hỗ trợ
100Base-TX	125Mbd	4B/5B	2 cặp dây UTP 100Ω, loại cat5 trở lên hoặc cáp STP 150Ω	Hỗ trợ
100Base-T4	33Mbd	8B/6T	4 cặp dây UTP 100Ω, loại cat3 trở lên	Không hỗ trợ
100Base-T2	25Mbd	PAM5x5	2 cặp dây UTP 100Ω, loại cat3 trở lên	Hỗ trợ
1000Base-T	125Mbd	PAM5x5	4 cặp dây UTP 100Ω, loại cat5 trở lên	Hỗ trợ
1000Base-X	1250Mbd	8B/10B	2 cặp dây UTP 150Ω.	Hỗ trợ



**Đầu nối RJ-45, chân và dây nối tới các cặp dây xoắn Ethernet:**

Pin	10Base-T Signal	100Base-TX Signal	100Base-T4 Signal	100Base-T2 Signal	1000Base-T Signal
1	TD+ (Transmit Data)	TD+ (Transmit Data)	TX_D1+ (Transmit Data)	BI_DA+ (Bidi Data)	BI_DA+ (Bidi Data)
2	TD- (Transmit Data)	TD- (Transmit Data)	TX_D1- (Transmit Data)	BI_DA- (Bidi Data)	BI_DA- (Bidi Data)
3	RD- (Receive Data)	RD- (Receive Data)	RX_D2+ (Receive Data)	BI_DB+ (Bidi Data)	BI_DB+ (Bidi Data)
4	Not used	Not used	BI_D3+ (Bidi Data)	Not used	BI_DC+ (Bidi Data)
5	Not used	Not used	BI_D3- (Bidi Data)	Not used	BI_DC- (Bidi Data)
6	RD+ (Receive Data)	RD+ (Receive Data)	RX_D2- (Receive Data)	BI_DB- (Bidi Data)	BI_DB- (Bidi Data)
7	Not Used	Not Used	BI_D4+ (Bidi Data)	Not used	BI_DD+ (Bidi Data)
8	Not Used	Not Used	BI_D4- (Bidi Data)	Not used	BI_DD- (Bidi Data)



Hình 8.30. Chuẩn mẫu dây cáp UTP và Jack đầu nối RJ-45

**d. Công nghệ Ethernet 100 Mbps**

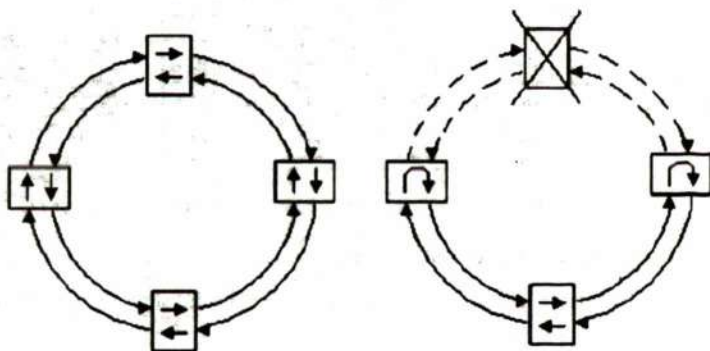
- Mạng Ethernet 100 Mbps được gồm 2 chuẩn: Fast Ethernet (IEEE 802.3u): 100Base-TX, 100Base-T4 và 100Base-FX.
- Các loại cáp sử dụng
  - 100BASE-T4 sử dụng bốn đôi dây cân bằng cáp UTP Cat-3 hoặc Cat-5.
  - 100BASE-TX sử dụng hai đôi UTP Cat-5 hoặc đôi dây STP.

- 100BASE-FX sử dụng đôi dây cáp quang đa mode.
- Mã hoá:
  - 100Base-TX và 100Base-FX sử dụng kỹ thuật mã hoá 4B/5B.
  - 100Base-T4 sử dụng kỹ thuật mã hoá 8B/6T.
- Phương thức điều khiển truy nhập CSMA/CD.
- 100Base-T4 sử dụng phương thức hoạt động bán song công.
- 100Base-TX sử dụng phương thức hoạt động song công.
- 100Base-FX sử dụng cả phương thức hoạt động song công và bán song công.
- 100VG-AnyLAN là công nghệ cạnh tranh với Fast Ethernet, ít được sử dụng.

### 8.4.3. FDDI (Fiber Distributed Data Interface)

#### 8.4.3.1. Các tính chất vật lý

Không giống như mạng 802.5, một mạng FDDI bao gồm một vòng đôi = hai vòng độc lập truyền dữ liệu theo hai chiều ngược nhau.



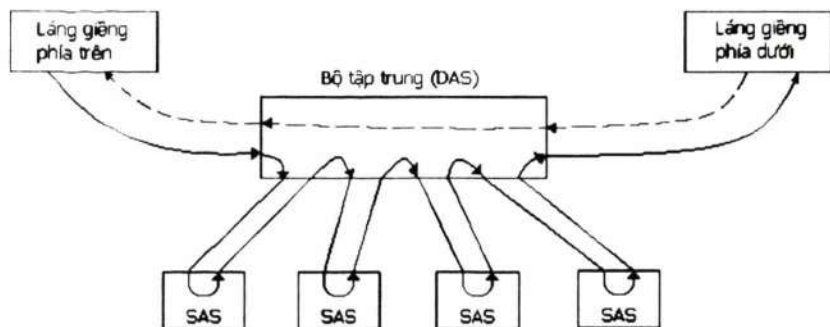
a) hoạt động bình thường; b) vòng chính bị hỏng

**Hình 8.31.** Vòng quang đôi

Vòng phụ không được sử dụng trong khi hệ thống hoạt động bình thường, nó chỉ vào cuộc khi vòng chính bị sự cố. Nghĩa là vòng chính sẽ

quanh lại vòng phụ để tạo ra một vòng hoàn chỉnh, và chính điều này giúp cho FDDI có khả năng chịu lỗi khi một công tắc bị đứt hay một trạm trong vòng bị hỏng.

Do phải chịu phí tổn khi cấu hình theo kiểu vòng đôi, nên FDDI còn cho phép một trạm chọn nối vào chỉ một vòng đơn thôi. Những trạm như vậy gọi là những "trạm nối đơn" (single attachment station – SAS). Những trạm nối ca vào hai vòng dĩ nhiên sẽ được gọi là những "trạm đầu đôi" (dual attachment station – DAS). Một bộ tập trung (concentrator) sẽ được sử dụng để nối các SAS vào vòng đôi.



Hình 8.32. Các SAS được nối vào bộ tập trung

Nếu một SAS bị hỏng hóc, bộ tập trung sẽ phát hiện ra tình trạng này và sử dụng cơ chế bỏ qua tín hiệu quang (optical bypass) để cô lập SAS bị hỏng, vì thế giữ cho vòng được thông suốt. Trong FDDI, bộ đệm của giao diện mạng có thể có kích thước khác nhau tại những trạm khác nhau, mặc dù kích thước của nó không bao giờ nhỏ hơn 9 bit và lớn hơn 80 bit. Một trạm cũng có thể bắt đầu phát các bit trong bộ đệm đi trước khi bộ đệm của nó bị đầy. Dĩ nhiên là tổng thời gian để một thẻ bài di chuyển hết một vòng là một hàm của kích thước của các bộ đệm này.

*Ví dụ:* FDDI là mạng tốc độ 100 Mbps, nó có thời gian xử lý 1 bit là 10 ns. Nếu mỗi trạm cài đặt buffer dài 10 bit và chờ cho đến khi buffer bị đầy một nửa mới bắt đầu truyền, thì mỗi trạm tạo ra thời gian trì hoãn là  $5 \times 10 \text{ ns} = 50 \text{ ns}$  đối với tổng thời gian xoay vòng mạng. FDDI còn có các tính chất vật lý khác. Chẳng hạn, một mạng đơn có giới hạn chuẩn là có tối

da 500 trạm làm việc, với khoảng cách xa nhất giữa một cặp trạm bất kỳ là 2 km. Nhưng trên hết, mạng lại bị giới hạn với kích thước tổng cộng là 200 km cáp quang. Do tính chất là vòng đôi, nên tổng kích thước cáp quang nối tất cả các trạm là 100 km.

Ngoài ra, mặc dù ký tự "F" ám chỉ cáp quang, nhưng chuẩn FDDI đã được định nghĩa để có thể chạy trên một số thiết bị tài khác, bao gồm cả cáp đồng trục và cáp xoắn đôi. Tuy nhiên cũng nên cẩn thận khi xét đến tổng kích thước mà vòng bao phủ. Như chúng ta sẽ thấy dưới đây, lượng thời gian bỏ ra để cho thẻ bài đi hết một vòng mạng sẽ đóng vai trò quan trọng trong giải thuật điều khiển truy cập. FDDI sử dụng phương pháp mã hoá 4B/5B. Do FDDI chuẩn mạng phổ biến đầu tiên sử dụng cáp quang, nên các con chip mã hoá dạng 4B/5B chạy trên tốc độ của FDDI có rất nhiều ngoài thị trường.

#### 8.4.3.2. Giải thuật "Thẻ bài được định thời" – Timed Token

Các luật quy định thời gian giữ thẻ bài trong FDDI phức tạp hơn trong 802.5 một ít. THT cho mỗi trạm được tính như trong phần trình bày về Token Ring, và được tinh chỉnh để có được một giá trị hợp lý.

Ngoài ra, để đảm bảo cho mỗi trạm có cơ hội truyền trong một khoảng thời gian cụ thể nào đó, nghĩa là để đặt một cận trên cho giá trị TRT mà mọi trạm đều thấy được, chúng ta định nghĩa thông số "đích thời gian xoay vòng của thẻ bài" (target token rotation time – TTRT). (Việc các trạm thống nhất với nhau về thời gian TTRT như thế nào sẽ được trình bày trong phần dưới đây). Mỗi trạm đo thời gian giữa hai lần liên tiếp thẻ bài đến nó, chúng ta gọi thời gian này là TRT đo được của trạm (measured TRT). Nếu thời gian TRT đo được của trạm lớn hơn thời gian TTRT được dàn xếp trước đó thì thẻ bài bị trễ, vì thế trạm sẽ không được truyền dữ liệu nữa. Ngược lại, thẻ bài đến sớm và trạm sẽ có quyền giữ thẻ bài trong khoảng thời gian (TTRT – TRT).

Tuy nhiên, có vấn đề phát sinh như sau: Nếu một nút có quá nhiều dữ liệu cần phải gửi có cơ hội giữ thẻ bài, nó sẽ tận dụng hết thời gian giữ thẻ bài được phép. Vì thế nút kế sau nó sẽ tính toán và thấy thời gian TTRT và



TTRT là bằng nhau, nghĩa là nút kế sau này sẽ không có quyền truyền dữ liệu nữa. Để tính đến khả năng này, FDDI định nghĩa hai lớp giao thông trên mạng: đồng bộ và dị bộ. Khi một trạm có được thẻ bài, nó luôn được phép gửi dữ liệu dạng đồng bộ mà không cần phải quan tâm là thẻ bài tới sớm hay trễ. Ngược lại, trạm có thể gửi dữ liệu dạng dị bộ chỉ khi thẻ bài tới sớm. Chú ý rằng các khái niệm đồng bộ và dị bộ ở đây có thể gây hiểu lầm. Bằng cách dùng khái niệm đồng bộ, FDDI ám chỉ rằng giao thông trên mạng là có nhạy cảm với độ trễ thông tin.

*Ví dụ:* Người ta sẽ muốn gửi âm thanh hay video trên mạng FDDI theo kiểu đồng bộ. Ngược lại, những ứng dụng nào quan tâm đến thông lượng của đường truyền thì sẽ thích kiểu truyền dị bộ hơn.

*Ví dụ:* Ứng dụng truyền file trên mạng sẽ muốn sử dụng kiểu lưu thông dị bộ trên FDDI.

Lại thêm vấn đề phát sinh: Do kiểu lưu thông dạng đồng bộ không có quan tâm đến việc thẻ bài đến sớm hay muộn, nên có khả năng nếu trạm cơ - giãn thời gian gửi dữ liệu đồng bộ thì thông số TTRT không còn ý nghĩa gì nữa! Để giải quyết vấn đề này, ta quy định: tổng thời gian gửi dữ liệu đồng bộ trong một vòng xoay của thẻ bài không được vượt quá TTRT.

#### **8.4.3.3. Quản lý thẻ bài**

Các cơ chế mà FDDI dùng để đảm bảo luôn có một thẻ bài hợp lệ chạy trên vòng cũng khác so với 802.5, do chúng dính với quá trình thiết đặt TTRT. Trước tiên, tất cả các trạm trong vòng luôn quan sát để đảm bảo là thẻ bài không bị mất. Để ý rằng một trạm trên vòng luôn có thể thấy được thời gian truyền hợp lệ của khung hay là thẻ bài. Thời gian rồi tối đa giữa những lần truyền hợp lệ mà mỗi trạm nên biết qua là bằng độ trễ của vòng cộng với thời gian bỏ ra để truyền một khung dây. (Trên một vòng có kích thước tối đa, thời gian tối đa để truyền một khung dây là nhỏ hơn 2,5 ms).

Do đó, mỗi trạm sẽ đặt bộ đếm lên, bộ đếm này sẽ hết hạn trong thời gian 2.5 ms. Nếu bộ đếm hết hạn, trạm sẽ nghi ngờ là có vấn đề không ổn và sẽ phát khung "thình cầu". Ngược lại khi bộ đếm chưa hết hạn mà trạm thấy được một sự truyền hợp lệ, nó sẽ đặt lại giá trị tối đa của bộ đếm là 2,5 ms.

Khung thỉnh cầu trong FDDI khác với trong 802.5 do nó có chứa giá trị TTRT mà trạm mời chào (bid for the TTRT), nghĩa là: đó là thời gian xoay vòng của thẻ bài mà trạm cảm thấy đủ để các ứng dụng chạy trên nó thoải mái các ràng buộc về thời gian. Một trạm có thể gửi khung thỉnh cầu trong khi không có thẻ bài trong tay, và nó thường làm vậy khi có nghi ngờ mất thẻ bài hoặc khi mới tham gia vào vòng lần đầu tiên. Nếu khung thỉnh cầu đi được hết một vòng và quay lại, trạm phát sẽ xoá nó đi và hiểu rằng giá trị TTRT được mời chào trong đó là giá trị nhỏ nhất. Bây giờ có thể coi là trạm đang nắm thẻ bài, nghĩa là nó có trách nhiệm chèn vào mạng một thẻ bài hợp lệ, và có thể tiếp tục với giải thuật dùng thẻ bài thông thường. Khi một nút nhận được một khung thỉnh cầu, nó kiểm tra xem giá trị TTRT được mời chào trong đó có nhỏ hơn giá trị TTRT của chính nó không. Nếu nhỏ hơn, nó sẽ đặt lại giá trị TTRT của nó thành giá trị TTRT được mời chào. Nếu lớn hơn, trạm sẽ xoá khung thỉnh cầu này đi, chèn vào vòng một khung thỉnh cầu mới chứa giá trị TTRT được mời chào bằng giá trị TTRT của nó. Trường hợp cuối cùng, nếu giá trị TTRT được mời chào bằng với giá trị TTRT của trạm, trạm sẽ so sánh địa chỉ của trạm gửi khung thỉnh cầu và địa chỉ của nó, địa chỉ nào lớn hơn sẽ thắng – nghĩa là bên thắng sẽ sở hữu khung thỉnh cầu. Do đó, nếu một khung thỉnh cầu đi hết một vòng và trở về trạm gửi, trạm gửi sẽ biết rằng nó là người mời chào giá trị TTRT duy nhất và nó có thể tạo ra một thẻ bài mới một cách an toàn. Tại thời điểm đó, tất cả các trạm bây giờ đều đồng ý rằng giá trị TTRT là giá trị đủ nhỏ để làm cho tất cả chúng cảm thấy ổn.

## 8.5. CÁC THIẾT BỊ MẠNG LAN

### 8.5.1. Modem

Modem (viết tắt từ *modulator and demodulator*) là một thiết bị điều chế sóng tín hiệu tương tự nhau để mã hoá dữ liệu số, và giải điều chế tín hiệu mạng để giải mã tín hiệu số. Một ví dụ quen thuộc nhất của Modem bằng tần tiếng nói là chuyển tín hiệu số '1' và '0' của máy tính thành âm thanh mà nó có thể truyền qua dây điện thoại của Plain Old Telephone Systems

(POTS), và khi nhận được ở đầu kia, nó sẽ chuyển âm thanh đó trở về tín hiệu '1' và '0'.

- Modem thường được phân loại bằng lượng dữ liệu truyền nhận trong một khoảng thời gian, thường được tính bằng đơn vị bit trên giây, hoặc "bps".

- Modem có hai loại: Loại lắp thẳng vào trong máy tính bằng một card riêng được gọi là Modem trong (Internal Modem), hoặc Modem ngoài (External Modem),

Modem ngoài được nối thông qua cổng nối tiếp của máy tính như cổng COM1, COM2.

- Khi nói đến Modem, người ta thường quan tâm tới tốc độ truyền. Đơn vị là Baud = Bit/giây (Thường được ký hiệu là bps, Kbps). Tốc độ thường từ 9600 bps đến 33600bps. Hiện nay đa số Modem có tốc độ 56Kbps.

### 8.5.2. Router (Thiết bị định tuyến)

- Router là thiết bị mạng lớp 3 của mô hình OSI (Network Layer). Router kết nối hai hay nhiều mạng IP với nhau. Các máy tính trên mạng phải "nhận thức" được sự tham gia của một router, nhưng đối với các mạng IP thì một trong những quy tắc của IP là mọi máy tính kết nối mạng đều có thể giao tiếp được với router.

- Ưu điểm của router

Về mặt vật lý, Router có thể kết nối với các loại mạng khác lại với nhau, từ những Ethernet cục bộ tốc độ cao cho đến đường dây điện thoại đường dài có tốc độ chậm.



Hình 8.33. Modem



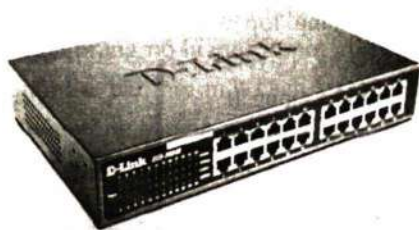
Hình 8.34. Router

- Nhược điểm của router

Router chậm hơn Bridge vì chúng đòi hỏi nhiều tính toán hơn để tìm ra cách dẫn đường cho các gói tin, đặc biệt khi các mạng kết nối với nhau không cùng tốc độ. Một mạng hoạt động nhanh có thể phát các gói tin nhanh hơn nhiều so với một mạng chậm và có thể gây ra sự nghẽn mạng. Do đó, Router có thể yêu cầu máy tính gửi các gói tin đến chậm hơn. Một vấn đề khác là các Router có đặc điểm chuyên biệt theo giao thức – tức là, cách một máy tính kết nối mạng giao tiếp với một router IP thì sẽ khác biệt với cách nó giao tiếp với một router Novell hay DECnet. Hiện nay vấn đề này được giải quyết bởi một mạng biết đường dẫn của mọi loại mạng được biết đến. Tất cả các router thương mại đều có thể xử lý nhiều loại giao thức, thường với chi phí phụ thêm cho mỗi giao thức.

### 8.5.3. Switch (Bộ chuyển mạch mạng)

Switch đôi khi được mô tả như là một Bridge có nhiều cổng. Trong khi một Bridge chỉ có 2 cổng để liên kết được 2 segment mạng với nhau, thì Switch lại có khả năng kết nối được nhiều segment lại với nhau tùy thuộc vào số cổng (port) trên Switch.



Hình 8.35. Switch

Cũng giống như Bridge, Switch cũng "học" thông tin của mạng thông qua các gói tin (packet) mà nó nhận được từ các máy trong mạng. Switch sử dụng các thông tin này để xây dựng lên bảng Switch, bảng này cung cấp thông tin giúp các gói thông tin đến đúng địa chỉ.

Ngày nay, trong các giao tiếp dữ liệu, Switch thường có 2 chức năng chính là chuyển các khung dữ liệu từ nguồn đến đích, và xây dựng các bảng



Switch. Switch hoạt động ở tốc độ cao hơn nhiều so với Repeater và có thể cung cấp nhiều chức năng hơn như khả năng tạo mạng LAN ảo (VLAN).

#### 8.5.4. Hub (Bộ tập trung)

Hub được coi là một Repeater có nhiều cổng. Một Hub có từ 4 đến 24 cổng và có thể còn nhiều hơn. Trong phần lớn các trường hợp, Hub được sử dụng trong các mạng 10BASE-T hay 100BASE-T. Khi cấu hình mạng là hình sao (Star topology), Hub đóng vai trò là trung tâm của mạng.

- Với một Hub, khi thông tin vào từ một cổng và sẽ được đưa đến tất cả các cổng khác.

- Hub có 2 loại là Active Hub và Smart Hub. Active Hub là loại Hub được dùng phổ biến, cần được cấp nguồn khi hoạt động, được sử dụng để khuếch đại tín hiệu đến và cho tín hiệu ra những cổng còn lại, đảm bảo mức tín hiệu cần thiết.



Hình 8.36. Hub

Smart Hub (Intelligent Hub) có chức năng tương tự như Active Hub, nhưng có tích hợp thêm chip có khả năng tự động dò lỗi – rất hữu ích trong trường hợp dò tìm và phát hiện lỗi trong mạng.

#### 8.5.5. Card Ethernet

Các mạng Ethernet hiện đại đều sử dụng cáp đôi xoắn vòng 8 dây. Các dây này được sắp xếp theo thứ tự đặc biệt và đầu nối RJ-45 được gắn vào phần cuối cáp. Cáp RJ-45 trông giống như bộ kết nối ở phần cuối dây điện thoại, nhưng lớn hơn. Các dây điện thoại dùng bộ kết nối RJ-11, tương

phần với bộ kết nối RJ-45 dùng trong cáp Ethernet. Bạn có thể thấy ví dụ một cáp Ethernet với đầu nối RJ-45.

### 8.5.6. Bridge (Thiết bị cầu nối giữa các mạng)

- Bridge là thiết bị mạng thuộc lớp 2 của mô hình OSI (Data Link Layer). Bridge được sử dụng để ghép nối 2 mạng để tạo thành một trạng lớn duy nhất. Bridge được sử dụng phổ biến để làm cầu nối giữa hai mạng Ethernet. Bridge quan sát các gói tin (packet) trên mọi mạng. Khi thấy một gói tin từ một máy tính thuộc mạng này chuyển tới một máy tính trên mạng khác, Bridge sẽ sao chép và gửi gói tin này tới mạng đích.

Ưu điểm của Bridge là hoạt động trong suốt, các máy tính thuộc các mạng khác nhau vẫn có thể gửi các thông tin với nhau đơn giản mà không cần biết có sự "can thiệp" của Bridge. Một Bridge có thể xử lý được nhiều lưu thông trên mạng như Novell, Banyan... cũng như là địa chỉ IP cùng một lúc.



Hình 8.37. Bridge

- Nhược điểm của Bridge là chỉ kết nối những mạng cùng loại và sử dụng Bridge cho những mạng hoạt động nhanh sẽ khó khăn nếu chúng không nằm gần nhau về mặt vật lý.

### 8.5.7. Repeater (Khuếch đại tín hiệu đường truyền mạng)

- Trong một mạng LAN, giới hạn của cáp mạng là 100m (cho loại cáp mạng CAT 5 UTP – là cáp được dùng phổ biến nhất), bởi tín hiệu bị suy hao

trên đường truyền nên không thể đi xa hơn. Vì vậy, để có thể kết nối các thiết bị ở xa hơn, mạng cần các thiết bị để khuếch đại và định thời lại tín hiệu, giúp tín hiệu có thể truyền dẫn đi xa hơn giới hạn này.

- Repeater là một thiết bị ở lớp 1 (*Physical Layer*) trong mô hình OSI. Repeater có vai trò khuếch đại tín hiệu vật lý ở đầu vào và cung cấp năng lượng cho tín hiệu ở đầu ra để có thể đến được những chặng đường tiếp theo trong mạng. Điện tin, điện thoại, truyền thông tin qua sợi quang... và các nhu cầu truyền tín hiệu đi xa đều cần sử dụng Repeater.

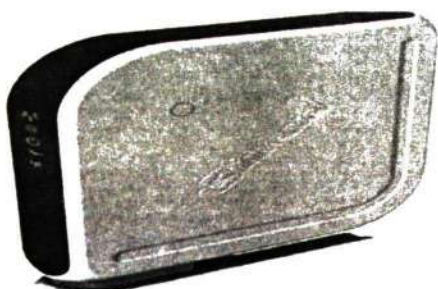


Hình 8.38. Repeater

### 8.5.8. Gateway (Ghép nối và chuyển đổi giữa các giao thức)

- Gateway cho phép nối ghép hai loại giao thức với nhau.

*Ví dụ:* mạng của bạn sử dụng giao thức IP và mạng của ai đó sử dụng giao thức IPX, Novell, DECnet, SNA... hoặc một giao thức nào đó thì Gateway sẽ chuyển đổi từ loại giao thức này sang loại khác.



Hình 8.39. Gateway

Qua Gateway, các máy tính trong các mạng sử dụng các giao thức khác nhau có thể dễ dàng "nói chuyện" được với nhau. Gateway không chỉ phân biệt các giao thức mà còn có thể phân biệt ứng dụng như cách bạn chuyển thư điện tử từ mạng này sang mạng khác, chuyển đổi một phiên làm việc từ xa...

### 8.5.9. Cáp mạng

- Cáp mạng: Có 3 loại phương tiện truyền thông được dùng trong mạng là cáp xoắn, cáp đồng trục và cáp quang.

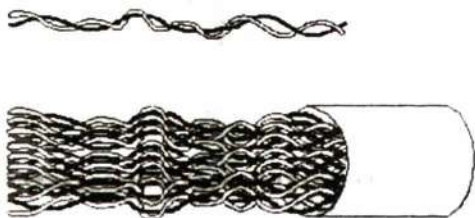
- Đặc tính của cáp:

Bao gồm sự nhạy cảm với nhiễu của điện, độ mềm dẻo, khả năng uốn nắn để lắp đặt, cự ly truyền dữ liệu, tốc độ truyền (Mbit/s). Hiện nay tốc độ truyền dữ liệu trên các loại cáp biến động từ 10Mb/s đến 100Mb/s và hơn nữa.

Một kết nối nối vào một cáp thường được gọi là một rẽ mạch. Kết nối vào một mạng thường được gọi là kết nối rẽ mạch. Kiểu của cáp sẽ làm cho việc kết nối vào mạng khó hay dễ.

- Cáp xoắn:

Gồm hai sợi dây đồng được xoắn cách điện với nhau. Nhiều đôi dây cáp xoắn gộp với nhau và được bọc chung bởi vỏ cáp hình thành cáp nhiều sợi. Cáp này có đặc tính dễ bị ảnh hưởng của nhiễu điện nên chỉ truyền dữ liệu ở cự ly khoảng 100m (328 feet).

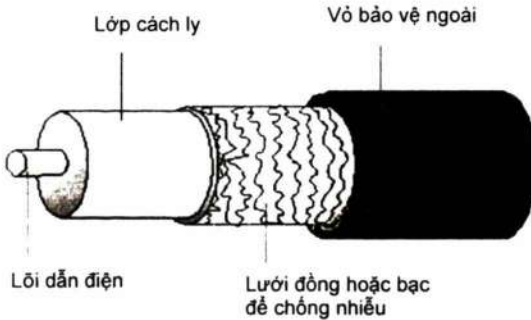


Hình 8.40. Cáp xoắn



- Cáp đồng trục:

Bao gồm một sợi dây dẫn ở giữa, bên ngoài bọc một lớp cách điện rồi đến một lớp lưới kim loại, tất cả được đặt trong một lớp vỏ bọc cách điện. Có hai loại cáp đồng trục phổ biến nhất dùng trong mạng gọi là cáp dày (thicknet) và cáp mỏng (thinnet).

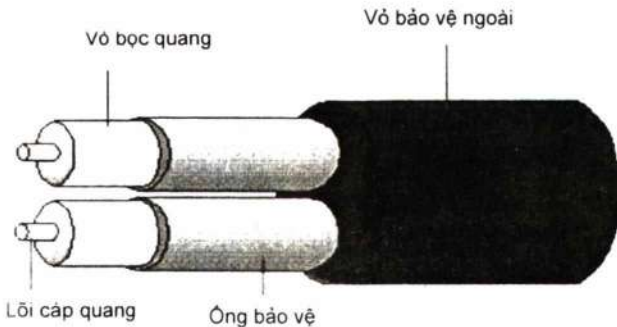


**Hình 8.41.** Cáp đồng trục

Cáp đồng trục có đặc tính ít bị ảnh hưởng của nhiễu và sự suy hao tín hiệu cho nên nó cung cấp một đường truyền dài và tốt hơn cáp xoắn.

- Cáp quang

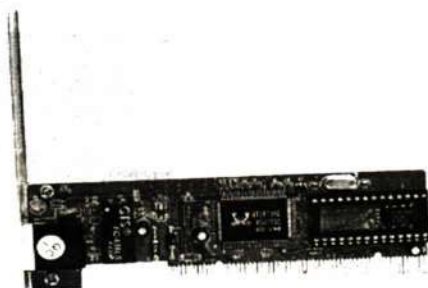
Có đặc tính là không bị ảnh hưởng của nhiễu điện và có thể truyền dẫn ở tốc độ rất cao (hàng 100 Mb/s).



**Hình 8.42.** Cáp quang.

### 8.5.10. Card mạng

Là vi mạch được nối vào máy thông qua BUS PCI hoặc ISA, đầu ra sử dụng các đầu nối để nối dây mạng. Card mạng dùng để thiết lập mạng cho mục đích giao tiếp giữa các máy tính với nhau. Để card mạng hoạt động được, ta phải thiết lập đúng trình điều khiển của nó, địa chỉ của các máy tính trên mạng, và cài đúng giao thức (Protocol) để giao tiếp.



Hình 8.43. Card mạng

## 8.6. CÔNG NGHỆ MẠNG ĐIỆN RỘNG WAN

### 8.6.1. Khái niệm về liên mạng (Internetworking)

Liên mạng (Internetworking) là một tập các mạng riêng lẻ được nối với nhau bởi các thiết bị mạng trung gian, có chức năng như là một mạng đơn. Các mạng thành phần tạo nên liên mạng được gọi là mạng con (*Subnetworks*). Các thiết bị được nối đến các mạng con được gọi là hệ thống đầu - cuối (*End nodes*) và những thiết bị nối các mạng con lại với nhau được gọi là các thiết bị liên kết liên mạng (*Intermediate nodes*). Thuật ngữ "*Internetworking*" thường được sử dụng dưới dạng rút gọn là "*Internet*". Một cách chung nhất, Internet là một tập hợp các mạng được nối với nhau. Khi sử dụng "I" hoa ở trước, thì thuật ngữ "*Internet*" là đề cập đến mạng Internetwork toàn cầu, bao gồm hàng triệu mạng trên thế giới liên kết với nhau và hoạt động theo chuẩn TCP/IP.

Liên mạng có thể được liên kết bởi LAN tới LAN, LAN tới WAN và WAN tới WAN. Có ba phương pháp liên kết liên mạng phổ biến tương ứng với 3 tầng cuối của mô hình OSI. Phương pháp liên kết tại tầng vật lý, cùng cấu trúc và phương thức trao đổi thông tin. Bộ lặp Repeater hoạt động tại tầng vật lý, là thiết bị được sử dụng để mở rộng chiều dài của một mạng LAN. Phương pháp liên kết tại tầng liên kết dữ liệu (Datalink), có cấu trúc khác nhau và phương thức trao đổi thông tin khác nhau. Cầu (Bridge) và các bộ chuyển mạch (Switched) tầng 2 hoạt động tại tầng liên kết dữ liệu. Những thiết bị này hỗ trợ cho các giao thức tầng vật lý khác nhau và có thể liên kết giữa các mạng LAN có cấu trúc khác nhau. Phương pháp liên kết sử dụng tầng mạng (Network Layer) hay tầng Internet (Internet Layer) cho các mạng khác nhau về phần cứng, khác nhau về phần mềm, khác nhau về giao thức và thường cung cấp những chức năng và ứng dụng khác nhau. Thiết bị liên kết liên mạng trợ giúp cho các giao thức mạng như IP, IPX, Apple Talk. Việc nối kết được thực hiện bởi việc định dạng gói tin từ một mạng đến một mạng khác bởi thông tin điều khiển tầng mạng như địa chỉ nguồn, địa chỉ đích. Thực hiện chuyển đổi giao thức mạng (Network Protocol Translation). Một thiết bị cung cấp các liên kết tại tầng mạng được gọi là một bộ định tuyến (Router). Chức năng chủ yếu của một Router là liên kết các mạng khác nhau về vật lý và chuyển đổi các gói tin từ một mạng này sang một mạng khác, quyết định đường đi của các gói tin đến node đích.

## 8.6.2. Mạng tích hợp đa dịch vụ số ISDN (Integrated Service Digital Network)

### 8.6.2.1. ISDN là gì?

Khái niệm về mạng tích hợp đa dịch vụ số được CCITT định nghĩa là: *"Một mạng viễn thông, dựa trên kỹ thuật chuyển kênh và chuyển mạch gói, cung cấp các đường truyền số, có khả năng phục vụ nhiều loại dịch vụ khác nhau, bao gồm dịch vụ thoại và phi thoại. Các thuê bao liên kết mạng phải tuân theo các chuẩn..."*

- Mạng ISDN có những đặc điểm sau:

– Là một mạng đa dịch vụ, thay thế nhiều mạng viễn thông khác nhau đang cùng tồn tại bằng một mạng duy nhất có khả năng cung cấp tất cả các dịch vụ hiện tại và các dịch vụ tương lai với một giao tiếp thuê bao duy nhất.

– ISDN có hệ thống báo hiệu số 7 và các node chuyển mạch thông minh.

– Kiến trúc ISDN tương thích với mô hình OSI. Các giao thức đã được phát triển có liên quan tới các ứng dụng của mô hình OSI có thể sử dụng được trong ISDN. Các giao thức có thể phát triển sử dụng một cách độc lập cho các tầng khác nhau, cho các chức năng riêng của từng tầng mà không ảnh hưởng đến các tầng kế nhau.

– Mục tiêu chính của mạng:

Chuẩn hoá tất cả các thiết bị đầu - cuối, cho phép các phương tiện như âm thanh, hình ảnh, văn bản... được tích hợp chung vào một mạng duy nhất nhằm sử dụng có hiệu quả các tài nguyên của mạng. Nguyên lý chung của ISDN là liên kết các thiết bị đầu - cuối khác nhau lên cùng một đường dây thuê bao và có thể đồng thời truyền thông số giữa thuê bao và mạng. Chi phí được tính theo dung lượng thông tin truyền đi, không tính riêng cho mỗi loại dịch vụ sử dụng. Các dịch vụ khác nhau được hỗ trợ bởi hệ thống báo hiệu số 7 giữa mạng và báo hiệu DSS1 thuê bao.

#### **8.6.2.2. Các phân tử cơ bản của mạng ISDN**

– TE1 (Termination Equipment 1) là các thiết bị đầu - cuối có các thuộc tính ISDN như: điện thoại số ISDN, các đầu - cuối thoại, số liệu, digital fax...

– TE2 (Termination Equipment 2) là các thiết bị đầu - cuối không có tính năng ISDN, để có thể liên kết với ISDN phải có thêm các bộ phối ghép đầu - cuối TA (Terminal Adapter).

– NT1 (Network Termination 1): Thực hiện các chức năng thuộc tầng vật lý của mô hình OSI, tức là các tính năng về điện, về giao tiếp giữa ISDN và người sử dụng, các chức năng kiểm soát chất lượng đường truyền, đấu vòng...



– NT2 (Network Termination 2) là một thiết bị thông minh có khả năng đáp ứng các chức năng đến tầng mạng của mô hình OSI. NT2 có thể là tổng đài riêng PBAX, bộ điều khiển đầu - cuối hoặc là mạng cục bộ LAN.

– R, S, T, U: Các điểm chuẩn phân cách (R: rate, S: system, T: terminal, U: user).

### 8.6.2.3. Các loại kênh trong mạng ISDN

"Kênh" là đường truyền thông tin giữa người sử dụng và mạng, được gọi là kênh thuê bao. Trong ISDN kênh thuê bao chỉ truyền các tín hiệu số và được chia thành 3 loại kênh cơ bản: kênh D, kênh B và kênh H, được phân biệt với nhau về chức năng và tốc độ:

- Kênh D

Dùng để truyền báo hiệu giữa người sử dụng và mạng. Vì có thể không sử dụng hết băng tần của kênh, nên có thể dùng kênh D để truyền dữ liệu người sử dụng. Kênh D hoạt động với tốc độ 16 Kbps hoặc 64 Kbps, phụ thuộc vào giao diện người sử dụng.

- Kênh B

Dùng để truyền tín hiệu tiếng nói, âm thanh (Audio), số liệu và hình ảnh (Video) của người sử dụng. Kênh B luôn hoạt động ở tốc độ 64 Kbps. Ba loại liên kết có thể thiết lập qua kênh B:

– Chuyển mạch kênh (Circuit-switched): Quá trình thiết lập liên kết không thực hiện trên kênh B mà sử dụng hệ thống báo hiệu kênh chung.

– Chuyển mạch gói (Packet-switched): Thuê bao được nối tới một node chuyển mạch gói và số liệu sẽ được chuyển đổi nhờ chuẩn X.25.

– Liên kết bán cố định (Semipermanent): loại liên kết này không đòi hỏi thủ tục thiết lập liên kết, tương tự như thuê bao kênh riêng (Leased line).

- Kênh H

Cung cấp các dịch vụ tốc độ cao và ghép các luồng thông tin ở tốc độ thấp hơn. Có 4 loại kênh H:

- Kênh H0: tương đương với 6 kênh B, có tốc độ 384 Kbps.

- Kênh H10: tương đương với 23 kênh B, có tốc độ 1.472 Mbps.
- Kênh H11: tương đương với 24 kênh B, có tốc độ 1.536 Mbps.
- Kênh H12: tương đương với 30 kênh B, có tốc độ 1.920 Mbps.

#### **8.6.2.4. Giao diện ISDN**

- Giao diện BRI (Basic Rate Interface)

Có cấu trúc kênh là 2B+D, trong đó kênh D hoạt động với tốc độ 16 Kbps. Tổng cộng tốc độ của giao diện này là 144 Kbps nhưng trong thực tế cấu trúc của giao diện tốc độ cơ sở có thể lên tới 192 Kbps. Giao diện này dành cho các thuê bao nhỏ để cung cấp các dịch vụ truy nhập mạng bằng các thiết bị đầu - cuối đa năng hoặc các thiết bị riêng lẻ.

- Giao diện PRI (Primary Rate Interface)

Dùng cho thuê bao có dung lượng lớn như tổng đài PBAX hoặc các mạng cục bộ LAN. Do các tiêu chuẩn truyền dẫn khác nhau nên có 2 loại truy nhập là: 23B+D cho tiêu chuẩn Bắc Mỹ 1544 Kbps và 30B+D cho tiêu chuẩn Châu Âu 2048 Kbps (ở giao diện này kênh D luôn có tốc độ là 64 Kbps).

#### **8.6.2.5. Chức năng các tầng trong kiến trúc ISDN**

Mạng ISDN là sự tích hợp kỹ thuật chuyên mạch kênh và chuyển mạch gói, trên cơ sở số hoá toàn bộ mạng lưới. Vì vậy nó có ưu thế về dịch vụ mà chưa một mạng nào có được.

- Tầng vật lý trong ISDN

Cấu trúc trong tầng này phụ thuộc vào hướng liên kết từ thiết bị đầu - cuối đến mạng hay từ mạng đến thiết bị đầu - cuối. Giao diện của tầng gồm:

- NT Frame (Network to Terminal).
- TE Frame (Terminal to Network).

- Tầng 2 trong ISDN

Tương ứng với tầng liên kết dữ liệu (Data Link Layer) của mô hình OSI. Hoạt động trong tầng này có giao thức LAP-D (Link Access Protocol – D channel). LAP-D được dẫn xuất từ giao thức HDLC (High Level Data Link Control). LAP-D thực hiện các chức năng sau đây:

- Cung cấp dịch vụ thiết lập một hay nhiều liên kết Data Link trên cùng kênh D cho các hoạt động của các thực thể tầng 3.
- Tạo khung (Frame).
- Kiểm soát đồng bộ.
- Phát hiện lỗi và tự động phát lại khung có lỗi.
- Ghi nhận các sai sót về thủ tục.
- Kiểm soát luồng.
- Các chức năng giám sát tầng 2.

Flag	Address field	Control field	Information	FCS	Flag
------	---------------	---------------	-------------	-----	------

**Hình 8.44.** Cấu trúc khung của LAPD

Trong đó:

- Flag: biểu thị sự bắt đầu hay kết thúc của khung.
- Address: Địa chỉ ISDN.
- SAPI (Service Access Point Identifier): Điểm truy nhập dịch vụ cho tầng 3 (6 bit).
- C/R (Command/Response): Khung này là một lệnh hay một đáp ứng (1 bit).
- EA (Extended Address – Higher/Lower Order) bắt đầu hay kết thúc của trường địa chỉ (bằng 1 là byte cuối của địa chỉ) (2 bit).
- TEI (Terminal Endpoint Identifier): Địa chỉ đặc biệt hoặc ấn định ID cho mỗi thiết bị đầu - cuối ISDN liên kết với mạng ISDN thông qua giao diện S/T (7 bit).
- Control: Trường điều khiển.
- Information: Trường dữ liệu.
- PD (Protocol Discriminator) (1 byte).
- L (Length) cho biết chiều dài của trường CRV (1 byte).
- CRV (Call Reference Value) thiết lập số hiệu cho mỗi cuộc gọi (1 hoặc 2 byte).

- MT (Message Type) Loại Message (1 byte).
- MOIE (Mandatory/Optional Information Elements).
- CRC: Trường kiểm tra.
- Tầng 3 trong ISDN

Chức năng của tầng này là thiết lập, duy trì và giải phóng các liên kết. Các thực thể tầng 3 sẽ cung cấp các thông điệp (Message) để truyền trong các trường tầng 2. Các thông điệp thường có độ dài 8 bit và có nhiều loại thông điệp sử dụng trong các trường hợp khác nhau. Ví dụ như thông điệp SETUP (00000101) thiết lập cuộc gọi.

Mạng tích hợp đa dịch vụ số ISDN nếu được triển khai sẽ thực sự là một cuộc cách mạng trong công nghệ thông tin. Từ một mạng duy nhất nó có thể cung cấp các dịch vụ khác nhau mà hiện nay đang được cung cấp bởi các mạng viễn thông khác nhau. Việc triển khai ISDN không chỉ dừng lại ở việc nâng cấp các hệ thống viễn thông hiện có để có khả năng truyền tải được những dòng dữ liệu lớn với tốc độ nhanh mà còn phải tiến hành đồng bộ về phía người dùng và về phía các nhà cung cấp dịch vụ.

### **8.6.3. Mạng băng rộng B\_ISDN (Broadband ISDN)**

#### **8.6.3.1. Tổng quan về sự ra đời của B\_ISDN**

Giữa thập kỷ 80, CCITT đã triển khai nghiên cứu mô hình mạng viễn thông mới gọi là ISDN băng rộng (Broadband- ISDN). B\_ISDN là mạng thông tin số đa dịch vụ, trợ giúp tất cả các ứng dụng đa dịch vụ trên cùng một hệ thống mạng. Nghĩa là mạng phải có khả năng cung cấp các dịch vụ truyền thông với tốc độ thay đổi từ một vài Kbps đến hàng trăm Gbps cho các loại kênh Analog và kênh Digital bao gồm những dịch vụ đang có và những dịch vụ sẽ có trong tương lai.

Công nghệ truyền dẫn không đồng bộ ATM dựa trên nguyên lý truyền dẫn và chuyển mạch gói được CCITT chọn làm giải pháp cho B\_ISDN. Đầu những năm 90, các khuyến nghị cho B\_ISDN dựa trên công nghệ ATM đã được ban hành.



- Giao tiếp B\_ISDN ban đầu cung cấp tốc độ truyền 51 Mbps, 155 Mbps hoặc 622 Mbps trên đường cáp quang.
- Tầng vật lý hỗ trợ B\_ISDN được cung cấp bởi SONET (Synchronous Optical Network) và ATM (Asynchronous Transfer Mode).
- Tầng Client có thể hỗ trợ Frame Relay, SMDS hoặc IEEE 802.2 B\_ISDN có thể được xem như một mạng thông tin được phát triển từ mạng ISDN băng hẹp hiện đang được sử dụng.

### **8.6.3.2. Đặc điểm của dịch vụ B\_ISDN**

Mục tiêu của B\_ISDN là kết hợp tất cả các dịch vụ hiện có vào một mạng truyền thông duy nhất. Về cơ bản nó cung cấp các dịch vụ băng hẹp. Ngoài ra, nó có khả năng cung cấp nhiều dịch vụ băng rộng như điện thoại thấy hình, hội nghị từ xa, truyền số liệu tốc độ cao...

B\_ISDN có khả năng cung cấp dịch vụ băng rộng tốc độ đến Mbit/s, còn các tần số mà nó sử dụng và phân bố thời gian sử dụng thì có phạm vi rất rộng. Đặc tính phân bố khác của tín hiệu dịch vụ B\_ISDN là các tín hiệu liên tục. Các tín hiệu tiếng nói và hình ảnh có thể cùng "sống chung" với các tín hiệu nhóm như số liệu đầu - cuối. Tuy nhiên, các tín hiệu dữ liệu khác nhau có các tốc độ biến đổi rất rộng. Mặt khác, các tín hiệu hình ảnh và âm thanh đòi hỏi phải xử lý theo thời gian thực.

Kỹ thuật chuyển mạch gói lý tưởng đối với tốc độ thấp hoặc số liệu nhóm, trong khi đó, đối với tín hiệu thoại và hình ảnh thì chuyển mạch kênh là thích hợp. Ngoài ra với các tín hiệu thoại cũng thích hợp chuyển mạch phân chia thời gian và với các tín hiệu video tốc độ cao thì thích hợp với chuyển mạch kênh phân chia theo không gian. Vì vậy, tìm được một hệ thống truyền dẫn có khả năng trao đổi các tín hiệu tốc độ thấp/cao và các tín hiệu liên tục/nhóm là cực kỳ khó khăn.

### **8.6.3.3. Cấu trúc chức năng của B\_ISDN**

Mô hình cấu trúc chức năng chung của ISDN băng rộng về cơ bản giống như ISDN băng hẹp. Có nghĩa là về mặt cấu hình tiêu chuẩn, nhóm chức năng và điểm gốc, cả hai cấu trúc đó là như nhau. Nó chỉ ra rằng

B\_ISDN được hình thành trên cơ sở khái niệm của ISDN. Cấu trúc của ISDN băng rộng bao gồm khả năng mức cao và khả năng mức thấp.

Khả năng mức cao là chức năng liên quan đến thiết bị đầu - cuối (TE) và khả năng mức thấp bao gồm khả năng ISDN băng hẹp dựa trên khả năng băng rộng, 64 bit/s và khả năng báo hiệu liên tổng đài.

#### **8.6.3.4. So sánh giữa ISDN và B\_ISDN**

B\_ISDN là một mạng số liên kết đa dịch vụ như ISDN, nhưng việc thiết lập B\_ISDN thực hiện khác với thiết lập ISDN. B\_ISDN nó bảo đảm liên kết các tín hiệu băng rộng và có khả năng đồng thời xử lý các tín hiệu băng rộng, băng hẹp. Mô hình cấu trúc cơ bản của B\_ISDN và ISDN như nhau. Tuy nhiên, chúng chỉ tương tự nhau về mặt khái niệm mà không tương thích về mặt hoạt động. Các thiết bị B\_ISDN không thể hoạt động nếu đầu nối vào mạng ISDN hoặc TE của ISDN không thể đầu nối tới NT của B\_ISDN.

Trong thực tế B\_ISDN khác rất xa với ISDN, vì ISDN tích hợp kỹ thuật chuyển mạch kênh và kỹ thuật chuyển mạch gói, trong khi đó B\_ISDN sử dụng công nghệ ATM hoàn toàn khác với các hệ thống của ISDN. Nghĩa là, trong khi ISDN chủ yếu điều khiển hệ thống thông tin chuyển mạch kênh thì B\_ISDN chủ yếu sử dụng hệ thống thông tin chuyển mạch gói, đồng thời vẫn điều khiển hệ thống thông tin kênh. B\_ISDN khác hẳn so với ISDN.

Sự phát triển của ISDN và B\_ISDN là bước đệm cho sự ra đời các kỹ thuật mạng viễn thông mới với mục đích cung cấp đa dịch vụ trên cùng một mạng viễn thông duy nhất. Mạng thế hệ sau NGN đang được nghiên cứu và phát triển đáp ứng nhu cầu ngày càng cao của xã hội, đó là một bước tiến mới trong kỹ thuật mạng viễn thông? Một câu hỏi được giải đáp trong tương lai không xa.

### **8.6.4. Mạng chuyển mạch gói X.25**

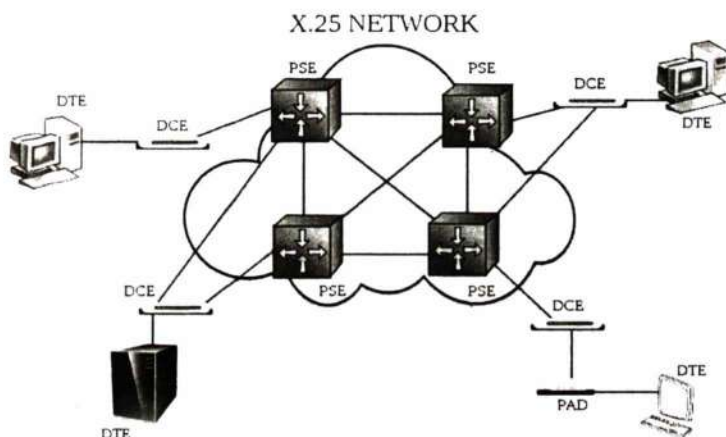
#### **8.6.4.1. Khái quát kỹ thuật mạng X.25**

X.25 định nghĩa chuẩn giao diện giữa các thiết bị đầu - cuối số liệu của người sử dụng DTE (Data Terminal Equipment) với thiết bị kết cuối kênh dữ liệu DCE (Data Circuit Terminating). X.25 có chức năng vừa điều khiển

giao diện DTE/DCE vừa thực hiện chức năng truyền dữ liệu giữa DTE với node của mạng chuyển mạch gói. Các mạng X.25 cung cấp các lựa chọn cho chuyển mạch ảo hoặc cố định. X.25 cung cấp các dịch vụ tin cậy cũng như điều khiển luồng dữ liệu từ node tới node (End to End).

Trong đó:

- PSE – Packet Switching Exchange
- DCE – Data Communications Equipment
- DTE – Data Terminal Equipment



**Hình 8.45.** Mô hình mạng X.25

Các mạng X.25 có tốc độ tối đa 64 Kbps. Tốc độ này thích hợp với các tiến trình truyền thông chuyển giao tệp và các thiết bị đầu - cuối có lượng lưu thông lớn. Tuy nhiên với tốc độ như vậy không thích hợp với việc cung cấp các dịch vụ đòi hỏi từ 1 Mbps trở lên. Vì vậy các mạng X.25 không hấp dẫn khi cung cấp các dịch vụ ứng dụng LAN trong môi trường WAN. Năm 1976, CCITT công bố khuyến nghị loại X về giao thức X.25 trong các mạng chuyển mạch gói công cộng (*Public Packet Switched Networks*).

#### 8.6.4.2. Giao thức X.25

X.25 hoạt động trên 3 tầng: tầng vật lý, tầng liên kết dữ liệu và tầng mạng.

- Tầng vật lý

Tương ứng với tầng vật lý mô hình OSI, giao thức X.25.1 xác định các vấn đề về điện, hàm, thủ tục và kiểu các bộ đầu chuyên được sử dụng. Bao gồm các chuẩn của CCITT X.26/27 và EIA (*USA Electronic Institute Association*), RS:X.21, X.21 Bis, V.32...

- Tầng liên kết dữ liệu

X.25.2 cung cấp các liên kết giữa hai thiết bị đầu - cuối của một tuyến thông tin có độ tin cậy cao, kiểm soát luồng và kiểm soát lỗi. LAP-B (*Link Access Procedure Balanced*) là giao thức LLC tầng con của liên kết dữ liệu, chuẩn hướng bit, hoạt động theo chế độ song công và đồng bộ.

- Tầng cấp mạng

X.25.3 là giao thức giữa một DTE và một DCE. DTE có thể là một PAD còn DCE có thể là một thiết bị X.25. Giao thức X.25 cung cấp các khả năng chọn mạch ảo thường trực hay theo nhu cầu. X.25 yêu cầu cung cấp dịch vụ tin cậy và tính năng điều khiển luồng dữ liệu End to End. Do các thiết bị trên mạng có thể hoạt động theo nhiều mạch ảo, nên X.25 phải cung cấp tính năng điều khiển luồng cho mỗi mạch.

**Bảng 8.3.** Bảng tổng kết các chức năng của các tầng trong mô hình X.25

Tầng 1	Đồng bộ hoá liên kết
Tầng 2	Phát hiện lỗi và phát lại. Điều khiển luồng
Tầng 3	Tạo số thứ tự gói tin. Truyền dữ liệu theo phương thức Datagram. Thực hiện ghép kênh. Thiết lập kết nối và giải phóng kênh ảo. Thực hiện báo hiệu

#### 8.6.4.3. Hoạt động của giao thức X.25

X.25 hoạt động dựa trên cơ sở kênh cố định PVC (Permanent Virtual Chanel) và kênh ảo chuyển mạch SCV (Switch Virtual Chanel). PCV thay



thể chức năng cho kênh liên kết điểm - điểm cố định giữa các thiết bị đầu - cuối. Sử dụng loại kênh này, giao diện có hiệu quả hơn nhờ sự liên kết được đảm bảo và không bị trễ cuộc gọi. SVC sử dụng tối đa sự mềm dẻo linh hoạt của chuyển mạch gói trong thực tế.

Hoạt động của X.25 theo các giai đoạn: giai đoạn thiết lập kênh ảo, giai đoạn trao đổi thông tin và giai đoạn giải phóng kênh ảo. Ngay sau khi thiết lập kênh ảo, một thông báo tóm tắt của cấu trúc gói tin sẽ được node nguồn gửi đi đến node đích. Nếu chấp nhận, node đích sẽ hiển thị và thông báo lại cho node nguồn. Đường truyền song hướng được thiết lập. Giai đoạn trao đổi dữ liệu: Node nguồn gửi khung thông tin, node đích sẽ tiến hành kiểm tra tính hợp lý của khung thông qua các bit FCS. Nếu không hợp lý thì loại bỏ khung và gửi thông báo lại cho node nguồn biết, yêu cầu truyền lại. Nếu khung là hợp lý thì node này tiếp tục các thủ tục truyền gửi khung tới node tiếp theo trong mạng, đồng thời thông báo lại cho node nguồn biết là đã nhận được thông tin. Node nguồn sau khi đã nhận được thông báo âm từ node đích, tiếp tục gửi gói tin tiếp theo... Sau khi kết thúc, kênh ảo sẽ được giải phóng.

Như vậy hoạt động của X.25 cho phép sử dụng một cách có hiệu quả kênh thông tin liên kết giữa người sử dụng và các node mạng. Các thủ tục của tầng mạng đảm bảo trao đổi thông tin có tỷ lệ lỗi bit thấp, với xác suất lớn các gói tin được gửi tới đích không có lỗi, đúng thứ tự, điều này rất cần thiết đối với các đường truyền có độ tin cậy không cao.

## 8.6.5. Mạng chuyển mạch khung Frame Relay

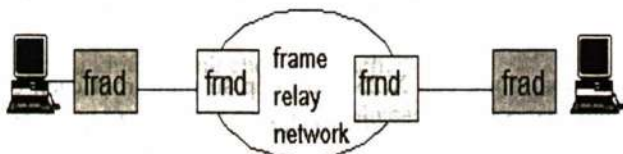
### 8.6.5.1. Giới thiệu chung

Những năm cuối của thế kỷ XX các hệ thống viễn thông sử dụng công nghệ cáp quang có độ tin cậy cao, đảm bảo tốc độ và chất lượng truyền dẫn, giảm thiểu tình trạng nghẽn mạch và tỷ lệ lỗi dữ liệu. Các giao thức trước đây cho mạng chuyển mạch gói đặc tả các thủ tục quản lý lưu lượng, quản lý tắc nghẽn và xử lý lỗi, đảm bảo tính thống nhất, toàn vẹn thông tin trên đường truyền đã trở nên phức tạp, cồng kềnh, làm giảm thông lượng.

Frame Relay ra đời như là một công nghệ kế thừa những đặc điểm ưu việt của mạng chuyên mạch gói như tính tin cậy, mềm dẻo, khả năng chia sẻ tài nguyên. Đồng thời hạn chế tối đa thủ tục kiểm soát, hỏi đáp... không cần thiết gây ra độ trễ lớn. Nó cho phép tận dụng các ưu thế về tốc độ truyền tải và tính ổn định của công nghệ truyền dẫn, thoả mãn nhu cầu dịch vụ tốc độ cao, sử dụng nhiều thông lượng mạng điện rộng WAN trên đó truyền tải một lượng lớn dữ liệu với nhiều định dạng khác nhau.

Công nghệ Frame Relay tích hợp tính năng dồn kênh tĩnh và chia sẻ công nghệ X.25. Dữ liệu được tổ chức thành các khung có độ dài không cố định được đánh địa chỉ tương tự như X.25. Tuy nhiên, khác với X.25, Frame Relay loại bỏ hoàn toàn các thủ tục ở tầng 3 trong mô hình OSI. Chỉ một số chức năng chính ở tầng 2 được thực hiện. Vì vậy tốc độ truyền trong mạng Frame Relay cao hơn nhiều so với X.25 và mạng Frame Relay được gọi là mạng chuyên mạch gói tốc độ cao.

#### 8.6.5.2. Cấu hình tổng quát mạng Frame Relay



Hình 8.46. Cấu hình tổng quát mạng Frame Relay.

Các kênh riêng tạo ra liên kết vật lý giữa DTE và DCE. DTE còn được gọi là thiết bị truy nhập mạng FRAD (*Frame Relay Access Device*) thường là các Router, Bridge, ATM Switch... DCE còn được gọi là thiết bị mạng FRND (*Frame Relay Network Device*) là các thiết bị chuyển mạch Frame Relay Switch. FRAD và FRND chuyển đổi dữ liệu thông qua các quy định của giao tiếp UNI. Mạng trục của Frame Relay có thể là các mạng viễn thông IP, PSTN...

#### 8.6.5.3. So sánh Frame Relay với X.25

Sự khác biệt giữa căn bản giữa công nghệ Frame Relay và X.25 là Frame Relay không kế thừa công nghệ X.25 mà là một giao thức tiên tiến có

nhiều điểm tương đồng với X.25. X.25 là một giao thức của công nghệ chuyển mạch gói, đặc tả giao tiếp giữa DTE và DCE. Dữ liệu trong tầng 3 của X.25 sẽ được chia thành các gói (Packet), trong mỗi gói được bổ sung phần Network Header. Các gói này sẽ được chuyển xuống tầng 2, các hàm chức năng của LAP-B sẽ bổ sung Layer 2 Header và các Flag vào mỗi gói tạo thành các khung LAP-B. Các khung sẽ được chuyển xuống tầng vật lý và truyền đến đích.

Hoạt động của các thực thể chặt chẽ, các node mạng X.25 phải luôn biết trạng thái của mạng trong mỗi liên kết logic. Các gói tin điều khiển và báo nhận, báo mất (ACK/NACK) thường xuyên được truyền trên cùng liên kết của gói tin dữ liệu không chỉ tại các giao tiếp DTE-DCE mà còn tại tất cả các node mạng. Tại các node mạng phải duy trì bảng trạng thái cho mỗi liên kết logic để quản lý liên kết và điều khiển lỗi và lưu lượng, đảm bảo gói tin đến đúng địa chỉ đích được lưu trong Network Header và số lượng gói tin gửi vào mạng không được vượt quá khả năng xử lý của mạng. Như vậy các giao thức tại tầng mạng là tuyệt đối cần thiết nhất là khi triển khai hệ thống mạng X.25 trên các đường truyền có độ tin cậy thấp, dễ bị nhiễu loạn, suy giảm tín hiệu...

Frame Relay được thiết kế để loại bỏ những hạn chế trong các mạng X.25 khi triển khai trên tuyến truyền dẫn tốc độ cao bằng cách:

- Các gói tin điều khiển và dữ liệu được truyền trên các liên kết logic riêng biệt. Vì vậy, tại các node không cần duy trì bảng trạng thái, không xử lý các gói tin điều khiển.

- Dồn kênh, chuyển mạch các liên kết logic được thực hiện ở tầng liên kết.

- Loại bỏ các quá trình xử lý ở tầng mạng.

- Không điều khiển lưu lượng và điều khiển lỗi theo từng đoạn mạng (Hop-by-Hop Control). Trong trường hợp cần thiết sẽ để các tầng cao hơn đảm trách. Frame Relay chỉ sử dụng một phần các chức năng ở tầng 2 nên khung thông tin của Frame Relay sẽ có cấu trúc đơn giản hơn so với khung thông tin của X.25 nhưng vẫn duy trì đặc điểm của một khung thông tin quy định bởi giao thức điều khiển.



Khung Frame Relay không có Header của tầng mạng. Vì Frame Relay không sử dụng các thủ tục điều khiển lưu lượng, điều khiển lỗi của tầng mạng. Mặt khác, giao thức được sử dụng tại tầng liên kết chỉ là phần lõi của giao thức điều khiển (LAP-F Core) nên việc xử lý tại các node mạng sẽ ít hơn nhiều so với X.25. Kích thước phần dữ liệu (User Data) trong khung Frame Relay có thể tối đa 2048 byte trong khi phần dữ liệu trong khung X.25 chỉ có thể đạt tối đa 128 byte. DCE thực hiện ba chức năng chính:

- Kiểm tra các khung, loại bỏ các khung có lỗi.
- Căn cứ vào địa chỉ trong khung chọn đường.
- Kiểm tra có bị nghẽn hay không. Nếu có thì lập bit báo nghẽn hoặc loại bỏ khung tùy trường hợp cụ thể.

#### 8.6.5.4. Frame Relay và mô hình OSI

##### • Tầng vật lý

Các giao thức chuẩn định nghĩa giao tiếp vật lý giữa thiết bị truy nhập FRAD và thiết bị mạng FRND, giữa các node mạng theo chuẩn giao tiếp vật lý của ISDN. Frame Relay tương thích với nhiều giao diện vật lý khác nhau như V.35, X.21...

##### • Tầng liên kết

Các thủ tục liên kết của Frame Relay được định nghĩa bằng giao thức truy cập LAP-D và LAP-F. Giao thức truy cập LAP-F được cải tiến từ LAP-D và được sử dụng phổ biến trong các mạng Frame Relay. Để quản lý liên kết và truyền dữ liệu LAP-F chia thành 2 tầng chức năng là Upper Function (LAP-F Upper) và Core function (LAP-F Core).

Core Function: có các chức năng kiểm soát độ dài khung, phát hiện lỗi đường truyền, điều khiển nghẽn qua trường báo hiệu trong cấu trúc khung.

Upper Function: có chức năng điều khiển DLCI (Data Link Connection Identifier), xác định liên kết logic giữa FRAD và FRND.

##### • Tầng mạng (Network Layer)

Tầng mạng định nghĩa các khung dữ liệu lưu chuyển trong hệ thống, đảm bảo việc định tuyến trong một mạng hay giữa các mạng với nhau.

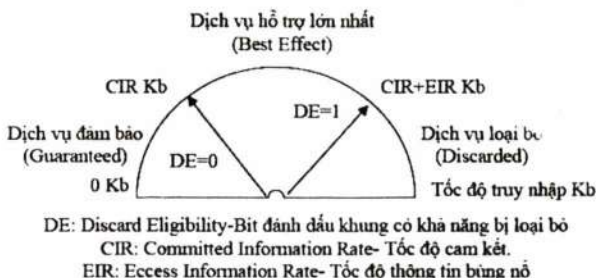


Trong Frame Relay, các giao tiếp giữa DTE và DCE tầng 3 không có thủ tục nên tốc độ nhanh hơn nhiều so với X.25. Tuy nhiên, nếu một liên kết logic được thiết lập động (SVC), Frame Relay có thể sử dụng một phần của giao thức đặc tả chuẩn Q.931 của giao thức điều khiển ISDN (còn gọi là Q.933) để thiết lập liên kết. Giao thức liên kết hai node mạng X.25 là X.75, còn để liên kết hai node mạng Frame Relay người ta sử dụng giao diện NNI (Network to Network Interface).

### 8.6.5.5. Điều khiển quản lý lưu lượng

Hầu hết các nhà cung cấp dịch vụ Frame Relay đều sử dụng phương thức tốc độ cam kết CIR (Committed Information Rate) để giải thích chính xác thông tin nào được truyền đến một dịch vụ đảm bảo, thông tin nào nhận được dịch vụ hỗ trợ lớn nhất và thông tin nào bị loại bỏ ở cổng vào của Frame RelayH node nguồn. Với lưu lượng trên kênh PVC có tốc độ trong khoảng giữa 0 Kbps và phụ thuộc người sử dụng. Khi đó chúng sẽ được truyền đi bình thường qua mạng mà không bị tổn hao đó là dịch vụ đảm bảo "Guaranteed". Đối với các khung thông tin vượt quá CIR một lượng tốc độ thông tin bùng nổ EIR (*Excess Information Rate*) thì có thể bị Frame Relay node tiếp theo huỷ nếu xảy ra nghẽn. Đây chính là dịch vụ hỗ trợ lớn nhất. Khi dữ liệu vượt quá ngưỡng CIR + EIR thì các khung thông tin sẽ bị huỷ ngay bởi Frame RelayH node nguồn cho đến khi tốc độ của người sử dụng giảm xuống dưới ngưỡng CIR + EIR.

Tốc độ EIR thường được các nhà khai thác mạng đặt bằng đúng tốc độ CIR.



Hình 8.47. Điều khiển quản lý lưu lượng

### 8.6.5.6. Các dịch vụ *Frame Relay*

Hiện nay, phần lớn các dịch vụ mạng *Frame Relay* được cung cấp dưới hai dạng:

- Mạng dịch vụ công cộng (Public Carrier–Provided Networks)

*Frame Relay* và FRAD, FRND của nhà cung cấp, khách hàng được tính cước trên cơ sở thông số mạng đã thuê, việc bảo trì và quản trị do các nhà cung cấp thực hiện.

- Mạng riêng doanh nghiệp

Các doanh nghiệp có quy mô toàn cầu triển khai các mạng *Frame Relay* riêng. Toàn bộ thiết bị mạng là tài sản của doanh nghiệp. Công tác quản trị, vận hành và bảo dưỡng do chính doanh nghiệp đó thực hiện.

Hiện tại, ở Việt Nam phổ biến hình thức mạng dịch vụ công cộng do giá thành sử dụng rẻ hơn, không đòi hỏi doanh nghiệp duy trì đội ngũ nhân viên kỹ thuật chuyên trách. Ngân hàng Á Châu (ACB) là một trong số các đơn vị đang khai thác hiệu quả dịch vụ này.

*Frame Relay* là công nghệ được ưu tiên lựa chọn bởi ngày càng có nhiều người dùng đang tìm kiếm các giải pháp mạng diện rộng trên nền tảng hạ tầng viễn thông hiện đại. Mặc dù đã có nhiều công nghệ mới ra đời có tính năng hiện đại hơn nhưng với xu thế khách hàng đang ưa chuộng mạng trên nền IP, *Frame Relay* tiếp tục thể hiện tính ưu việt qua khả năng kết hợp mạng IP với các ưu điểm như quản lý dịch vụ dễ dàng, truyền dữ liệu tốc độ cao an toàn, chi phí liên kết thấp. Có thể khẳng định, công nghệ *Frame Relay* vẫn có thể được tiếp tục sử dụng hiệu quả trong thời gian dài.

## CÂU HỎI VÀ BÀI TẬP

1. Nêu các phương pháp truy cập đường truyền.
2. Trình bày phương pháp truy cập đường truyền phân chia theo thời gian TDM.
3. Trình bày phương pháp truy cập ngẫu nhiên.
4. Nêu các thiết bị mạng Lan.
5. Trình bày các công nghệ Internet.
6. So sánh các thiết bị mạng Lan.
7. Trình bày các loại cáp mạng.
8. Trình bày hoạt động của bộ chuyển mạch Switch, Hub và Router.
9. So sánh các phương pháp truy cập đường truyền.
10. Trình bày công nghệ diện rộng WAN.
11. Nêu các dịch vụ Frame Relay.
12. Trình bày hoạt động của giao thức X.25.

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Trần Quang Vinh, *Cấu trúc máy tính*; NXB Giáo dục Việt Nam, 2010.
- [2] Văn Thế Minh, *Kỹ thuật vi xử lý*, NXB Giáo dục, 2008.
- [3] Tống Văn On, *Giáo trình cấu trúc máy tính*, NXB Giáo dục Việt Nam, 2010.
- [4] Trần Quang Vinh, *Nguyên lý phần cứng và kỹ thuật ghép nối máy vi tính*; NXB Giáo dục Việt Nam, 2010.
- [5] Nguyễn Mạnh Giang, *Lập trình bằng ngôn ngữ Assembly cho máy tính PC-IBM*, NXB Giáo dục, 2007.
- [6] Nguyễn Nam Trung, *Cấu trúc máy tính và các thiết bị ngoại vi*; NXB Khoa học và Kỹ thuật, 2004.
- [7] Nguyễn Thúc Hải, *Mạng máy tính và các hệ thống mở*; NXB Giáo dục, 1999.
- [8] James F. Kurose and Keith W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*; Addison Wesley, 2001.
- [9] Andrew S.Tanenbaum, *Computer Networks*; Prentice Hall, 2003.



*Chịu trách nhiệm xuất bản:*

Chủ tịch Hội đồng Thành viên kiêm Tổng Giám đốc NGÔ TRẦN ÁI  
Tổng biên tập kiêm Phó Tổng Giám đốc NGUYỄN QUÝ THAO

*Chịu trách nhiệm nội dung:*

Phó Tổng biên tập NGÔ ÁNH TUYẾT  
Giám đốc Công ty CP Sách ĐH – DN NGÔ THỊ THANH BÌNH

*Biên tập nội dung và sửa bản in:*

ĐỖ HỮU PHÚ

*Thiết kế mỹ thuật và trình bày bìa :*

NGUYỄN NGỌC ANH

*Thiết kế sách và chế bản :*

ĐỖ PHÚ

---

Công ty CP Sách Đại học – Dạy nghề, Nhà xuất bản Giáo dục Việt Nam  
giữ quyền công bố tác phẩm.

---

**GIÁO TRÌNH MÁY TÍNH VÀ MẠNG MÁY TÍNH**

---

**Mã số: 7B842Y3-DAI**

Số đăng kí KHXB : 171-2013/CXB/ 1-154/GD

In 500 cuốn (QĐ in số : 13), khổ 16 x 24 cm.

In tại Công ty CP in Phúc Yên.

In xong và nộp lưu chiểu tháng 03 năm 2013.

